Dynamic Base Station Clustering and Scheduling for Coordinated Multi-Point in Cellular Networks

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Sebastian Scholz

geb. in Backnang

Hauptberichter:	Prof. Dr.-Ing. Andreas Kirstädter
Mitberichter:	Prof. Dr.-Ing. Christian Wietfeld
Tag der Einreichung:	19. Januar 2018
Tag der mündlichen Prüfung:	9. Dezember 2019

Institut für Kommunikationsnetze und Rechnersysteme
der Universität Stuttgart

2019

Abstract

Today's cellular wireless networks face ever-increasing traffic demands, while operating close to the theoretical limits of the scarce radio spectrum. Consequently, the traditional approach to increase the network capacity by using more spectrum is not sustainable and new ways to cope with the demands have to be found. Besides higher throughput demands also new applications with special requirements, like low latency or high reliability, arise. Examples are connected vehicles for autonomous driving or smart devices in the Internet of Things.

Several approaches to increase the network performance by improvements in the physical and medium access control layer have been discussed and also partly introduced in Long Term Evolution (LTE) developed by the 3rd Generation Partnership Project. Examples include Multiple Input Multiple Output (MIMO) and Multi-User MIMO (MU-MIMO) transmission, link adaptation by variable modulation and coding schemes or channel and traffic aware scheduling. Also, the coordination between base stations, which is known as Coordinated Multi-Point (CoMP), was studied. But due to the high complexity CoMP is not introduced in real-world networks today. The complexity of CoMP depends on the degree of coordination. While the synchronization and coordination overhead for coordinated scheduling/coordinated beamforming is relatively low, it is higher for CoMP variants based on Joint Processing (JP). However, the performance improvement is generally higher, if the coordination is tighter. Approaches to reduce the complexity of CoMP have been proposed by creating clusters of base stations and allowing coordination only between base stations belonging to the same cluster. However, due to the distributed nature of cellular networks, these attempts were not practically feasible.

Invisible for the users of the network, the operators strive to simplify network management and operation. This is achieved by network automation as provided by Self Organizing Networks (SON) and new network architectures like the Cloud Radio Access Network (C-RAN), which builds on top of standard Information Technology (IT) components and cloud software concepts. The main goal of the C-RAN approach is to centralize all network intelligence and processing. This is achieved by splitting the functionality of the former base stations into two parts. The Remote Radio Heads (RRHs), which consist of antennas, power amplifiers and analog-digital-converters, are placed at the previous base station locations and provide the basic connectivity between User Equipment (UE) and the network. All signal and protocol processing is performed by Baseband Units (BBUs), which are located in a central data center. RRHs and BBUs are connected by a high capacity fronthaul network, e.g., realized by optical fibers. Besides the simplified network operation the network centralization is also an enabler for advanced SON concepts.

This thesis presents the Dynamic Cloud Clustering and Scheduling Framework (DCCSF), which solves the task of dynamic base station clustering in combination with scheduling in a C-RAN architecture. The dynamic clustering is thereby performed by a specifically tailored algorithm. Because Joint Transmission (JT), which is a variant of JP-CoMP, offers the best performance improvements, DCCSF is designed to support JT only. DCCSF is embedded into a C-RAN architecture, by introducing a new component, the cluster manager, in the data center. The cluster manager carries out the necessary procedures for DCCSF. Furthermore, DCCSF encompasses a virtual representation of each configured cluster in the data center, which is called cluster entity. A cluster entity bundles BBUs, which perform the actual processing, a scheduler, which carries out the resource allocation, and all UE state information of the cluster. The input on which the dynamic clustering algorithm creates the base station clusters is provided by the UEs. By careful design of the algorithm, the complexity is kept low and the necessary changes to introduce DCCSF in existing systems are minimal. The dynamic clustering algorithm has two basic configuration parameters: the maximum cluster size, i.e., the number of base stations per cluster, and the interval between cluster reconfigurations. Both parameters aim to limit the complexity of DCCSF. The main novelty of DCCSF is the reduction of the scheduling effort by introducing the concept of partitions, which further divide a cluster into smaller fractions. The scheduler uses the partitions to determine which UEs should be served by which base stations and for which UEs JT is applied. The partitions are created similarly as the clusters from the information provided by the UEs. Although LTE serves as a basis for DCCSF, it could also be adapted to upcoming 5G networks.

The second contribution of the thesis is a comprehensive system-level simulation model to evaluate the performance of CoMP in urban environments. This model consists of sub-models of the cellular network based on LTE, the wireless channel, an abstraction of the performance of CoMP, data traffic and user mobility models. The major feature of the user mobility model is a group-like user movement, which is observed in urban environments. The network data traffic model represents web browsing traffic in the way it is generated by many applications in cellular networks.

Multiple simulation studies show the significant performance gains DCCSF achieves in comparison to static clustering schemes or systems without CoMP. DCCSF thereby improves the total system capacity, the rate per user and the rate per transmitted traffic object. Thus, the performance from the perspective of the network operator, but also the user perceived performance is improved. In detailed evaluations the influence of the two configuration parameters of DCCSF is analyzed. A further part of the evaluation is dedicated to the influence of different dynamic effects on the performance of DCCSF and CoMP in general. The evaluations focus especially on the relation between DCCSF and different mobility patterns. This comprises the mobility itself, but also the distribution of the users on the considered area.

We can conclude from the evaluations that DCCSF significantly improves the cellular network performance. The required cluster sizes to achieve this are relatively small. A cluster size of three base stations already results in substantial performance improvements. Also, the necessary cluster reconfiguration interval depends on the speed of the users. E.g., for movement speeds of up to 100 km/h a cluster reconfiguration interval of 1 s is sufficient. Finally, we can observe that DCCSF performs best if the user movements are correlated and local user hotspots are available.

Kurzfassung

Heutige zelluläre Mobilfunknetze stehen einerseits stetig ansteigenden Verkehrsvolumenanforderungen gegenüber und arbeiten andererseits nahe am theoretischen Limit des begrenzt verfügbaren Frequenzspektrums. Daher ist der klassische Ansatz der Kapazitätssteigerung durch Bereitstellung von mehr Spektrum nicht zukunftsweisend und neue Ansätze zur Erfüllung der Bedürfnisse werden benötigt. Neben höheren Bandbreitenanforderungen werden zelluläre Mobilfunknetze auch mit neuen Anwendungen umgehen müssen, die spezielle Anforderungen haben, wie beispielsweise kurze Latenzen oder hohe Zuverlässigkeit. Beispiele sind vernetzte Fahrzeuge für autonomes Fahren und intelligente Geräte im Internet der Dinge (Internet of Things).

In der Vergangenheit wurden verschiedene Ansätze zur Kapazitätssteigerung durch Verbesserungen in der physikalischen Ebene und des Medienzugriffs diskutiert und teilweise in Long Term Evolution (LTE) des 3rd Generation Partnership Project eingeführt. Beispiele hierfür umfassen Übertragung mit Multiple Input Multiple Output (MIMO) und Multi-User MIMO (MU-MIMO), Anpassung an den Funkkanal durch variable Modulation und Kodierung oder Ressourcenzuteilung unter Berücksichtigung von Kanal- und Verkehrskenntnis. Außerdem wurde die Koordination zwischen Basisstationen, welche auch als „‚Coordinated Multi-Point‘“ (CoMP) bekannt ist, untersucht, jedoch aufgrund der hohen Komplexität nicht in realen Netzen eingeführt. Die Komplexität hängt dabei vom Grad der Koordination ab. Während die Synchronisations- und Koordinierungsaufwände für „Coordinated Scheduling/Coordinated Beamforming“ relativ gering sind, sind sie für CoMP-Varianten basierend auf „‚Joint Processing‘“ (JP) größer. Allerdings sind die Leistungsverbesserungen allgemein höher, wenn die Koordination enger ist. Durch Bildung von Gruppen („Clustern“) von Basisstationen und Anwendung von CoMP nur zwischen Basisstationen, die zum gleichen Cluster gehören, kann die Komplexität von CoMP reduziert werden. Jedoch stellt die dezentrale Architektur von zellulären Mobilfunknetzen ein Hindernis für die Einführung der CoMP-Cluster dar.

Unsichtbar für die Nutzer des Mobilfunknetzes haben die Netzbetreiber das Bestreben, den Betrieb und die Verwaltung ihrer Netze zu vereinfachen. Dies wird zum einen durch die Automation des Netzes, beispielsweise durch „Self Organizing Networks“ (SON), und zum anderen durch neue Netzwerkarchitekturen wie das „Cloud Radio Access Network“ (C-RAN) ermöglicht. Das Konzept des C-RAN basiert auf der Verwendung von Standard IT-Komponenten und Cloud-Software-Paradigmen. Das Hauptziel von C-RAN ist die Zentralisierung der Verarbeitung und Berechnung. Dies wird durch die Aufspaltung der Funktionalität der bisherigen Basisstationen in zwei Teile erreicht. Die „Remote Radio Heads“ (RRHs), welche aus Antennen, Leistungsverstärkern und Analog-Digital-Wandlern bestehen, befinden sich weiterhin an den Standorten der bisherigen Basisstationen und stellen die grundlegende Verbindung zwischen Endgeräten und Mobilfunknetz her. Alle Signal- und Protokollverarbeitungen werden von den „Baseband Units“ (BBUs) durchgeführt, welche sich in einem zentralen Rechenzentrum befinden. RRHs und BBUs sind durch ein leistungsfähiges Fronthaul-Netz miteinander verbunden. Dieses kann zum Beispiel als optisches Netz realisiert werden. Neben dem vereinfachten Netzbetrieb ermöglicht die Netzwerkzentralisierung außerdem die vereinfachte Einführung von SON-Techniken.

Diese Arbeit präsentiert das „‚Dynamic Cloud Clustering and Scheduling Framework‘“ (DCCSF), welches die Aufgabe der dynamischen Gruppierung von Basisstationen in Kombination mit der Ressourcenzuteilung („Scheduling“) in C-RAN-Architekturen löst. Die dynamische Gruppierung der Basisstationen wird dabei von einem speziellen Algorithmus durchgeführt. Weil „‚Joint Transmission‘“ (JT), welches eine Variante von JP-CoMP darstellt, die höchsten Leistungsverbesserungen ermöglicht, ist DCCSF nur für diese CoMP-Variante ausgelegt. DCCSF ist in die Architektur eines C-RANs eingebettet und führt eine neue Komponente, den „‚Cluster Manager‘“, im Rechenzentrum ein. Der Cluster Manager führt alle für DCCSF notwendigen Prozeduren aus. Außerdem bildet DCCSF jeden konfigurierten Cluster im Rechenzentrum durch eine virtuelle Repräsentation ab, welche „‚Cluster Entity‘“ genannt wird. Eine Cluster Entity bündelt alle dem Cluster zugewiesenen BBUs, welche die eigentlichen Berechnungen ausführen, einen Scheduler, der für die Ressourcenzuteilung zuständig ist, und die Statusinformationen aller mobilen Endgeräte eines Clusters. Die Informationen, welche der dynamische Gruppierungsalgorithmus für die Bildung der Cluster von Basisstationen verwendet, werden von den Endgeräten der Nutzer zur Verfügung gestellt. Durch den angepassten Entwurf des dynamischen Gruppierungsalgorithmus, werden die Komplexität und der zusätzliche Aufwand reduziert und die notwendigen Änderungen im System gering gehalten. Der Algorithmus wird durch zwei Parameter konfiguriert: die maximale Größe der Cluster (die Anzahl der Basisstationen in einem Cluster) und die Zeit zwischen aufeinanderfolgenden Rekonfigurationen von Clustern. Beide Parameter zielen darauf ab, die Komplexität von DCCSF zu beschränken. Die Hauptneuheit von DCCSF ist die Reduktion des Aufwands für die Ressourcenzuteilung durch Einführung des Konzepts der Partitionen, welche einen Cluster in kleinere Teile unterteilen. Der Scheduler verwendet die Partitionen, um daraus abzuleiten, welche Nutzer von welchen Basisstationen bedient werden und für welche Nutzer JT angewendet wird. Obwohl LTE als Basis für DCCSF verwendet wird, lässt sich das Konzept auch auf zukünftige Netze der fünften Generation (5G) übertragen.

Der zweite Beitrag dieser Arbeit ist ein vollständiges Simulationsmodell für die Leistungsbewertung von CoMP auf Systemebene. Das Gesamtmodell umfasst mehrere Komponenten, wie Modelle für das LTE-basierte zelluläre Mobilfunknetz und des Funkkanals, eine Abstraktion der Leistungsfähigkeit von CoMP und Modelle für den Datenverkehr und die Mobilität der Nutzer. Das Hauptmerkmal des Mobilitätsmodells ist die Bewegung der Nutzer in Gruppen, welche in städtischen Umgebungen auftritt. Das Datenverkehrsmodell stellt durch Internetsurfen erzeugten Verkehr dar, so wie er von vielen Anwendungen in Mobilfunknetzen erzeugt wird.

Die durchgeführten Simulationsstudien zeigen die deutliche Leistungssteigerung durch Einführung von DCCSF im Vergleich zu statischer Gruppierung von Basisstationen und Systemen ohne CoMP. DCCSF erhöht dabei die Gesamtkapazität des Netzes, die Rate der einzelnen Nutzer und die Rate der einzelnen Verkehrsobjekte. Die Leistungsfähigkeit kann daher sowohl aus Sicht des Netzbetreibers als auch aus Sicht der Nutzer gesteigert werden. In detaillierten Simulationsstudien wird der der Einfluss der beiden Konfigurationsparameter von DCCSF analysiert. Ein weiterer Teil der Leistungsbewertung widmet sich dem Einfluss verschiedener dynamischer Effekte auf DCCSF und CoMP im Allgemeinen. Besonderes Augenmerk gilt dem Einfluss der Nutzermobilität auf DCCSF. Dabei wird auf die Bewegung selbst, aber auch auf die Verteilung der Nutzer auf der betrachteten Fläche eingegangen.

Aus den Simulationsergebnissen lässt sich schließen, dass DCCSF die Leistungsfähigkeit des zellulären Mobilfunknetzes deutlich steigert. Die dafür notwendigen Größen der Cluster sind relativ klein. Eine Clustergröße von drei Basisstationen führt schon zu erheblichen Leistungssteigerungen. Weiterhin hängt die notwendige Intervalldauer für die Rekonfiguration der Cluster von der Geschwindigkeit der Nutzer ab. Beispielsweise ist für Bewegungsgeschwindigkeiten von bis zu 100 km/h ein Rekonfigurationsintervall von 1 s ausreichend. Außerdem lässt sich feststellen, dass DCCSF besonders gut funktioniert, wenn die Bewegung der Nutzer korreliert ist und es Orte mit hoher Nutzerdichte gibt.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

1G	first generation	1
2G	second generation	1
3G	third generation	1
3GPP	3rd Generation Partnership Project	1
4G	fourth generation	1
5G	fifth generation	2
ABS	Almost Blank Subframes	18
ACK	Acknowledgement	60
API	Application Programming Interface	66
ASK	Amplitude Shift Keying	42
BBU	Baseband Unit	2
BCCH	Broadcast Control Channel	10
BCH	Broadcast Channel	10
BPSK	Binary Phase Shift Keying	42
BS	Base Station	1
BSR	Buffer Status Report	77
C-RAN	Cloud Radio Access Network	2
Capex	Capital Expenses	14
CCCH	Common Control Channel	10
CCDF	Complementary Cumulative Distribution Function	157
CDF	Cumulative Distribution Function	127
CDR	Call Detail Record	68
CE	Cluster Entity	77
CM	Cluster Manager	76
CMR	Cluster Measurement Report	79
CoMP	Coordinated Multi-Point	2
CPRI	Common Public Radio Interface	78
CQI	Channel Quality Indicator	49
CRI	CSI-RS Resource Indicator	49
CRS	Cell-Specific Reference Signals	49
CS/CB	Coordinated Scheduling/Coordinated Beamforming	15
CSI	Channel State Information	19
CSI-IM	Channel State Information Interference Measurement	50
CSI-RS	Channel State Information Reference Signals	49
CU	Central Unit	17
DASH	Dynamic Adaptive Streaming over HTTP	63
DCCH	Dedicated Control Channel	10
DCCSF	Dynamic Cloud Clustering and Scheduling Framework	3
DL-SCH	Downlink Shared Channel	10

DM-RS	Demodulation Reference Signals	47
DPC	Dirty Paper Coding	42
DPS	Dynamic Point Selection	16
DRX	Discontinuous Reception	61
DTCH	Dedicated Traffic Channel	10
eICIC	enhanced Inter-Cell Interference Coordination	18
eMBB	enhanced Mobile BroadBand	14
eNodeB	evolved NodeB. Sometimes also abbreviated as eNB.	8
EPA	Equal Power Allocation	44
EPC	Evolved Packet Core	7
FDD	Frequency Division Duplex	12
FEC	Forward Error Correction	11
feICIC	further enhanced Inter-Cell Interference Coordination	19
FIFO	First In, First Out	107
FSC	Facing Sectors Clustering	24
FSTD	Frequency-Switched Transmit Diversity	46
FTP	File Transfer Protocol	106
GSM	Global System for Mobile Communications	1
HARQ	Hybrid Automatic Repeat Request	11
HII	High Interference Indicator	18
HSPA	High Speed Packet Access	1
HSS	Home Subscriber Server	9
HTTP	HyperText Transfer Protocol	62
IAT	Inter-Arrival Time	108
ICIC	Inter-Cell Interference Coordination	18
IMT-Advanced	International Mobile Telecommunications-Advanced	1
IoT	Internet of Things	1
IP	Internet Protocol	2
IT	Information Technology	151
ITU-R	Radiocommunication Sector of the International Telecommunication Union	1
JP	Joint Processing	15
JT	Joint Transmission	16
LoS	Line of Sight	53
LPP	LTE Positioning Protocol	80
LTE	Long Term Evolution	1
M2M	Machine to Machine	1
MAC	Medium Access Control	9
MCS	Modulation and Coding Scheme	11
MET	Multi-User Eigenmode Transmission	34
MIMO	Multiple Input Multiple Output	4
MME	Mobility Management Entity	8
MMSE	Minimum Mean Square Error	41
mMTC	massive Machine Type Communication	14
MTC	Machine Type Communication	14
MU-MIMO	Multi-User MIMO	7
NFV	Network Function Virtualization	79

NLoS	Non-LoS	53
NP	Nondeterministic Polynomial Time	29
OBSAI	Open Base Station Architecture Initiative	78
OFDM	Orthogonal Frequency Division Multiplexing	10
OFDMA	Orthogonal Frequency Division Multiple Access	1
OI	Overload Indicator	18
Opex	Operating Expenses	14
OS	Operating System	63
OSI	Open Systems Interconnection	9
P-GW	Packet Data Network Gateway	9
PBCH	Physical Broadcast Channel	9
PDCCH	Physical Downlink Control Channel	10
PDCP	Packet Data Convergence Protocol	9
PDSCH	Physical Downlink Shared Channel	9
PEP	Performance Enhancement Proxy	61
PF	Proportional Fair	32
PHR	Power Headroom Report	77
PMI	Precoding Matrix Indicator	47
PRB	Physical Resource Block	11
PUCCH	Physical Uplink Control Channel	49
PUSCH	Physical Uplink Shared Channel	49
QAM	Quadrature Amplitude Modulation	11
QoE	Quality of Experience	69
QoS	Quality of Service	9
QPSK	Quadrature Phase Shift Keying	11
RAN	Radio Access Network	122
RAT	Radio Access Technology	7
RB	Resource Block	11
RBG	Resource Block Group	11
RE	Resource Element	10
RI	Rank Indicator	49
RLC	Radio Link Control	9
RNTP	Relative Narrowband Transmit Power	18
RR	Round Robin	69
RRC	Radio Resource Control	12
RRH	Remote Radio Head	2
RS	Reference Signals	48
RSRP	Reference Signal Received Power	20
RSRQ	Reference Signal Received Quality	79
RSSI	Received Signal Strength Indicator	79
RTT	Round Trip Time	60
S-GW	Serving Gateway	8
SC	Site Clustering	23
SC-FDMA	Single-Carrier Frequency Division Multiple Access	2
SDN	Software-Defined Networking	21
SFBC	Space-Frequency Block Coding	46
SINR	Signal to Interference and Noise Ratio	33

SMS	Short Message Service	68
SNR	Signal to Noise Ratio	33
SOCP	Second Order Cone Program	29
SON	Self Organizing Network	68
SRS	Sounding Reference Signals	48
SU-MIMO	Single-User MIMO	38
TCP	Transmission Control Protocol	59
TDD	Time Division Duplex	12
TM	Transmission Mode	45
TP	Transmission Point	2
TTI	Transmission Time Interval	10
TX	transmit	125
UDP	User Datagram Protocol	59
UE	User Equipment	1
UMTS	Universal Mobile Telecommunications System	1
URLLC	Ultra-Reliable and Low Latency Communication	14
VM	Virtual Machine	77
VoLTE	Voice over LTE	60
VRB	Virtual Resource Block	11
W-CDMA	Wideband Code Division Multiple Access	1
WF	Water Filling	44
WSRM	Weighted Sum-Rate Maximization	29
ZF	Zero Forcing	32
ZP CSI-RS	Zero Power Channel State Information Reference Signals	50

Notation and Symbols

Font Styles

$\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$	Vectors are denoted with bold small letters.
$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$	Matrices are denoted with bold capital letters.
$\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$	Sets are denoted with calligraphic capital letters. Sets are defined with brackets, e.g., $\mathcal{A} = [1, 2, 3]$. Empty sets are denoted with $\emptyset$. If the entries have a distinct order, the values can be accessed with the following operation: $value = \mathcal{A}[index]$. The index starts with zero and goes up to the length of the set minus one.
$\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$, $\boldsymbol{\mathcal{C}}$	Maps (associative arrays) are denoted with bold calligraphic capital letters. Maps are defined with braces, e.g., $\boldsymbol{\mathcal{A}} = \{\dots\}$. Values are accessed by the following operation: $value = \boldsymbol{\mathcal{A}}\{key\}$. If the entries of the map have a distinct order, the values can be accessed by providing an index: $value = \boldsymbol{\mathcal{A}}[index]$. The index starts with zero and goes up to the length of the map minus one.
`STATE`	Typewriter style is used to indicate states.

Mathematical Operations and Symbols

$\mathrm{ld}(\bullet)$	Binary logarithm $\mathrm{ld}(\bullet) = \log_2(\bullet)$
$\mathbb{C}$	Complex numbers
$\mathbf{A}^H$	Hermitian transpose of a matrix
$\mathbf{I}$	Identity matrix
$\mathbf{A}^{-1}$	Inverse of a matrix
$\mathbf{A}^{+}$	Moore-Penrose (pseudo-)inverse of a matrix
$\cap$	Intersection of two sets
$\cup$	Union of two sets or of a single element and a set. E.g., $a = 1$ and $\mathcal{A} = [2, 3, 4]$ then $a \cup \mathcal{A} = [1, 2, 3, 4]$

Cluster Measurement Report Messages

m	CMR message with attributes $m.w$, $m.\boldsymbol{\mathcal{B}}$ and $m.\mathrm{UE}$
$m.\mathrm{UE}$	UE issuing the CMR message m
$m.w$	Weight of UE issuing the CMR message m
$m.\boldsymbol{\mathcal{B}}$	Map associating TPs to their RSRP
$\mathcal{M}$	Set of received CMR messages at the CM

Cluster Definitions

$\boldsymbol{\mathcal{B}}_{\text{UE}}$	Map of best TP per UE (key: UE u, value: TP b)
T_R	Cluster reconfiguration interval
S_C	Cluster size of cluster C
$S_{C,\max}$	Maximum cluster size of all clusters in the system
w_C	Weight of cluster C
C	A cluster with the attributes $C.\mathcal{B}$ which is the set of TPs belonging to the cluster and $C.\mathcal{U}$ which is the set of UEs belonging to the cluster. The cardinality of a cluster is defined as $\lvert C\rvert = \lvert C.\mathcal{B}\rvert$
$\boldsymbol{\mathcal{C}}$	Map of cluster candidates (key: cluster candidate C, value: weight w_C)
$\mathcal{C}$	The configured clustering, i.e., the set of selected clusters
$C.\mathcal{B}$	Set of TPs in cluster C
$C.\mathcal{U}$	Set of UEs in cluster C
P	Partition with attributes $P.\mathcal{B}$ and $P.\mathcal{U}$
$\mathcal{P}$	Partitioning, set of partitions ($\mathcal{P} = [P_1, P_2, \dots]$)
$\mathcal{S}_{\mathcal{P}}$	Set of partitionings ($\mathcal{S}_{\mathcal{P}} = [\mathcal{P}_1, \mathcal{P}_2, \dots]$)

Symbols and Constants

$\boldsymbol{\mathcal{A}}_C$	Resource allocation for cluster C. Keys are partitions and values are resource allocations $\boldsymbol{\mathcal{A}}_P$ of partitions P	
$\boldsymbol{\mathcal{A}}_{\mathcal{P}}$	Resource allocation for partitioning $\mathcal{P}$. Keys are partitions and values are resource allocations $\boldsymbol{\mathcal{A}}_P$ of partitions P	
$\boldsymbol{\mathcal{A}}_P$	Resource allocation for partition P. Keys are RBs and values are sets of scheduled UEs	
s_C	Scheduling performance metric for cluster C	
$s_{\mathcal{P}}$	Scheduling performance metric for partitioning $\mathcal{P}$	
s_P	Scheduling performance metric for partition P	
A_s	Area of one square used to calculate the local user density	m^2
b	Transmission Point (TP)	
f_c	Carrier frequency of the wireless system	Hz
$\mathbf{H}$	Complex channel matrix	
$\mathbf{H}_{\text{eff}}$	Effective channel, channel including (linear) transmitter-side precoding, $\mathbf{H}_{\text{eff}} = \mathbf{HW}$	
$h_{u,b}$	Channel attenuation between UE u and TP b	dB
$h_{u,b,r}$	Frequency selective channel attenuation between UE u and TP b on RB r	dB
T_C	Coherence time	s
τ_c	Core network delay in the downlink direction	s
t_{RAN}	RAN delay in the downlink direction	s
ρ	Local user density	m^{-2}
$d_{u,b}$	Distance between UE u and TP b	m
D_{is}	Inter-site distance	m
D_s	Doppler spread	Hz
b_h	Height of a TP/BS over ground	m

u_h	Height of a UE over ground	m
$\mathbf{n}$	Noise vector	
σ^2	Power of the thermal noise at the receiver	W
N_{TP}	Number of TPs	
N_{RX}	Number of receive antennas per UE	
N_{TX}	Number of transmit antennas per BS	
N_{UE}	Number of UEs	
P_{TX}	Transmit power per transmit antenna	W
$\mathbf{W}$	Precoding matrix	
r_o	Data rate per traffic object	bit/s
R_o	Average data rate per traffic object	bit/s
r_u	Data rate of UE u	bit/s
R_u	Average data rate per UE	bit/s
r	Resource Block index (RB)	
α_s	Scheduling ratio	%
$N_{\mathrm{options}}^{\mathrm{DCCSF}}$	Number of scheduling options considered in DCCSF	
$N_{\mathrm{options},P}^{\mathrm{DCCSF}}$	Number of scheduling options considered in DCCSF in partition P	
$N_{\mathrm{options}}^{\mathrm{exhaustive}}$	Number of scheduling options considered in exhaustive search	
$N_{\mathrm{options},\mathcal{P}}^{\mathrm{exhaustive}}$	Number of scheduling options considered in exhaustive search in partitioning $\mathcal{P}$	
$\mathcal{R}$	Set of RBs per slot of 0.5 ms	
$\mathcal{B}$	Set of TPs	
$\mathcal{U}$	Set of UEs	
$\gamma_{\max}$	Maximum SINR used in system-level simulations to reflect the maximal spectral efficiency of LTE	dB
γ	SINR	dB
S	Similarity of all user movements	$-1 \leq S \leq 1$
S_{u_i,u_j}	Similarity of the movements of users u_i and u_j	$-1 \leq S_{u_i,u_j} \leq 1$
s	Spectral efficiency	bit/s/Hz
c	Speed of light, 2.9979×10^8 m/s	m/s
t_p	Periodicity of CSI-RS transmissions	s
t	Transmit antenna	
u	User Equipment (UE)	

1 Introduction

1.1 Evolution of Cellular Mobile Networks

The basis of today's cellular mobile communication networks has been laid by experiments performed by H. Hertz at the end of the 19th century where he showed that electromagnetic waves could be used to transmit information. Only a few years later G. Marconi used this concept to transmit data over the Atlantic Ocean [TV05]. In the middle of the 20th century, AT&T introduced the concept of cells to realize a commercial nationwide communication network in the United States of America [DPS16].[1]

This development led to the introduction of the so-called first generation (1G) networks. They relied on analog signal transmission, so the service quality was not satisfactory. Also, there was no common standard available at that time, such that the interoperability between the networks was not guaranteed.

Later in the 1980s advances in digital communication technology led to the development of second generation (2G) networks. In Europe, the Global System for Mobile Communications (GSM) standard was developed to provide mobile telephony. Similar activities were also organized by US and Japanese standardization bodies. 2G mainly targeted low bandwidth applications like telephony and data services with a maximum data rate of 9.6 kbit/s. In the mid-1990s the need for more bandwidth was evident, which led to extended versions of 2G networks. 2G systems are still in use today and provide a backbone for many services with low service but high coverage requirements. Especially for Machine to Machine (M2M) communication in the upcoming Internet of Things (IoT) 2G is still often used.

The third generation (3G) was developed by multiple standardization organizations in parallel. During this process, the European and Japanese proposals were merged into what is now known as the Universal Mobile Telecommunications System (UMTS) and the standardization organization 3rd Generation Partnership Project (3GPP) was formed. The system is based on Wideband Code Division Multiple Access (W-CDMA). With the extensions of High Speed Packet Access (HSPA) and HSPA+ it supports higher data rates for packet data.

For the fourth generation (4G), which is the latest generation deployed today, the Radiocommunication Sector of the International Telecommunication Union (ITU-R) published requirements known as International Mobile Telecommunications-Advanced (IMT-Advanced) in 2008 [ITU08]. 3GPP Long Term Evolution (LTE) is the dominant standard today. However, the first release of LTE (release 8) did not fulfill all IMT-Advanced requirements. Only release 10, which is known as LTE-Advanced, achieves this. LTE uses multicarrier techniques to transmit the data. In the downlink (from Base Station (BS) to User Equipment (UE)) Orthogonal Frequency Division Multiple Access (OFDMA) is used, whereas

[1] The following historical overview of cellular communication networks is mainly inspired by [TV05], [STB11] and [DPS16].

for the uplink Single-Carrier Frequency Division Multiple Access (SC-FDMA) is applied. LTE is tailored for a packet-only operation, such that it uses the Internet Protocol (IP) for all data transmissions, including telephony services. With further improvements in subsequent releases, LTE operates closely to the Shannon limit under good radio channel conditions [Mog+07]. However, the received signals in real systems are often distorted by bad channel quality and interference from neighboring transmitters, i.e., BSs in the downlink or UEs in the uplink. Consequently, performance improvements are hard to achieve by enhancements in the physical layer. Instead, the network has to be enhanced from protocol and network operation side.

In the context of research and development of fifth generation (5G) cellular networks, besides improvements in the physical layer to support new applications with special requirements like low latency or high reliability, also advanced network architectures and improved network operations are analyzed. An example for a new network architecture is the Cloud Radio Access Network (C-RAN) approach, which splits the former BSs into Remote Radio Heads (RRHs) and Baseband Units (BBUs) [Che+15]. The RRHs consist of all analog transmission components and are located at the previous BS positions. The BBUs are placed in a central office and perform all necessary signal and protocol processing to operate the network. The BBUs are realized by standard Information Technology (IT) components and cloud software concepts. Coordinated Multi-Point (CoMP) is an example for improved network operation that coordinates the signal transmission of multiple transmitters to reduce interference and to improve the overall network performance. For the realization multiple techniques have been discussed [Saw+10; Lee+12]. Generally there exists a trade-off between performance gains and additional complexity of CoMP.

1.2 Problem Statement

The bandwidth demand in cellular networks is ever-increasing, as new applications like video streaming and social networking become usable on mobile devices. For the near future, it is even expected that cellular traffic will exceed the traffic in fixed access networks [San16]. Additionally, the overall downlink traffic volume is larger than the uplink volume and the growth is expected to be higher [Eri17]. As the available radio bandwidth is a scarce resource, simply increasing the bandwidth is no viable solution. Other ways to handle the demand are required.

One option is CoMP, which is a possibility to increase the capacity of cellular networks by coordinating the signal transmission or reception of multiple transmitters or receivers. Because the downlink is more important, the focus of this thesis is only on downlink CoMP. CoMP uses the more general term Transmission Point (TP) to denote the transmitters. The TPs in cellular networks are realized by the BSs for downlink CoMP. However, CoMP techniques introduce additional complexity in the network, because the signal transmission of multiple TPs has to be coordinated. Another drawback is the increased signaling overhead for radio channel measurements and reporting of the measured channel state required to enable CoMP [MF11].

A promising way to evade those drawbacks is the concept of clustering [Bas+17]. This means that not all available TPs may cooperate, but coordination is only possible between subsets of TPs. These subsets of TPs and the associated UEs are called clusters. Clusters can be defined statically, meaning that TPs always belong to the same cluster and UEs are assigned to one of the available clusters. However, this is only an inferior solution, because UEs in cellular networks are often not static, but moving and the clusters should be defined such that they actually reflect the positions of the UEs. This is achieved by dynamic clustering. For the definition of the clusters the current network state has to be gathered first and based on this the most suitable clustering has to be derived. To achieve optimal performance, the dynamic clustering cannot be determined isolated from the other system functionalities. E.g., the configured clustering directly influences which UEs can be scheduled. Thus, scheduling and clustering must follow the same optimization goals to achieve optimal performance. The same also holds for the relation between dynamic clustering and signal processing for precoding. On the other hand, introducing CoMP increases the complexity of scheduling and signal processing, because more degrees of freedom are available to serve the UEs. This reasons why a framework for joint dynamic clustering and scheduling is required to make CoMP feasible in cellular networks. Such a framework is developed in this thesis.

As the development of CoMP and dynamic clustering is mainly driven by communication engineering, also the evaluation in existing research is based on the methods of communication engineering. Consequently, the scenarios considered in typical evaluations are small in terms of evaluated time and the number of coordinated TPs and neglect relevant aspects from higher layers. These encompass the characteristics of the transmitted traffic, user mobility and user densities in the network. So the need for a system-level assessment of the performance of dynamic clustering for CoMP is evident and is provided in this thesis.

1.3 Contributions

This thesis provides the following two main contributions:

1. A novel framework for joint dynamic clustering and scheduling for CoMP in cellular networks embedded into a C-RAN architecture.
2. A system-level simulation model to evaluate CoMP and the proposed framework in a dynamic urban scenario, including user mobility and data traffic.

We develop the Dynamic Cloud Clustering and Scheduling Framework (DCCSF) as a solution to the mentioned problems of combining dynamic clustering and scheduling for CoMP in cellular networks to increase the system performance. DCCSF is designed with a focus on direct applicability in cellular networks realized as a C-RAN. Thus, the major building block of DCCSF is a specifically tailored algorithm executing a heuristic that dynamically defines the clusters. Besides the clusters, DCCSF generates additional information that is used by the schedulers in the system to assign resources with reduced effort. Even if LTE serves as a basis for the design of DCCSF, the concept is generically applicable to cellular networks if they support CoMP. Therefore, DCCSF is a candidate to improve the performance of upcoming 5G networks.

DCCSF is controlled by two configuration parameters: the maximum allowed cluster size, i.e., the number of TPs within a cluster, and the interval between cluster reconfigurations. Both parameters fulfill the purpose to trade off the performance gain with the additional complexity caused by DCCSF. While larger clusters generally result in increased performance, this also increases the complexity and additional effort of DCCSF. The same holds for the cluster reconfiguration interval. Short intervals allow adapting to the current situation more quickly and thus improve the performance of DCCSF, but on the other hand, might be demanding due to the frequent reconfigurations.

We study the characteristics of DCCSF in system-level simulations. In order to do so, we develop a comprehensive simulation model. Key aspects are an abstraction of CoMP transmissions and the inclusion of dynamic network effects. The first aspect is necessary to allow the evaluation of the system on longer timescales, while still realistic results are obtained. As we discuss in the course of the thesis, dynamic network effects, e.g., user mobility and user-generated network traffic, influence the system performance. Especially in the case of dynamic clustering, where the performance improvements are highly dependable on finding a suitable clustering for the current situation and a fast adaption to changing situations, dynamic effects have to be modeled accurately. In this thesis, we provide a complete model of dynamic effects occurring in urban environments, with a particular focus on vehicular users, as they will play an important role in next-generation networks and autonomous driving. The developed model is not only suitable for evaluating the performance of DCCSF, but can also be used for general evaluations of CoMP in cellular networks.

1.4 Outline

This thesis is structured in seven chapters, including this introduction in Chapter 1.

Chapter 2 presents the relevant background starting with the architecture of cellular networks, which is shown using the example of LTE. This is followed by an introduction of different variants of CoMP in cellular networks and dynamic clustering. We develop a classification scheme for dynamic clustering approaches and present an extensive literature review. Scheduling, which is the second aspect of DCCSF, is also treated in this chapter. Then we provide a brief overview of transmission techniques for Multiple Input Multiple Output (MIMO), which is the basis of several CoMP realizations. Finally, we discuss how accurate knowledge of the wireless channel is obtained by the transmitter in cellular networks in general and in LTE networks in particular.

Chapter 3 deals with several aspects of dynamic effects occurring in cellular networks. The main cause of these effects is the behavior of the users. On the one hand, users are often not static but moving, which influences the characteristics of the wireless channel. On the other hand, users cause network data traffic with application specific behavior. We also present how traditional cellular networks cope with the user-generated dynamic effects.

The main contribution of this thesis, the framework for dynamic clustering and scheduling in cellular networks, is developed in Chapter 4. From the stated design goals, we derive the architecture of DCCSF and present its two aspects, namely dynamic clustering and scheduling. Additionally, we demonstrate how DCCSF is integrated in the normal system operation. We conclude this chapter by classifying DCCSF according to the scheme we develop in Chapter 2 and compare it with other dynamic clustering algorithms.

In Chapter 5, we lay the basis for the following performance evaluation. We introduce all parts of the simulation model consisting of an LTE cellular network, a network traffic model, user mobility models and a method to abstract the performance of CoMP from real transmission techniques. We also demonstrate how a specific scheduling algorithm is included in DCCSF.

Chapter 6 presents the results of the performance evaluation. The first part of this chapter deals with the fundamental system behavior, while the second part is dedicated to multiple sources of network dynamics, as introduced in Chapter 3. Multiple movement patterns and their influence on the network performance are evaluated. Additionally, the relation between the network data traffic and the characteristics of DCCSF is assessed.

Finally, we conclude the thesis in Chapter 7. Additionally, an outlook on future research directions is given.

2 Coordinated Multi-Point (CoMP) in Cellular Mobile Networks

This chapter introduces the necessary background of CoMP in cellular networks. Because examples for real-world implementations are given for several aspects of CoMP, we first introduce an existing cellular network architecture in Section 2.1. Here LTE serves as an example. Section 2.2 introduces the concept of CoMP and shows how it can be implemented in today's and future cellular networks. As the subject of this thesis is on combining dynamic clustering and scheduling for CoMP, we discuss these topics in Sections 2.3 and 2.4. Even if the focus of the thesis is on the system-level behavior of CoMP, it is necessary to discuss the required transmission techniques, to derive the principal characteristics. We introduce the basics of CoMP and Multi-User MIMO (MU-MIMO) transmissions in Section 2.5. Finally, we complete this chapter with an elaboration on obtaining knowledge of the channel state in Section 2.6, as precise channel information is a requirement for CoMP.

2.1 Long Term Evolution (LTE) System Architecture

LTE serves as an example for a cellular radio network in this work. Therefore, this section provides the necessary background information in Sections 2.1.1 to 2.1.5. Section 2.1.6 gives an outlook on alternative network architectures, which are beneficial in case cooperative techniques are introduced in the network. Finally, Section 2.1.7 gives a brief outlook on 5G networks.

The first specification of LTE was published as release 8 in the year 2009. Subsequent releases followed every one to two years. Release 9 followed in 2010, release 10 in 2011, release 11 in 2013, release 12 in 2015, release 13 in 2016 and release 14 in 2017. Each release bundles a set of new techniques and improvements to increase the system performance and adapt the network to upcoming challenges.

2.1.1 System Architecture

Besides the standardization of LTE as a new Radio Access Technology (RAT), also the whole network architecture was redesigned with the goal to simplify the architecture in comparison to previous generations. The new architecture of the core network, called Evolved Packet Core (EPC), is responsible for all non radio access related functionalities required to operate a cellular network.

Figure 2.1 depicts the basic entities and interfaces of the EPC. More details can be found in the official 3GPP description [3GPP 36.300a] and in secondary literature [DPS16; Ols+09]. As we can see, there is a clear separation of control data, which is handled in the

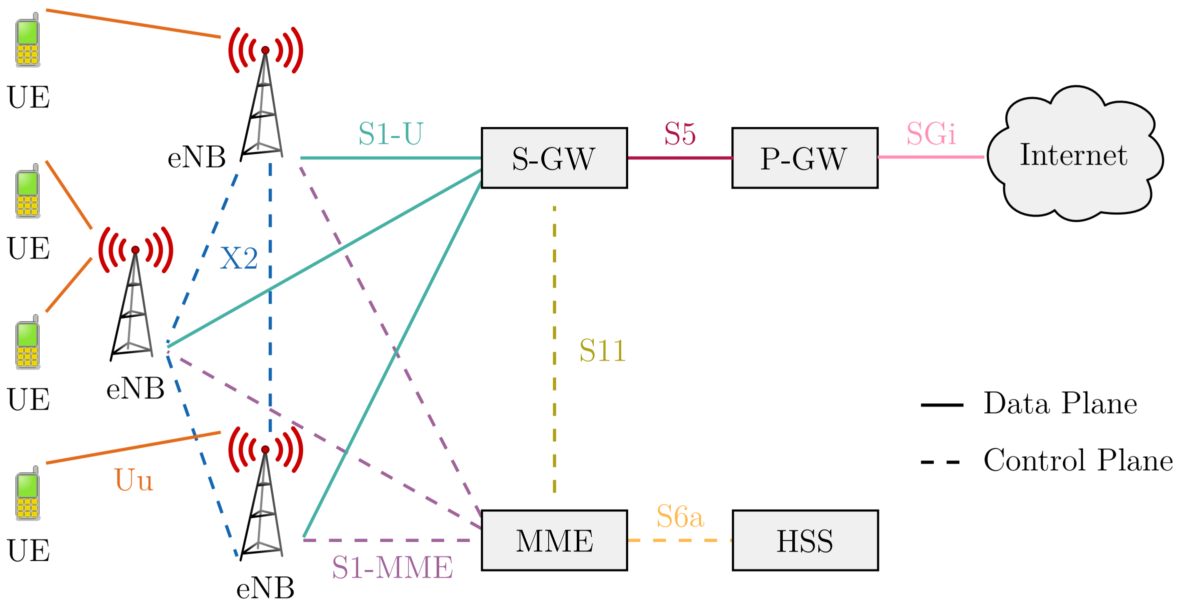


Figure 2.1: LTE network architecture

control plane, and user data, which is transferred in the data plane. One reason for this separation is that this allows scaling both planes independently of each other if the traffic volume increases. E.g., the traffic on the control plane is mainly influenced by the number of users in the network, whereas the traffic volume on the data plane depends on the capability of the devices and their behavior. Also, we have to note that the connections shown in Figure 2.1 are logical connections, which means that there is not necessarily a physical link between the connected entities. Because the EPC builds on top of an IP network, all possibilities of routing and traffic engineering can be used to provide these logical connections. Similarly, the entities do not necessarily need to be realized as separate devices. Instead, it is also possible to combine entities in a single hardware device.

We introduce the architecture starting from the user side. The User Equipment (UE) is connected via the air interface (Uu interface) to the Base Station (BS) offering the best channel quality. In an LTE network the functionality of BSs is provided by the evolved NodeBs (eNodeBs). A single eNodeB might be responsible for multiple cells[1]. On the data plane the eNodeB is connected to the Serving Gateway (S-GW) via the S1-U interface and on the control plane it is connected to the Mobility Management Entity (MME) via the S1-MME interface. To ensure synchronization between data plane and control plane the S11 interface is available between S-GW and MME. Additionally, there exists the possibility of directly interconnecting eNodeBs via the X2 interface. During normal operation of the network this connection is used to prepare and perform handovers of UEs from one eNodeB to another. In the case of coordination between BSs this connection could also be used to exchange scheduling or channel information.

The S-GW is the mobility anchor for the UEs when moving in the network, i.e., when moving between LTE eNodeBs as well as when moving between other packet data technologies like GSM and UMTS. Additionally, the required measurements and statistics for

[1] In the following, we use the term BS to denote a cell, independently if it is served by its own eNodeB or if the eNodeB is serving multiple cells.

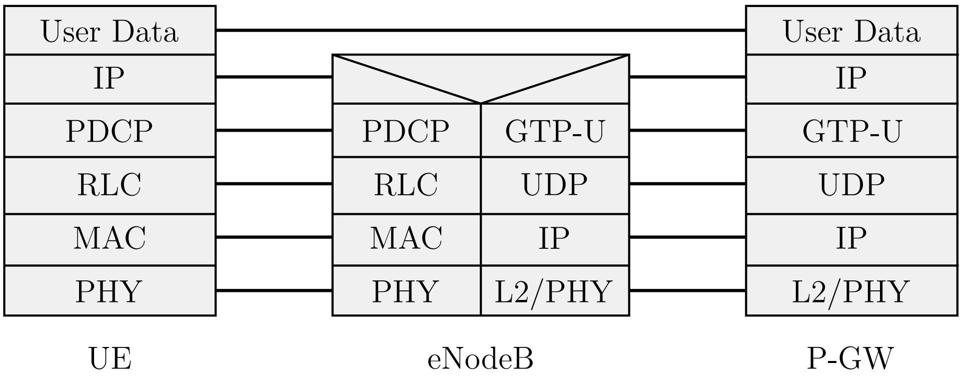


Figure 2.2: LTE data plane protocol layers between UE and P-GW [Ols+09]

charging are collected by the S-GW and forwarded to the Packet Data Network Gateway (P-GW). S-GW and P-GW are connected by the S5 interface. The P-GW connects the EPC to external IP networks, e.g., the Internet. This connection is provided by the SGi interface. Additional functions like IP address assigning, charging and Quality of Service (QoS) enforcement are also performed by the P-GW on a per UE basis.

On the control plane the MME manages mobility and security functions. The setup and teardown of data bearers, the logical channels used for data transmission between UEs and the external network, are also handled by the MME. For all operations the MME depends on a database containing all relevant information about the UEs, like credentials for authentication and authorization. This database is located in the Home Subscriber Server (HSS) and is accessible via the S6a interface.

2.1.2 Protocol Layers

Figure 2.2 shows the protocol stack for the data plane connection between UE and P-GW. The figure does not include the S-GW between eNodeB and P-GW, because from the view of the shown protocol layers it operates transparently. The PHY layer (layer 1 in the Open Systems Interconnection (OSI) model) performs the operations of coding, multi-antenna processing, like beamforming and precoding, and mapping to the LTE radio resources. In LTE, layer 2 of the OSI model is further subdivided into Medium Access Control (MAC), Radio Link Control (RLC) and Packet Data Convergence Protocol (PDCP). The MAC sublayer is responsible for scheduling, also called dynamic resource allocation, handling retransmissions in case of transmission failures and multiplexing of different RLC logical channels. The RLC sublayer performs the tasks of data segmentation and aggregation, duplicate detection and in order delivery of data to the higher layers. Thereby, it makes use of the RLC buffer, which is located between the RLC and PDCP sublayers. The PDCP sublayer is responsible for IP header compression and ciphering. In the case of handover it also detects duplicates and ensures in order delivery of packets.

The interfaces between the different sublayers are provided by physical, transport and logical channels. The most important downlink channels are depicted in Figure 2.3. In the downlink, the physical channels Physical Broadcast Channel (PBCH) and Physical Downlink Shared Channel (PDSCH) perform the mapping of data to the available LTE

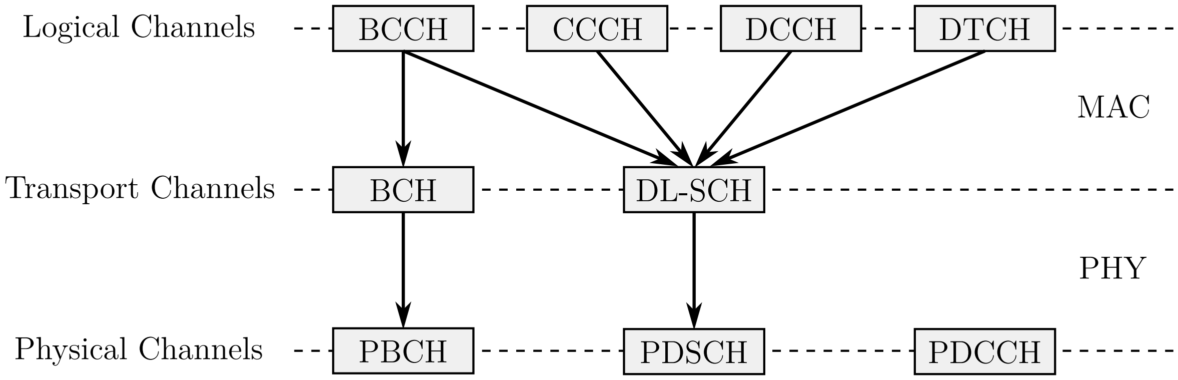


Figure 2.3: LTE downlink channel mapping [DPS16]

radio resources. The decoding of the PDSCH is carried out with additional information provided in the Physical Downlink Control Channel (PDCCH). The mainly used transport channel is the Downlink Shared Channel (DL-SCH), which supports the key features of LTE, like dynamic resource allocation, spatial multiplexing and error handling. Besides the DL-SCH, the Broadcast Channel (BCH) is used to transmit parts of the Broadcast Control Channel (BCCH) logical channel. These parts contain the basic system configuration required by the UEs to detect and register in the system. Further system configuration information is then transmitted via the DL-SCH. In the Common Control Channel (CCCH) and the Dedicated Control Channel (DCCH) additional device specific control information is transmitted. Finally, unicast user payload data is transmitted in the Dedicated Traffic Channel (DTCH). Therefore, the DTCH transfers the highest data volume.

Additional channels for special cases like multicast transmissions or direct device to device communication exist, but as they are not relevant for this thesis, they are not considered here. For the uplink a similar separation into channels exists. However, the total number of channels is smaller.

2.1.3 Radio Resources

The downlink of the LTE radio network uses Orthogonal Frequency Division Multiple Access (OFDMA) as multiple access scheme. This allows to use the three resource dimensions frequency, time and space, as Figure 2.4 shows. For the uplink Single-Carrier Frequency Division Multiple Access (SC-FDMA) is used, which has similar properties. Because in the thesis the downlink is investigated, we only present OFDMA in more detail.

In LTE the downlink resource assignment is handled by the scheduler, which is located in the eNodeB. Dynamic scheduling is performed on a subframe level. The duration of a subframe is 1 ms and is called Transmission Time Interval (TTI). Ten consecutive subframes form a frame. A subframe is further divided into two slots of 0.5 ms. Each slot comprises seven Orthogonal Frequency Division Multiplexing (OFDM) symbols in the time domain.[2] Data is transmitted in Resource Elements (REs). Each RE carries a modulation symbol and is a single subcarrier in a single OFDM symbol. To simplify

[2] Valid if the normal cyclic prefix length is used. For the extended cyclic prefix only 6 OFDM symbols are usable.

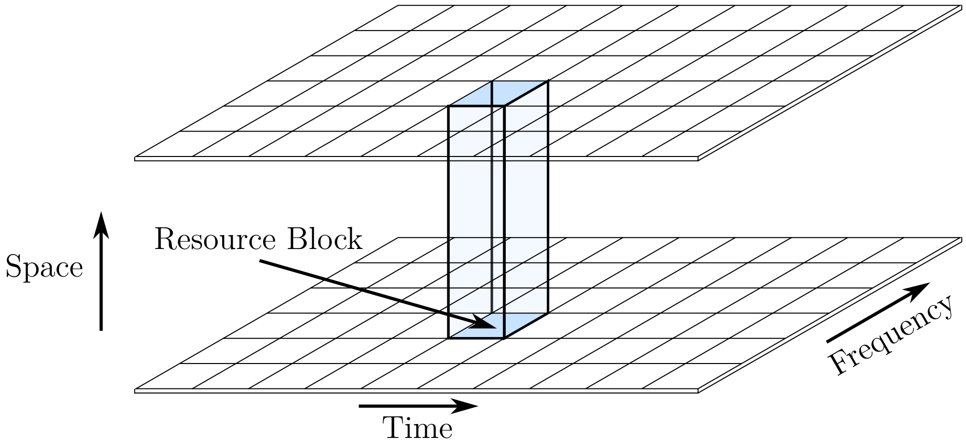


Figure 2.4: LTE resource grid

resource allocation, LTE introduces Resource Blocks (RBs)[3] consisting of twelve successive REs in the frequency domain and seven successive REs in the time domain. A RB is the smallest resource portion the scheduler can assign.[4] The used subcarrier spacing is 15 kHz, so a RB uses a total bandwidth of 180 kHz and has a duration of one slot. Although the principal LTE specification is independent of the total available bandwidth, in practice only a limited set of bandwidths ranging from 1.4 MHz (6 RBs) up to 20 MHz (100 RBs) are used [3GPP 36.104]. Release 10 allows to aggregate up to five 20 MHz carriers, resulting in a total bandwidth of 100 MHz. With release 13 this was extended to 32 aggregated carriers with a total bandwidth of 640 MHz.

The actual data transmission is carried out by one of the specified modulations [3GPP 36.211]. With release 8 either Quadrature Phase Shift Keying (QPSK), 16-QAM or 64-QAM, i.e., 2, 4 or 6 bit per modulation symbol, can be used. Quadrature Amplitude Modulation (QAM) is a modulation scheme that combines amplitude and phase modulation. Release 12 introduced 256-QAM, allowing to transmit 8 bit per modulation symbol. Because transmission over a wireless channel is prone to errors, error recovery mechanisms are necessary. In general, two possibilities to handle this are available. The first is to perform error coding, which adds redundancy information to the transmitted data. The second is to handle errors by retransmitting the erroneous data. LTE uses Hybrid Automatic Repeat Request (HARQ), which combines both possibilities. For Forward Error Correction (FEC) turbo coding with a fixed code rate of ⅓ is employed. During the process of rate matching the fixed code rate is transformed to the required code rate, depending on the current channel conditions. The combination of modulation order and code rate is called Modulation and Coding Scheme (MCS) and is selected according to the channel conditions. The MCS selection is also called link adaptation. How efficient the system uses the available radio channel is described by the spectral efficiency, which defines how much data can be transmitted per time and radio spectrum (bit/s/Hz).

[3] In fact, LTE introduces Virtual Resource Blocks (VRBs) and Physical Resource Blocks (PRBs) to separate the assignment and addressing of RBs from the transmission in the OFDM resource grid. But because this separation is not relevant for this thesis, we use the term RB.

[4] To reduce the signaling overhead, multiple RBs are grouped to a Resource Block Group (RBG) and the scheduler assigns whole RBGs.

By using multiple transmit and receive antennas the spatial domain is exploited and multiple data streams can be transmitted in a single RB. This is called Multiple Input Multiple Output (MIMO). LTE release 8 supports up to four layers of spatial multiplexing in the downlink direction. Release 10 introduces MIMO support with up to four layers in the uplink and increases the number of layers in the downlink to eight [DPS16]. Even if release 8 already supports MU-MIMO, the utilization is inflexible. Dynamic switching between MIMO and MU-MIMO is not possible and when using MU-MIMO only a single stream can be transmitted to a single UE. Release 10 resolves these issues [Dup+11]. More details are discussed in Section 2.5.3.

Uplink and downlink transmissions always have dedicated radio resources, such that they work independently of each other. The separation can be either in the time domain (Time Division Duplex (TDD)) or in the frequency domain (Frequency Division Duplex (FDD)). Because FDD deployments are more widespread in Germany [BNetzA16], we concentrate on FDD operation in the following.

2.1.4 Network Topology

Basic network coverage in LTE networks is provided by the so called macro-BSs. The location of a BS is called site. To increase the network capacity, multiple BSs can be placed on the same site. In this case, each BS serves a sector of the site. In typical deployments a site consists of three sectors. The distance between sites is assumed to be 500 m in the standardization process [3GPP 36.814] and ranges up to 1000 m for average inter-site distances in more realistic deployments [Ros+13].

To improve the network performance in hotspots[5], smaller BSs, often called pico-BSs or micro-BSs, are deployed within the coverage range of a macro-BS. Such a deployment is then called a heterogeneous network. Macro-BSs and pico-BSs implement the same functionality. The main difference is that pico-BSs have limited capabilities, e.g., in terms of transmit power or number of antennas.

Because LTE applies a frequency reuse factor of one, i.e., neighboring BSs use the same radio frequencies, users located on the border between multiple BSs suffer from high interference. As we will see in Section 2.2.2, LTE provides several means to overcome this problem.

2.1.5 Mobility Management

To allow mobility of the UEs, the network must be able to keep track of the UE's position and select the serving eNodeB accordingly. The change from one eNodeB to another is called handover. Depending on the Radio Resource Control (RRC) state of the UE, i.e., `RRC_IDLE` or `RRC_CONNECTED`, two handover possibilities exist in LTE. For more details refer to [Cox14] and the 3GPP standards [3GPP 36.331; 3GPP 36.304]. In the `RRC_IDLE` state the energy consumption of the UE and the signaling load is reduced, as the UE only becomes active to respond to location tracking messages and to perform handover related

[5] Hotspots are areas with higher network load, like city centers, train stations or event areas.

measurements. Additionally, the UE decides itself if a handover is required and if this is the case also performs the necessary actions itself.

The handover in the `RRC_CONNECTED` state is more challenging and is controlled by the network, because ongoing data transmission may not be affected by the handover procedure. In this case the handover is a two step process. In the first step, the serving eNodeB requests the UE to perform measurements of the signal levels of the serving cell and its neighbors. In the second step, the serving eNodeB decides based on the reported measurements if a handover is required and if so performs the necessary actions.

The measurement is initiated by sending an RRC Connection Reconfiguration message from eNodeB to UE. This message contains a list of measurement objects. A measurement object describes an LTE carrier frequency and channel bandwidth. Additionally, the message contains a list of reporting configurations, which describe how and when the UE shall report the measurements back to the serving eNodeB. Finally, the message contains a mapping between each measurement object and one of the specified reporting configurations. The eNodeB can either request measurement reports once or periodically. The reporting period is configurable from 20 ms to 60 min. It can further be distinguished between event triggered and unconditional reports. A trigger for a reporting event could be when the signal level of a neighboring eNodeB is higher than that of the current serving eNodeB for a certain duration. If the serving eNodeB receives such a measurement report, it initiates the handover. How this is performed, is influenced by many factors. E.g., if the serving eNodeB and the neighboring have a direct X2 connection, if the UE would be moved into a new S-GW serving area or if the UE would move into a new MME pool area. For a detailed discussion refer e.g., to [Ols+09, chapter 12.4].

2.1.6 Cloud Radio Access Network (C-RAN) Architecture

The ongoing trend of moving software and application logic into the cloud has not stopped at cellular networks. In the case of cellular networks this is known as Cloud Radio Access Network (C-RAN), which means that the processing, i.e., signal and protocol processing, is moved from the traditional BSs to central processing units located in the cloud. As we will see in Section 2.2.3, this concept has manifold advantages if CoMP is implemented in the network. The C-RAN concept can in principle be applied to any cellular network technology. However, research and development is focused on LTE and future networks. C-RAN was first introduced by Lin et al. [Lin+10] and presented in more detail by the China Mobile Research Institute [CMRI11]. The functionality of the previous BS is split into two parts: the Remote Radio Head (RRH) and the Baseband Unit (BBU). The RRH is still located on-site and consists of antennas, power amplifiers and analog-digital-converters. All processing is performed by the BBUs, which are located in a central office. RRH and BBU are interconnected via the fronthaul network. Due to processing times and propagation delay the maximum distance between RRH and BBU is limited to 15 km to 40 km [Net13; Che+15], such that a metropolitan area can be served by one central office. It is envisioned to use standard processing hardware like off-the-shelf servers, which already provide sufficient processing capacity to perform all necessary tasks [Zhu+11; Aly+15; WP16]. The overall network architecture can be found in Figure 2.5. Even if not depicted in the figure, parts of the EPC can be located in the same central office as the BBU pool.

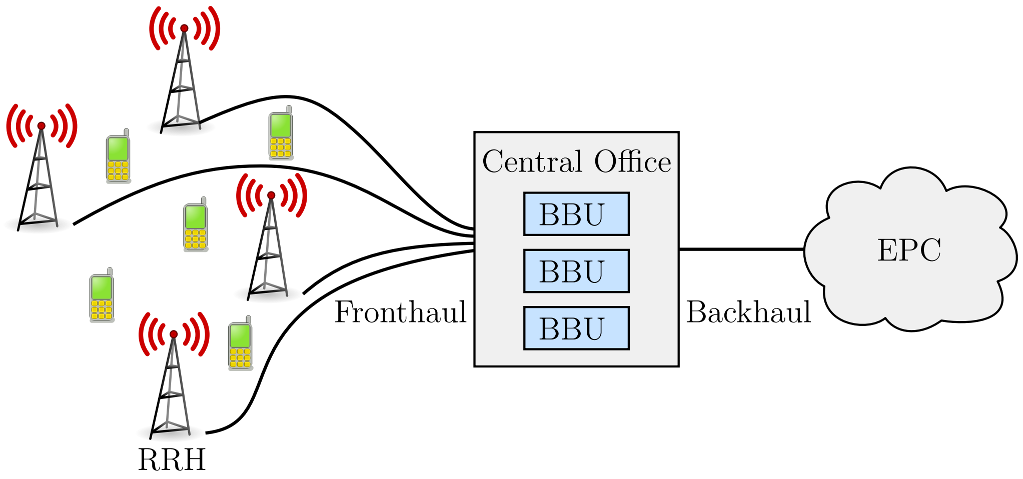


Figure 2.5: C-RAN architecture

The promise of a C-RAN architecture is to reduce the total cost of the network. Both the Capital Expenses (Capex) and Operating Expenses (Opex) can be reduced [Che+15]. The network operation becomes easier, because the network intelligence is not distributed on the field anymore. So maintenance and updates become easier. Also, hardware failures can be mitigated by automatic reconfiguration of the BBU pool in the central office. Furthermore, due to the centralized processing of a larger area in a single BBU pool, statistical multiplexing gains can be achieved. This means that less processing hardware has to be installed in comparison to traditional networks. In previous works, we have shown that a dynamic assignment of processing tasks on the available BBUs further increases the achievable multiplexing gain [Wer+15; SG16].

2.1.7 Evolution to 5G

As the 5G standardization process has just started and first implementations will not be available until the year 2020, we give here only an outlook on how an envisioned 5G system will look like. However, it is commonly accepted that 5G networks will need to support diverse applications and use-cases [MET16]. These new applications can be classified as enhanced Mobile BroadBand (eMBB), Ultra-Reliable and Low Latency Communication (URLLC) and massive Machine Type Communication (mMTC). eMBB covers applications similar to these already handled in today's LTE networks, with the difference that bandwidth requirements and device densities will further increase. eMBB is generally seen as human-centric communication [DPS16]. While LTE release 13 already introduced enhanced support for Machine Type Communication (MTC), the requirements will further increase. E.g., the number of devices will be higher and the energy consumption should be further reduced. The last class of applications is labeled with URLLC, which requires low latencies and high reliability and availability.

To fulfill these requirements 5G networks will need to support peak data rates of 20 Gbit/s, a three times higher spectral efficiency compared to LTE, end-to-end latencies down to 1 ms, a 100 times better energy efficiency and mobility of devices up to 500 km/h [ITU15; DPS16]. Because it is hardly possible to fulfill all requirements at the same time, 5G is not seen as a single RAT, but as a platform of multiple RATs to provide a network for each application that satisfies the application's requirements [DPS16]. This is achieved by introducing the concept of network slices, where each slice provides a virtual network for a specific application class. The network slices share the same physical infrastructure and spectrum.

Another design goal for 5G is backward compatibility to LTE networks. Therefore, the 5G design will be split into two parts. Where possible, the existing LTE standard will be evolved and extended to support parts of the new requirements. Only where this is not possible, new 5G technology will be introduced [DPS16]. This means for the concepts presented in this thesis that they will be still applicable in 5G networks.

2.2 CoMP in Cellular Networks

In general, Coordinated Multi-Point (CoMP) describes that the Transmission Points (TPs), i.e., BSs and UEs, in cellular networks coordinate or cooperate during signal transmission or reception. CoMP can be applied for uplink and downlink transmissions. The degree of cooperation is thereby highly variable.

In this section, we first introduce a classification of CoMP variants (Section 2.2.1), before presenting how CoMP is already supported in LTE networks (Section 2.2.2). Finally, we show how CoMP can be introduced in future C-RAN architectures (Section 2.2.3).

2.2.1 Classification of CoMP Variants

According to Figure 2.6 we can further distinguish between downlink CoMP schemes following the Coordinated Scheduling/Coordinated Beamforming (CS/CB) and Joint Processing (JP) principles [MF11]. The difference between both is that with CS/CB the BSs coordinate their transmissions, but a single UE is still served by only one BS, whereas with JP the BSs actually transmit cooperatively, such that a single UE is potentially served by multiple BSs at the same time. For uplink transmissions a similar distinction is possible. Either the uplink scheduling can be coordinated for different UEs or the received signals can be processed jointly. In general, uplink CoMP schemes do not require as much support from the radio interface as it is necessary for downlink CoMP, because most processing can be performed at the BS side, e.g., by combining the different signals received by multiple BSs. Because the focus of this thesis is on downlink transmissions, we restrict the following discussion on downlink CoMP schemes. Therefore, we use the more general term TP instead of BS to denote the transmitter and UE to denote the receiver.

In CS/CB-based CoMP schemes the TPs try to avoid or reduce interference by coordinating their transmissions. In OFDMA-based networks, there are the three dimensions time, frequency and space available for coordination. Often they are not treated separately, as the resource allocation process also operates on all three dimensions. A UE is served by exactly one TP. As a consequence, the data for this UE only has to be available at the serving TP. In contrast, in JP-based CoMP variants the UE data has to be

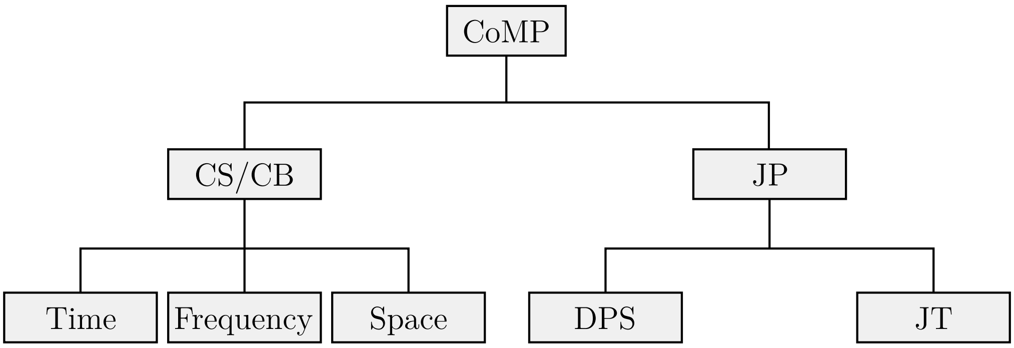


Figure 2.6: Overview of downlink CoMP variants

available at all involved TPs, which is then transmitted to the UE cooperatively. Using Dynamic Point Selection (DPS) the UE is still served by one TP, but the serving TP can change every transmission interval. While this requires in traditional cellular networks to perform a complete handover procedure, with DPS this can be performed immediately, because the UE data and required context information is available at all involved TPs. Joint Transmission (JT) goes even one step further, such that a single UE is served by multiple TPs at the same time. Therefore, the involved TPs jointly transmit the same data on the same physical resources to the UE. As a precondition, the TPs have to precode the transmitted signals so that they interfere constructively at the receiver. The precoding requires detailed channel knowledge at all involved TPs, otherwise the different signals would interfere destructively. A simplified variant of JT is non-coherent JT, where precoding is only performed individually for each TP. Thus, the signals do not interfere perfectly constructively and only the received signal powers sum up. The performance gains are smaller in comparison to coherent JT, but the complexity of joint precoding is avoided.

It is well-known that the performance of CoMP schemes with tighter coordination, i.e., DPS and especially JT, is higher than for CS/CB-based schemes [Lee+12; Maa+12]. Other studies report a performance gain for cell-edge users of 14 % for CS/CB and 35 % for JT in comparison to a non-coordinated network applying MU-MIMO transmissions [Lee+12]. For heterogeneous networks the reported performance improvements are even higher. Maattanen et al. [Maa+12] compare the performance gains of DPS and JT and show that JT offers higher improvements for cell-edge users while the total capacity is better as well. The focus of this thesis is therefore on JT, as it promises the highest performance gains and the issue of high complexity can be resolved by advanced network architectures, e.g., C-RAN deployments.

While the theoretical framework of JT was developed in the early 2000s [Bai+00; SZ01] and field trials showed impressive performance improvements [Irm+11], the higher complexity prevented introduction in commercial networks. So recent work focuses on reducing the complexity of the required signal processing, to make CoMP feasible [Ahm+16]. Before we introduce another possibility to reduce the complexity in Section 2.3, we present the different CoMP variants in more detail and highlight the state of the art in LTE networks.

For the introduction of CoMP in real systems, different design options exist, as Figure 2.7 indicates. First, CoMP can be implemented statically. In the case of CS/CB this means

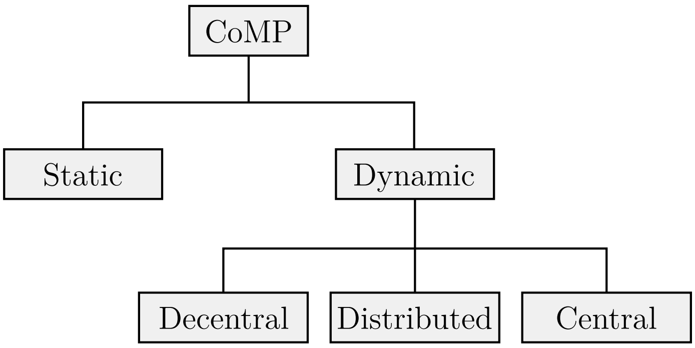


Figure 2.7: Realization possibilities for CoMP

that the shared resources, i.e., time, frequency and spatial resources, are statically assigned to TPs for exclusive use. An example is the frequency reuse method, which was already used in GSM networks [TV05]. At the cost of not using all the available bandwidth in all cells, frequency reuse reduces the interference between neighboring cells. Similar options exist for the utilization of time resources, such that a TP does not use all available time-slots to reduce interference. But the conditions in a cellular network are not static, but changing over time, i.e., users move, traffic loads change and the quality of the radio channel varies. So even if static CoMP schemes are easy to implement, their performance is limited. This is improved by using dynamic CoMP schemes.

For the implementation of dynamic CoMP schemes multiple design options are available [Nec09; MF11]. The first possibility is to introduce a Central Unit (CU) in the system. The CU collects all required information and then decides which TPs shall cooperate in which manner. This option offers the highest performance gains and is easy to implement, because the CoMP decision is based on full system knowledge. Nevertheless, this option also has drawbacks due to the centralization. All information has to be sent from the TPs, where it is collected, to the CU. So latency and bandwidth limitations could prohibit using this scheme in larger networks. Distributed schemes can circumvent these problems. In distributed CoMP schemes several CUs are introduced, each of them responsible for a smaller number of TPs. Because no central entity controls the whole network, the results achieved by distributed schemes are generally inferior in comparison to centralized schemes. Finally, decentral CoMP schemes work without a central entity. The TPs directly exchange the necessary information and agree on a coordination scheme. Similar as for distributed CoMP schemes, the resulting performance is generally inferior.

2.2.2 Standardization of CoMP in LTE

In the following, we give an overview how LTE supports CoMP. Similar to the scheduling and resource assignment, the standards do not specify any algorithms or implementations for CoMP. They only specify protocols and messages that can be used to introduce CoMP. The actual implementation is left to the vendors. Additionally, vendor specific extensions are possible, which are not considered here.

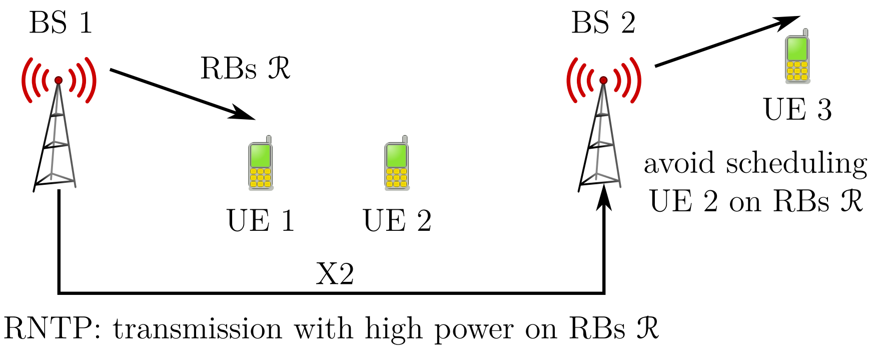

BS 1
RBs $\mathcal{R}$
BS 2
UE 3
avoid scheduling UE 2 on RBs $\mathcal{R}$
UE 1
UE 2
X2
RNTP: transmission with high power on RBs $\mathcal{R}$

Figure 2.8: Application of RNTP signaling

2.2.2.1 Release 8

The focus in the first release of LTE was on relatively loose coordination. Therefore, three X2 signaling messages have been defined to allow CS/CB [3GPP 36.423a]. In 3GPP terminology this is known as Inter-Cell Interference Coordination (ICIC). However, the standards only specify the messages, but not how the eNodeBs should react to them.

For downlink coordination the Relative Narrowband Transmit Power (RNTP) message was introduced, which provides a way to indicate for each RB individually if the transmit power will exceed a certain threshold. Figure 2.8 shows a possible application of RNTP signaling. In this example BS 1 intends to serve UE 1 on a set of RBs $\mathcal{R}$. Simultaneously BS 1 sends a RNTP message to neighboring BSs over the X2 interface. This message contains the information that the transmit power on RBs $\mathcal{R}$ will be high. Using this information, BS 2 can avoid scheduling UE 2 on the same RBs to avoid interference and instead schedule UE 3 on these resources.

For uplink coordination the High Interference Indicator (HII) and Overload Indicator (OI) messages have been standardized. The HII message contains information about which set of RBs is sensitive to interference caused by neighboring cells. The OI message contains information about received interference on each RB based on three levels (low, medium, high). It would be logical for an eNodeB to avoid scheduling on RBs contained in an HII message. Especially cell-edge UEs should not be scheduled, to reduce the interference for the eNodeB issuing the HII. Therefore, HII messages provide a proactive possibility to reduce interference. In contrast, eNodeBs send OI messages if interference has already been caused. As a reaction, the eNodeB receiving an OI message could reduce the scheduling activity on the RBs included in the message to reduce interference in the future.

2.2.2.2 Release 10

While the coordination in release 8 focused on the frequency domain and release 9 did not introduce new CoMP features, with release 10 also the time domain is used for coordination, which is then called enhanced Inter-Cell Interference Coordination (eICIC) by 3GPP. Still, only CoMP variants of the class of CS/CB are supported. Therefore, the concept of Almost Blank Subframes (ABS) was introduced, which is especially beneficial

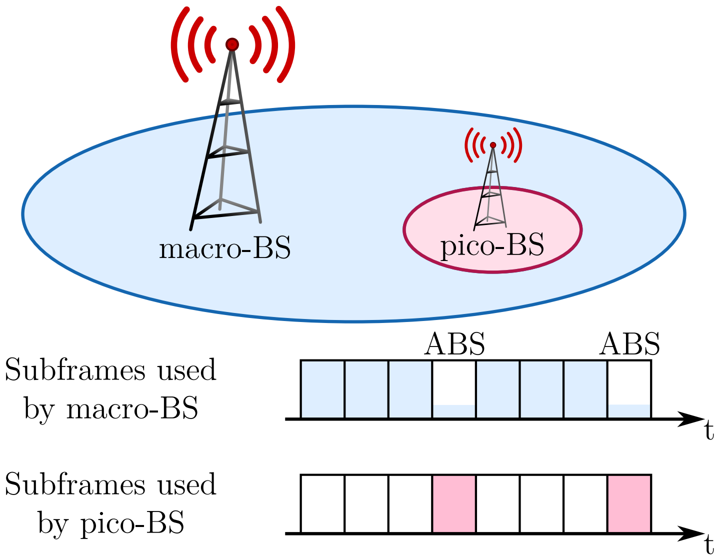


Figure 2.9: Principle of ABS

for coordination between a macro-BS and pico-BSs located within the coverage area of the macro-BS [3GPP 36.300b].

Figure 2.9 shows the principle of ABS, with one pico-BS within the coverage area of a macro-BS. As the lower part of the figure indicates, the macro-BS does not use all available subframes. Instead, it reduces the transmit power or does not use certain subframes for data transmission at all. These subframes are called Almost Blank Subframes (ABS). To ensure backward compatibility, the control channels have to be transmitted in all subframes including ABS. Because the transmit power of pico-BSs is lower than that of macro-BSs, the performance of UEs served by pico-BSs is severely impaired by the high interference received from the macro-BS. Using ABS the interference in selected subframes is reduced. The ratio between normal subframes and ABS is configurable. Pico-BSs may use all available subframes, regardless whether they are used by the macro-BS or not. However, this is not shown in Figure 2.9.

2.2.2.3 Release 11

With release 11 the activities regarding CoMP are twofold. On the one hand the work on eICIC was finished, which is then called further enhanced Inter-Cell Interference Coordination (feICIC) and on the other hand features to support JP have been introduced [3GPP 36.819]. The basis for both are enhancements in the radio-interface, including improved Channel State Information (CSI) acquisition and signaling as well as a refined reference symbol structure, which is described in Section 2.6.2. In release 11, UEs can detect interference better, because the cell ID and the ABS configuration of each interfering TP are signaled to the UEs. The possible options for JP are DPS and JT. Because an ideal backhaul, i.e., a high bandwidth and low latency network, is assumed, a central component carrying out the JP decisions is required. However, this central component is not standardized.

2.2.2.4 Release 12

While the focus of JP in previous releases was on centralized variants assuming an ideal backhaul, release 12 introduced features supporting decentralized JP schemes. E.g., the X2 application protocol was extended to allow the exchange of Reference Signal Received Power (RSRP) measurement reports from individual UEs between eNodeBs [3GPP 36.423b]. The interval between periodic RSRP updates is configurable between 120 ms and 640 ms in steps of 120 ms. Additionally, the exchange of CoMP hypothesis was introduced in the X2 application protocol. A CoMP hypothesis includes a hypothetical resource allocation in combination with an associated cost or benefit value. In comparison to the messages introduced in release 8 (HII, OI and RNTP) this allows a more fine-grained negotiation of the used CoMP scheme. As in the previous releases only the messages are standardized, but not the reaction of an eNodeB to these messages.

2.2.2.5 Release 13 and beyond

In release 13 the work of the previous release was continued. Release 13 allows the exchange of detailed Channel State Information (CSI) of individual UEs between eNodeBs. The availability of detailed channel information from multiple eNodeBs is a requirement for the introduction of JP-based CoMP schemes. This enables a more fine-grained information exchange in comparison to only exchanging RSRP information.

Besides that a new study item was started, in which self organizing concepts for CoMP are evaluated [3GPP 36.742]. Especially the fact that dynamic effects, e.g., traffic load variations or user movement, play an important role for the benefit of CoMP, was not considered in previous releases, because the set of eNodeBs that exchange information over the X2 application protocol had to be configured by the operator. Self organizing concepts enable the network to determine the most beneficial set of cooperating eNodeBs by itself. In research this is a well-known problem and is discussed in Section 2.3.

For release 14 new studies regarding multiple realization options of CoMP have been conducted [3GPP 36.741]. This includes CS/CB and non-coherent JT in the scenarios "Macro Cell Deployment", "Indoor Small Cells" and "Heterogenous Urban". The study concludes that non-coherent JT provides average performance gains of approximately 26 %, while CS/CB results in average improvements of approximately 18 %. Additionally, the gains depend on the network load. Higher loads result in improved performance of CS/CB, while the non-coherent JT performance decreases with higher load.

2.2.3 CoMP in C-RAN Architectures

The combination of C-RAN and CoMP is a promising approach to solve the challenges related to CoMP [Wan+09; Liu+15]. Within the central office the necessary synchronization and data exchange becomes possible, because the latency is low and the available bandwidth is high. However, new challenges are introduced, as the fronthaul capacity between BBUs and RRHs is still constrained. Newer research also considers the processing capacity and load balancing effects within the cloud when CoMP is introduced in the cellular network [TP15]. Furthermore, C-RAN architectures facilitate introducing concepts of Software-Defined Networking (SDN) in cellular networks, which allow flexible network orchestration and function placement in the cloud environment [Tan+17]. The necessary changes and new components to support CoMP can then be implemented as cloud software functions.

2.3 CoMP Clustering

One way to reduce the complexity of signal processing and channel measurement necessary for CoMP is the introduction of clusters. A cluster is defined as a set of TPs between which coordination is allowed. The main source of interference is therefore caused by TPs of neighboring clusters, which is called inter-cluster interference. Intra-cluster interference, i.e., interference between TPs belonging to the same cluster, is minimized by one of the aforementioned CoMP techniques. In several studies the need of clustering to make CoMP feasible has been highlighted [PGH08; Bas+17]. In the following, we use the term clustering for the set of clusters configured in the system and define the size of a cluster by the number of TPs within the cluster. The symbols $\mathcal{C}$, C and S_C are used for clustering, cluster and cluster size, respectively.

Finding a good clustering however is no trivial task, because the number of possible clusters scales exponentially with the number of TPs N_{TP} within the network. More precisely the number of clusters is $2^{N_{\text{TP}}} - 1$. Even if the maximum cluster size is limited to $S_{C,\max}$, the number of possible clusters is still $\sum_{s=1}^{S_{C,\max}} \binom{N_{\text{TP}}}{s}$. For more details refer to Appendix A.1. Therefore, determining the optimal clustering by exhaustive search is not suitable in real-world networks, because the process must complete in reasonable time. Additionally, the complexity should be reduced by introducing clusters instead of increasing the complexity by solving difficult optimization problems.

If the clustering is static, i.e., the clusters are only configured once, the large number of possible clusters is not an issue, as the cluster definition could be included in the network planning phase. Even if Davydov et al. [Dav+13] show that large static clusters can improve the network performance, dynamic clustering promises higher performance gains [PGH08]. The reasons are obvious, because the network conditions are also not static. Users move in the cellular network, traffic demands follow diurnal and short-term patterns and the conditions of the radio channel change over time.

For this reason, in the literature several approaches to determine and configure dynamic clusterings have been proposed (see Table 2.1). Mainly they can be grouped into two categories. The first group treats clustering as an optimization problem and uses the methods of mathematical programming to solve it. The second group tries to solve the problem heuristically with the help of algorithms. Both techniques are presented in more detail in Section 2.3.4.

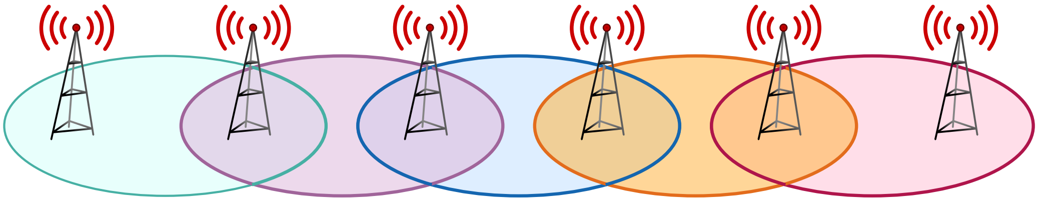


Figure 2.10: Overlapping clusters

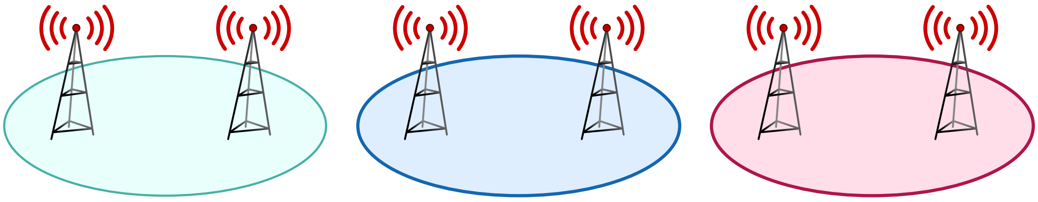


Figure 2.11: Non-overlapping clusters

2.3.1 Definition of Clusters

Before we actually present methods to generate clusterings, we further elaborate on two different cluster definitions. It can either be allowed that clusters overlap each other, as shown in Figure 2.10, or they can be non-overlapping (disjoint of each other), as in Figure 2.11. In this context overlapping means that at least one TP is included in more than one cluster. Overlapping clusters have the great advantage that for each UE, or at least for a greater number of UEs, a good cluster can be used for data transmission. Thereby, the consequences of inter-cluster interference are reduced. There exist several possibilities to define overlapping clusters. E.g., they could be defined such that a TP belongs to different clusters for different system resources, as shown in Figure 2.12, as proposed by Ramprashad and Caire [RC09] and Marsch [Mar10]. The benefit is that UEs that are on a cluster border on one frequency resource are in the middle of a cluster on another frequency resource. Therefore, they do not suffer from inter-cluster interference if they are served by the appropriate frequency resources. Here similar problems as for traditional frequency reuse occur, which have to be resolved during the creation of the clustering. Especially in heterogeneous networks where clusters of different sizes are desirable, the task of assigning clusters on disjoint frequency resources is challenging. Additionally, the scheduling is more complex, too. The reason is that UEs should ideally be served by clusters where they are in the center to increase the performance. So instead of performing the scheduling per TP, as in the case of networks without CoMP, or per cluster, as in the case of networks with non-overlapping clusters, now the scheduling has to be performed network-wide. However, recent works have proposed methods to perform scheduling and precoding for overlapping clusters without significantly increasing the complexity [DY14; KK16],

Even if non-overlapping clusters suffer from inter-cluster interference, from an implementation viewpoint they are appealing, because defining the clustering and scheduling is easier. Frequencies do not have to be assigned to individual clusters, because a frequency reuse of one can be applied, like in LTE networks. Because each UE belongs to exactly one cluster the scheduling is performed on a per cluster level, which means that one scheduler per

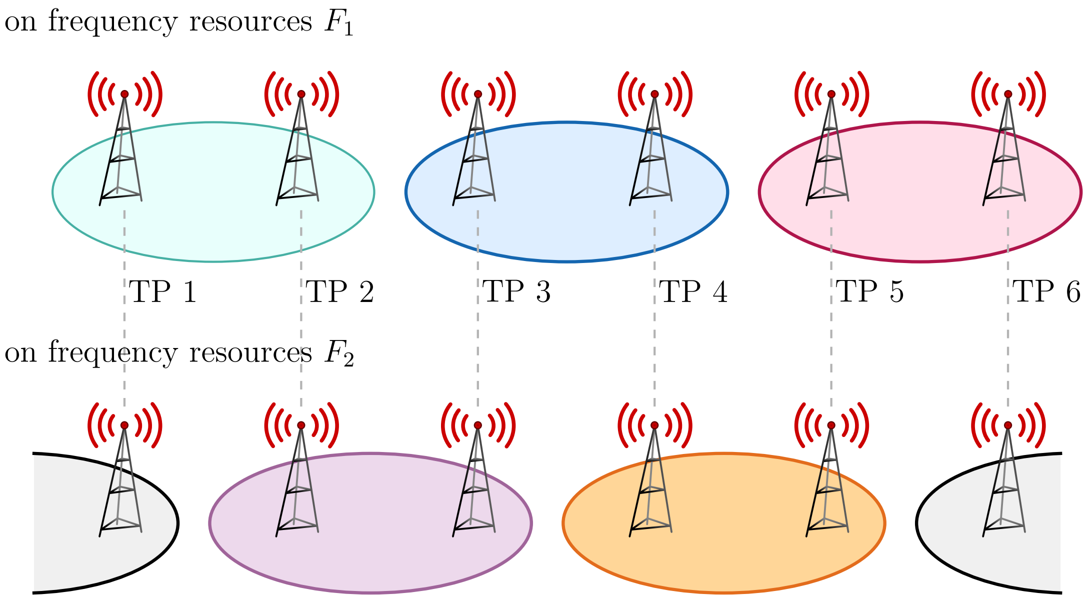


Figure 2.12: Overlapping clusters with frequency reuse

cluster exists. For this purpose existing scheduling approaches as known for single cells can be reused.

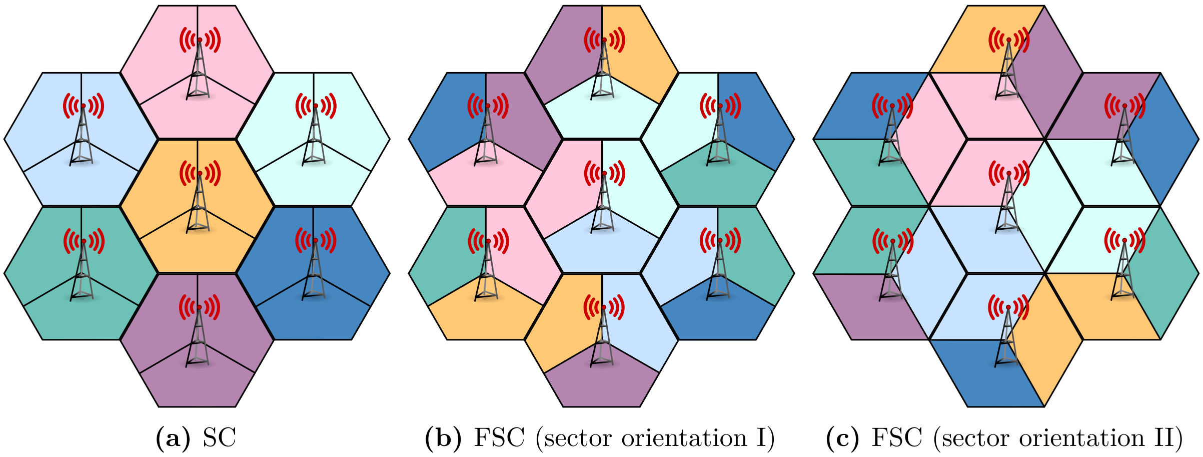
In the following, we address overlapping clustering only briefly in Section 2.3.2. For the remaining part we use either static or dynamic non-overlapping clusters as introduced in Sections 2.3.3 and 2.3.4.

2.3.2 Ideal Clustering

Ideal Clustering is a variant of overlapping clustering. However, it cannot be implemented in real systems, but it serves to compare static or dynamic clustering variants [MF11]. Ideal Clustering means that each UE is served by its best cluster, where the maximum size of the cluster $S_{C,\max}$ is configurable. More formally this means that for a UE u the list of all TPs $\mathcal{B}$ is sorted according to the channel attenuation $h_{u,b}$ in ascending order, i.e., in the sorted list $[b_1, b_2, \ldots, b_{N_{\mathrm{TP}}}]$ TP b_1 has the lowest attenuation. Then the cluster for UE u is defined as $C_u = \left[b_1, b_2, \ldots, b_{S_{C,\max}}\right]$.

2.3.3 Static Clustering

Because of their simplicity, static clustering variants have gained much interest in CoMP research [BH07; Mar10; AS12; MF11; Dav+13]. Illustrations of commonly discussed variants for hexagonal cell layouts are shown in Figure 2.13, but of course static clustering can be applied in general cell layouts, too. Sectors with the same color belong to one cluster. The Site Clustering (SC) variant promises low additional demand on the backhaul network, because only the TPs serving the sectors of a site coordinate their transmissions. The drawback is that the maximum number of coordinated TPs is limited, in common cases to three. On the other hand, implementation becomes feasible as the necessary



(a) SC **(b)** FSC (sector orientation I) **(c)** FSC (sector orientation II)

Figure 2.13: Static clustering for hexagonal cell layouts

information exchange within the same site could be handled easier. The performance can be improved for UEs on the sector border. UEs close the site border do not benefit as much, because interference from the neighboring site is at a similar level as the signal power.

Two variants of inter-site clustering are shown in Figures 2.13b and 2.13c. In the figures wrap-around is assumed, such that the scenario is repeated in all directions. The difference between the two variants is the orientation of the sectors. In Figure 2.13b, the sectors of different sites are aligned such that they are not directly facing each other, which generally reduces the interference caused in the neighboring sites. However, this might impair the performance of CoMP. Ali and Saxena [AS12] propose to resolve this issue by rotating the sector orientation as depicted in Figure 2.13c. We denote both variants as Facing Sectors Clustering (FSC). For both variants the expected performance improvement for UEs on the site border is higher than for SC. However, the requirements for the network interconnecting the sites are higher, because information between different sites has to be exchanged.

2.3.4 Dynamic Clustering

Dynamic clustering determines the best suitable clustering by taking the current situation into account. Besides pure dynamic variants also semi-dynamic clustering schemes exist, which combine static with dynamic clustering [Bas+17]. In semi-dynamic clustering schemes sets of static clusters are defined first. During runtime the system selects the best suitable clustering from the set of static clusters. Thereby, the system performance is improved in comparison to pure static clustering, although similar drawbacks as for pure dynamic clustering exist.

In the following, we present a classification of existing dynamic clustering schemes in Section 2.3.4.1. Then we introduce in Section 2.3.4.2 methods to determine a dynamic clustering. Finally, we present in Section 2.3.4.3 selected dynamic clustering algorithms.

Table 2.1: Literature overview of dynamic clustering approaches

Reference	Year	Degree of Distribution	Solution Method	Type of Clusters	System Integration
Papadogiannis, Gesbert, and Hardouin [PGH08]	2008	central	heuristic	non-overlapping	precoding
Liu and Wang [LW09]	2009	central	heuristic	non-overlapping	isolated, uplink transmission
Zhou et al. [Zho+09]	2009	decentral	heuristic	non-overlapping	isolated
Gong et al. [Gon+11]	2011	central	heuristic, based on cluster candidates	overlapping	scheduling
Sohn, Lee, and Andrews [SLA11]	2011	distributed	message passing algorithm, problem represented as graphical model	non-overlapping	precoding
Weber et al. [Web+11]	2011	central	heuristic, based on cluster candidates	non-overlapping	isolated
Baracca, Boccardi, and Braun [BBB12]	2012	central	optimization problem shown, heuristic solution, based on cluster candidates	non-overlapping	scheduling
Giovanidis, Krolikowski, and Brueck [GKB12]	2012	central	optimization problem, solved by graph theory	non-overlapping	isolated
Zhao et al. [Zha+12]	2012	central	heuristic	overlapping	isolated
Hong et al. [Hon+13]	2013	central	optimization problem shown, heuristic solution	overlapping	precoding
Lee et al. [Lee+13]	2013	central	for simplified assumptions optimal solutions, upper bounds for more complex scenarios	overlapping	precoding
Moon and Cho [MC13]	2013	central	heuristic, based on cluster candidates	non-overlapping	isolated

Reference	Year	Degree of Distribution	Solution Method	Type of Clusters	System Integration
Baracca, Boccardi, and Benvenuto [BBB14]	2014	central	optimization problem shown, heuristic solution, based on cluster candidates	non-overlapping	scheduling
Dai and Yu [DY14]	2014	central	optimization problem shown, solved by iterative approximations	overlapping	scheduling and precoding
Guidolin, Badia, and Zorzi [GBZ14]	2014	distributed	game theory	non-overlapping	isolated
Li, Zhang, and Zeng [LZZ14]	2014	central	graph theory	non-overlapping	isolated
Rahman et al. [Rah+15]	2015	central	heuristic	non-overlapping	precoding
Tran and Pompili [TP15]	2015	central	optimization problem shown, solved by iterative approximations	non-overlapping	scheduling and precoding
Liu et al. [Liu+15]	2015	central	optimization problem for limited scenario, heuristic for larger scale, based on cluster candidates	both considered	precoding
Zhang et al. [Zha+15]	2015	central	heuristic	non-overlapping	isolated
Bassoy et al. [Bas+16]	2016	central	heuristic, based on cluster candidates	overlapping	isolated
Beylerian and Ohtsuki [BO16]	2016	central	heuristic, based on cluster candidates	overlapping	scheduling
Brandt, Mochaourab, and Bengtsson [BMB16]	2016	distributed	game theory	non-overlapping	precoding
Kang and Kim [KK16]	2016	central	heuristic, based on cluster candidates	overlapping	scheduling and precoding
Park, Lee, and Heath [PLH16]	2016	central	graph theory	non-overlapping	isolated

Reference	Year	Degree of Distribution	Solution Method	Type of Clusters	System Integration
Seno et al. [Sen+16]	2016	central	heuristic	non-overlapping	isolated
Zhang et al. [Zha+16]	2016	central	optimization problem	non-overlapping	embedded in C-RAN, considering processing and propagation delays
Haddadi and Ghasemi [HG17]	2017	central	heuristic, based on cluster candidates	overlapping and non-overlapping	isolated
Karavolos et al. [Kar+17]	2017	central	heuristic, based on cluster candidates	non-overlapping	isolated (DPS-CoMP support only)
Kinoshita et al. [Kin+17]	2017	central	heuristic	non-overlapping	isolated

2.3.4.1 Classification of Dynamic Clustering Approaches

In this section, we present a taxonomy of dynamic clustering variants. For this classification we focus on functional aspects of the approaches. Another taxonomy based on the objective of the clustering has been published by Bassoy et al. [Bas+17].

The categories of the functional classification are not exclusive and often the approaches do not fit perfectly into the categories, e.g., a clustering variant could generate partly overlapping clusters and partly disjoint clusters. Table 2.1 shows a non-exhaustive overview of literature related to dynamic clustering. The categories used in the classification are described in the following.

Degree of Distribution

In a central clustering approach a Central Unit (CU) is responsible to determine and configure the clusters, whereas for decentral approaches the clustering is negotiated between the TPs directly. Distributed approaches are special cases of centralized approaches, because in a distributed clustering scheme several CUs exist. Each of the CUs is responsible for a subset of the TPs.

Type of Generated Clusters

In the beginning of this section we already discussed the different types of clusters, namely overlapping and non-overlapping clusters. The clustering approach can be either one of the possibilities or even a combination.

Solution Method

As argued before, the number of possible clusters increases quickly, so finding the optimal clustering for a given situation is a complex task and not suitable for real networks. On the other hand, a heuristic implemented as an algorithm might find a solution in feasible time, but the solution might be non-optimal. Section 2.3.4.2 presents a more comprehensive discussion on methods to dynamically determine a clustering. We have identified methods based on optimization, graph theory, game theory and heuristics. If a heuristic is based on cluster candidates, which are either directly proposed and reported by the UEs or are derived from measurements signaled by the UEs, this is also indicated in Table 2.1.

System Integration

This category determines whether the clustering approach operates integrated into the other system functionalities or if it is working isolated. Here isolated means that the clustering is performed without any interaction with or knowledge from other system functionalities like scheduling or precoding. On the opposite, integrated approaches may interact or exchange information. This could result in better performance, because a global optimal solution for the independent problems can be found. However, the complexity to obtain a global optimum is often not bearable.

2.3.4.2 Methods to Obtain Dynamic Clusterings

Because an exhaustive search is not a viable approach due to the large number of combinations, other ways to determine a clustering have been presented in the literature. Additionally, the clustering process should be considered in combination with precoding and scheduling to achieve better results. Precoding and scheduling are also non-trivial problems, such that finding the optimal solution for realistic scenarios is hardly possible.

Optimization

The task of cluster formation can be described as an optimization problem, as shown in Equation (2.1) [DY14; TP15]. The goal of the problem is to maximize the weighted sum rate of the system. This maximization is subject to additional constraints. Some example constraints are shown in Equation (2.1). The maximization could either be done every scheduling interval or on a longer timescale. While solving the optimization problem every scheduling interval could lead to a different clustering per interval, solving it on a longer timescale requires knowledge of future rates of the UEs. Frequent cluster reconfigurations might not be possible in real systems and knowledge of future rates is not available. Therefore, both options have drawbacks if used in real networks. In Equation (2.1), α_u denotes the weight and r_u the rate of UE u. To simplify the notation the time index is omitted. The UE weights α_u allow implementing different scheduling metrics, by updating the weights according to the scheduling priority.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{u \in \mathfrak{U}} \alpha_u \cdot r_u && \text{(2.1a)} \\
\text{subject to} \quad & \text{clustering constraints} && \text{(2.1b)} \\
& \text{scheduling constraints} && \text{(2.1c)} \\
& \text{transmit power constraints} && \text{(2.1d)} \\
& \text{backhaul constraints} && \text{(2.1e)}
\end{aligned}
$$

This optimization problem belongs to the well-known category of non-convex and Nondeterministic Polynomial Time (NP)-hard Weighted Sum-Rate Maximization (WSRM) problems [LZ08]. Finding the optimal solution is therefore non-trivial. A sub-optimal algorithm for the combined linear precoding and dynamic clustering problem that solves the WSRM problem has been presented by Tran et al. [Tra+12]. To do so the original problem is formulated as a Second Order Cone Program (SOCP), which is then solved iteratively. During the iterative process, the non-convex problem is approximated with a convex one, which can be solved efficiently. The authors also show that this approach converges to a local optimum of the original problem. Global optimality cannot be guaranteed however.

Based on this, Tran and Pompili [TP15] and Dai and Yu [DY14] introduce dynamic clustering variants that also consider compute capacity constraints of a C-RAN architecture and backhaul constraints, respectively. Again local optima are obtained, but global optimality cannot be guaranteed. The problems are solved by combining the SOCP approach with the ℓ_1-norm reweighting approximation. Originally proposed in the context of compressive sensing[6] [CWB08], the ℓ_1-norm reweighting approximation allows to approximate a non-convex ℓ_0-norm by its ℓ_1-norm, which is convex, i.e.,

$$||\mathbf{x}||_0 \approx \sum_i \beta_i \cdot |x_i| \tag{2.2}$$

The ℓ_0-norm counts the non-negative entries of a vector $\mathbf{x}$. For the approximation the weights β_i for the vector components x_i have to be adjusted iteratively.

We can conclude that in theory the dynamic clustering problem can be treated as optimization problem. However, due to the high complexity, finding an optimal solution is hardly possible even for small networks. Real networks are larger and have the additional requirement that the clustering process must be integrated in the normal operation, which requires to determine the clustering in reasonable time. The proposed approximations to simplify the optimization problem allow solving larger problems, but then finding the global optimum is not guaranteed anymore.

Game Theory

Based on the mathematical foundation of game theory on cooperation games (or coalition games), several proposals have been made to use this framework for tasks found in cellular networks. The basis of coalition games is a utility function $u(C)$ for a coalition C. The utility function reflects the benefits and costs of creating coalitions. Making use of the utility function the following two rules for the creation of coalitions are formulated:

Merge Rule Coalitions C_i and C_j are merged, if $u(C_i) + u(C_j) < u(C_i \cup C_j)$

Split Rule Coalition $C = \bigcup_{i=1}^{k} C_i$ is split into coalitions $C_1, \ldots, C_k$, if $\sum_{i=1}^{k} u(C_i) > u(C)$

Two coalitions C_i and C_j are merged if the utility of the merged coalition $C_i \cup C_j$ is higher than the sum utility of the two individual coalitions. On the other hand, a coalition C is split into subcoalitions C_i, if the sum utility of the subcoalitions is larger than the utility of the original coalition.

Saad et al. [Saa+09] introduce a spectrum sharing coalition game, where users dynamically form coalitions to cooperatively transmit data in the uplink direction. This coalition formation is performed by a distributed algorithm executed by each user, which performs coalition merging and splitting. The algorithm aims for maximizing a utility function that represents the achievable capacity. Khan et al. [Kha+11] also formulate a dynamic coalition formation game for cooperative spectrum sharing.

[6] A signal processing technique to reconstruct a signal.

Guidolin, Badia, and Zorzi [GBZ14] and Brandt, Mochaourab, and Bengtsson [BMB16] deal with the TP clustering problem for downlink CoMP. Here a coalition represents a cluster of TPs. Guidolin, Badia, and Zorzi [GBZ14] formulate a central and a distributed cluster formation algorithm using similar coalition merge and split operations as in [Saa+09]. The distributed algorithm is executed by the TPs of a cellular network. Brandt, Mochaourab, and Bengtsson [BMB16] combine a distributed game theoretic coalition formation algorithm with precoding techniques. While the coalition formation algorithm operates on a longer timescale, as the average channel quality does not change rapidly, the precoding is performed on a short-term scale. Additionally, the precoding algorithm takes the interference caused by neighboring coalitions into account.

Game theoretic approaches are well suited to solve the dynamic clustering problem. However, their integration into the network operation and the performance of the generated solutions still has to be assessed.

Graph Theory

The interference relation between TPs and UEs can be described as a graph, where the vertexes of the graph represent the UEs and the weighted edges stand for the interference relation between two UEs [Nec09]. If two UEs have a stronger interference relation, they should not be served on the same resources and applying CoMP would be beneficial. An example graph including a possible clustering is depicted in Figure 2.14, where the width of the edges indicates the interference relation. Not all interference relations are shown in this figure. For the generation of the graph and the representation of the interference multiple options are available. By partitioning the interference graph into multiple unconnected sub graphs, a clustering can be obtained. E.g., Qin and Tian [QT14] present a greedy algorithm that splits the graph at the edges with the lowest weights. In this way TPs that generate high interference to each other form a cluster. Li, Zhang, and Zeng [LZZ14] show a similar approach for arbitrary cell layouts. In their approach two types of vertexes exist, which represent TPs and UEs. The edges represent the channel between TPs and UEs. Using the K-means clustering method[7] the CoMP clusters are determined. Park, Lee, and Heath [PLH16] propose a solution for the dynamic clustering problem for irregular cell layouts. However, their solution is limited to clusters consisting of only two TPs.

Applying graph theory to determine the CoMP clustering is an appealing approach and it has been shown that graph theory is well suited to solve interference coordination problems. However, the construction of the network graph might require complete system information, which comes with additional signaling and channel measurement overhead in real systems. Also, the applied solution algorithms are either NP-hard or not guaranteed to find the globally optimal solution.

[7] The K-means clustering method is popular in machine learning and data mining. K-means clustering maps a set of observations or data samples into k clusters. Each sample belongs to the cluster with the minimal distance between sample and cluster center. K-means clustering is a NP-hard problem. However, efficient heuristics exist that converge quickly to local optima.

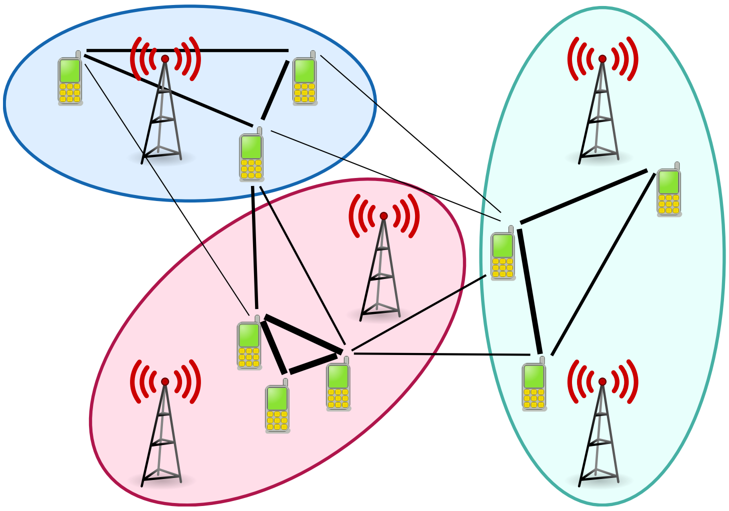


Figure 2.14: Illustration of an interference graph

Heuristic

As finding a globally optimal clustering is not possible in practice, several approaches to determine the clustering heuristically have been presented in the literature. Often the clustering problem is stated as a mathematical problem that is then solved by greedy algorithms, i.e., algorithms that find local optima [BBB14; Gon+11; Zho+09; MF11]. Three heuristic clustering approaches are presented in Section 2.3.4.3 in more detail.

Although finding globally optimal solutions with heuristics is not guaranteed, heuristics are well suited for the introduction in cellular networks. First, by carefully designing the solution algorithm the runtime can be limited, which is a requirement in real systems. By utilizing system knowledge, the complexity can be further reduced and better solutions can be found. E.g., this could include the locations of the TPs. Additionally, by using heuristics the characteristics of game or graph theoretic solutions can be replicated. Finally, heuristic approaches are common in cellular networks, e.g., for scheduling or precoding, such that introducing a heuristic for dynamic clustering is a straightforward approach.

2.3.4.3 Selected Dynamic Clustering Algorithms

In the following, we present an overview of selected dynamic clustering algorithms, which are similar to the one developed in Chapter 4. The notation of the presented algorithms is adapted to match the notation we use throughout this thesis as close as reasonable.

Algorithm I by Baracca et al.

Baracca et al. introduced a dynamic clustering algorithm that can be used in combination with any possible scheduling and precoding scheme [BBB12]. They evaluated the algorithm in combination with Proportional Fair (PF) scheduling and Zero Forcing (ZF) precoding. The basic idea of the algorithm is to combine clustering and scheduling and to use cluster candidates from the UEs. Every scheduling interval a new clustering, consisting of a subset of the cluster candidates, is selected and the UEs increasing the weighted system sum rate are scheduled. The cluster candidates are created based on the Signal to Noise Ratio (SNR) or in other words on the channel attenuation $h_{u,b}$ between TP b and UE u. So UE u proposes the set of TPs $\mathcal{B}_u$ as a cluster candidate:

$$\mathcal{B}_u = [b \in \mathcal{B} | h_{u,b} \geqq h_{u,b,\text{TH}}] \tag{2.3}$$

$\mathcal{B}$ is the set of all TPs in the system, $h_{u,b}$ is the channel attenuation between TP b and UE u and $h_{u,b,\text{TH}}$ is a configurable threshold. $h_{u,b,\text{TH}}$ can be either given as an absolute threshold or relative in comparison to the attenuation between UE u and its anchor TP[8] b_u. This choice can lead to different sizes of the cluster candidates, what has to be taken into account in the next steps. The cluster candidates are then collected in a Central Unit (CU), which is responsible for cluster creation and configuration. The set of cluster candidates $\mathcal{C} = [\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{N_\text{UE}}]$ is created by the CU with N_UE representing the number of UEs in the system. Because usually the same cluster is proposed by multiple UEs, the cardinality of $\mathcal{C}$ is in many cases smaller than the total number of UEs in the system ($|\mathcal{C}| \leq N_\text{UE}$). The remaining cluster candidates in $\mathcal{C}$ are therefore $C_1, C_2, \ldots$. The set $\mathcal{U}_C$ represents the UEs that can be scheduled in cluster C. In the original work, three different configurations are presented, however we stick to the case where $\mathcal{U}_C = [u \in \mathcal{U} | b_u \in \mathcal{C}_C]$. This means each UE is assigned to the cluster candidate that contains its anchor TP.

The task of the CU is to find for each scheduling interval the clustering that maximizes the weighted system sum rate. For sake of simplicity we neglect the time index t in the following. Therefore, the CU determines for each cluster candidate the subset of actually scheduled UEs $\mathcal{U}_{C,\text{scheduled}} \subseteq \mathcal{U}_C$ and estimates the weighted sum rate R_C as:

$$R_C = \max_{\mathcal{U}_{C,\text{scheduled}} \subseteq \mathcal{U}_C} \sum_{u \in \mathcal{U}_{C,\text{scheduled}}} \alpha_u \cdot \text{ld}\,(1 + \gamma_u) \tag{2.4}$$

α_u is the scheduling weight and γ_u the Signal to Interference and Noise Ratio (SINR) of UE u.

To determine the optimal clustering, an integer optimization problem has to be solved for each scheduling interval.

$$\text{maximize} \quad \sum_{C \in \mathcal{C}} R_C \cdot x_C \tag{2.5a}$$

$$\text{subject to} \quad \sum_{C \in \mathcal{C}} a_{b,C} \cdot x_C \leq 1 \quad \forall b \in \mathcal{B} \tag{2.5b}$$

Equation (2.5a) constructs the clustering that maximizes the weighted system sum rate and Equation (2.5b) ensures that each TP is contained in at maximum one cluster. The binary variables x_C and $a_{b,C}$ are defined as:

$$x_C = \begin{cases} 1 & \text{Cluster } C \text{ selected} \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

$$a_{b,C} = \begin{cases} 1 & b \in C \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

[8] The authors use the term anchor TP to indicate the TP with the lowest channel attenuation.

Algorithm 2.1 Greedy cluster selection by Baracca et al.

```
1: $\mathcal{C}^{(A)} \leftarrow \mathcal{C}$
2: $x_C \leftarrow 0 \quad \forall C \in \mathcal{C}^{(A)}$
3: while $\mathcal{C}^{(A)} \neq \emptyset$ do
4:     $C^* \leftarrow \arg \max_{C \in \mathcal{C}^{(A)}} R_C / S_C$
5:     $x_{C^*} \leftarrow 1$
6:     $\mathcal{C}^{(A)} \leftarrow \mathcal{C}^{(A)} \setminus \left[C \in \mathcal{C}^{(A)} | C \cap C^* \neq \emptyset\right]$
7: end while
```

The authors argue that this problem is NP-complete and therefore it cannot be solved by exhaustive search in real systems. Instead, they propose a greedy cluster selection algorithm to solve the problem, shown in Algorithm 2.1.

The algorithm first adds all cluster candidates to the set of available candidates (Line 1) and marks all as not scheduled (Line 2). The following loop searches for the cluster C^* that leads to the highest weighted sum rate in relation to the cluster size S_C. This cluster is set as selected in Line 5. The last statement in the loop avoids overlapping clusters by removing all cluster candidates from $\mathcal{C}^{(A)}$ that have common TPs with the selected cluster.

The proposed dynamic clustering algorithm has several drawbacks. First, the tight coupling of clustering and scheduling might lead to changing clusterings every TTI. Depending on the underlying system architecture it might not be possible to reconfigure the clustering that often. A solution for this problem would be to select and configure a clustering that maximizes the weighted system sum rate and keep the clustering for a certain time. Then only the scheduling for the given clustering has to be performed every TTI.

Another severe limitation of this dynamic clustering algorithm is that due to the greedy cluster selection approach it might happen that not all TPs are used to serve UEs in a TTI. On the one hand this increases the weighted system sum rate, because the interference for the served UEs is reduced, but on the other hand this also reduces the total system capacity. In the case that the clustering can be changed every TTI this might be tolerable, but if the clustering is only reconfigured in intervals, this is not suitable anymore.

Algorithm II by Baracca et al.

The same group of authors that presented the dynamic clustering algorithm shown in Algorithm I by Baracca et al., introduced an improved variant two years later [BBB14]. The main improvements are the introduction of Multi-User Eigenmode Transmission (MET) as precoding scheme and the extension to support multiple receive antennas at the UEs. The actual cluster selection has been changed, such that the UEs not only propose a single cluster candidate that contains all TPs satisfying certain conditions, but the UEs propose multiple candidates. Therefore, a maximum cluster size must be configured in the first step. Then the UEs propose candidates of size one up to the maximum cluster size.

The drawback of Algorithm I by Baracca et al. that not all available TPs are used to transmit data is resolved with this variant, because for the cluster selection process cluster candidates with all possible sizes are available. As this includes also clusters of size one, it is always possible to find a solution that uses all TPs. Nevertheless, this solution is only used, if it increases the weighted system sum rate. The problem of tight coupling of cluster selection and scheduling remains, however.

Algorithm by Weber et al.

This algorithm is introduced by Weber et al. [Web+11] and refined in [MF11, chapter 7.2.2]. It follows the principle of cluster candidates proposed by the UEs. The UEs periodically report channel quality measurements to their serving TP, where this information is forwarded to a CU. The measurement reports include the quality in terms of the RSRP for a subset $C \subseteq \mathcal{B}$, where only the TPs stronger than a configurable threshold are included. In the CU this information is aggregated as a list of pairs (C, N_C), where N_C denotes the number of occurrences of cluster candidate C during the measurement period T_R.

At the CU the cluster candidates are filtered according to the allowed cluster size, such that the set of cluster candidates is $\mathcal{C}^{(c)} = [C|S_{C,\min} \leq S_C \leq S_{C,\max}]$, with the cluster size $S_C = |C|$. After the filtering, the total number of proposals is $\left|\mathcal{C}^{(c)}\right|$. Based on the proposals a linear program is formulated to generate the best clustering.

$$\begin{aligned} \text{minimize} \quad & \sum_{C \in \mathcal{C}^{(c)}} u_C \cdot x_C && \text{(2.8a)} \\ \text{subject to} \quad & \mathbf{Ax} \geq \mathbf{1} && \text{(2.8b)} \\ & \mathbf{x} \in \{0,1\}^{[|\mathcal{C}^{(c)}| \times 1]} && \text{(2.8c)} \end{aligned}$$

The binary matrix $\mathbf{A} \in \{0,1\}^{[N_{\mathrm{TP}} \times |\mathcal{C}^{(c)}|]}$ states which TP is included in which cluster candidate, the vector $\mathbf{x}$ describes if a cluster candidate is chosen and u_C represents a cost function associated with each candidate. The constraint in Equation (2.8b) assures that each TP is included in at least one cluster. The vector $\mathbf{1} \in \{1\}^{[N_{\mathrm{TP}} \times 1]}$ represents a column vector of ones. By enforcing the equality in this constraint, non-overlapping clusters are selected. The second constraint in Equation (2.8c) ensures that the variables $x_C \in \mathbf{x}$ are binary variables. By experiments in the used simulation framework the following cost function was determined:

$$u_C = \frac{S_C}{\sum\limits_{b \in C} \mathrm{RSRP}_b \cdot 10^{N_C}} \tag{2.9}$$

The authors argue that the cost function should represent the trade-off between complexity, i.e., the cluster size, and the performance gain the cluster can achieve. The performance gain is represented with the sum of the RSRP (in linear scale) of all TPs of the cluster candidate and the number of UEs that would profit by this cluster N_C. The channel quality per TP RSRP_b is defined as the average RSRP reported by all UEs.

The optimization problem is then solved by a sub-optimal algorithm that performs the following steps.

1. Generate the cluster candidates $\mathcal{C}^{(c)}$ exhaustively, based on the proposals (C, N_C).

2. Calculate for each cluster candidate C the cost u_C.
3. Create an initial solution by adding the cluster candidates in increasing cost order, until all TPs are included in the solution or there are no more possible cluster candidates available.[9]
4. Improve solution by step-wise replacement of two or more clusters with one cluster candidate not yet included, whose cost is lower than the sum of the cost of the replaced clusters.

Step 4 of the algorithm could become complex for larger scenarios, because it involves an exhaustive search over all selected clusters. Furthermore, this search has to be performed for each of the proposed cluster candidates.

2.4 Scheduling for CoMP

In this section, we first introduce how the resource allocation process has to be extended to support CoMP. Here we focus on JT-CoMP, as it is the most promising candidate and scheduling within the CS/CB framework is already well understood [RTL98; DVR03]. Then we show for the special cases of overlapping and non-overlapping clusters how scheduling is performed. As discussed, scheduling and clustering strongly depend on each other and should be performed in a common step to achieve good performance. However, in real systems the frequency of cluster reconfigurations might be limited. Therefore, we assume in the following the case that scheduling is performed every TTI and cluster reconfiguration is restricted to longer intervals or at minimum once per TTI.

After the clustering has been determined, the scheduler has to decide which UEs should be served. Thereby, the scheduler maximizes a common metric, e.g., the sum rate, or tries to ensure a certain fairness between the UEs, e.g., by applying the PF principle [Kel97]. In systems with JT-CoMP the scheduling has to be extended, because more possibilities to serve a UE exist. This is exemplary shown in Figure 2.15 with a network consisting of two TPs and two UEs. In this small example, three different options to serve the UEs exist[10], even if only one frequency resource and one spatial layer per UE is considered. In the first two options the UEs are served by a single TP each and in the third option the TPs serve the UEs cooperatively, such that no interference occurs. With increasing number of TPs and UEs the number of scheduling options increases rapidly. E.g., in a system with three TPs and three UEs 40 scheduling options exist. In a system with three TPs and 30 UEs the total number of scheduling options is 68 035. For details how these numbers are derived see Appendix A.2. The rapid growth is reasoned by the fact that with increasing number of TPs or UEs more possibilities are available how JT can be used. E.g., in a system with three TPs it is possible that two TPs cooperatively serve two UEs, while the last TP serves another UE.

[9] This potentially generates overlapping clusters. If this is not intended, after each step all cluster candidates with TPs already included in the solution have to be removed.

[10] The fourth option, where no UE is scheduled is omitted here, because this is no reasonable option, if data should be transmitted.

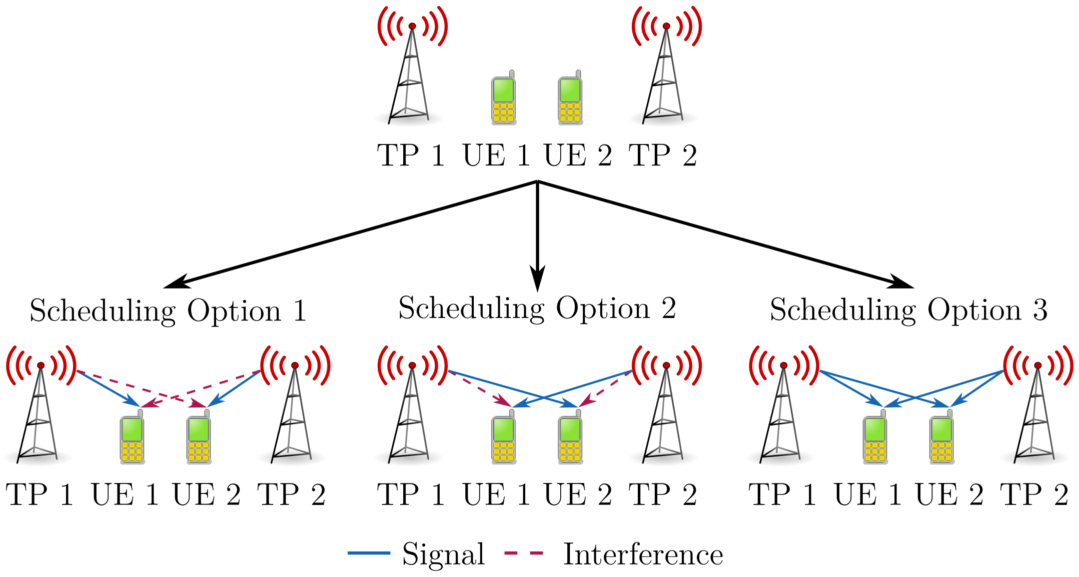


Figure 2.15: Scheduling options with CoMP

In order to select the scheduling option that maximizes the scheduling metric, all options have to be considered and their scheduling metrics have to be calculated. Therefore, the effort is high, even if only a few of the available scheduling options are reasonable. E.g., in the example in Figure 2.15 only scheduling options 1 and 3 are reasonable, because in option 2 the interference exceeds the signal power, which results in poor performance.

In order to overcome the high complexity of an exhaustive search, greedy scheduling algorithms have been proposed in the literature. Greedy algorithms select one UE after the other, where each newly selected UE leads to the highest improvement of the scheduling metric. This principle has been combined with clustered CoMP for overlapping clusters by Gong et al. [Gon+11] and non-overlapping clusters by Baracca, Boccardi, and Braun [BBB12]. However, greedy algorithms do not guarantee to find a globally optimal solution, which is the case if an exhaustive search over all scheduling options is performed. Instead, the solution might only be a local optimum with reduced performance. Thiele et al. [Thi+12] present another greedy scheduling algorithm. The idea of the algorithm is to use a static clustering, but introduce sub-clusters that are selected on a per user basis. So potentially overlapping sub-clusters within a larger cluster are configured. This reduces the complexity of precoding as the number of TPs jointly serving a UE is reduced.

Scheduling cannot be regarded as an isolated task. Instead, it is closely related with the used transmission technique, which influences the achievable performance and thereby the scheduling priority of the UE. While the scheduler requires information about the performance, the performance is on the other hand influenced by the scheduled users. Therefore, finding an optimal resource allocation is often not possible [Li+14].

2.4.1 Non-overlapping Clusters

For non-overlapping clusters it is straightforward to extend the scheduling process to apply JT. While in networks without CoMP one scheduler per TP exists, for CoMP one scheduler per cluster is necessary. So each scheduler works independently of the other clusters and schedulers. Allowing cooperation between the schedulers would circumvent the goal of clustering, as its goal is to restrict cooperation only within the cluster. The same should also hold for scheduling.

Similarly as in classical networks the scheduler decides which time-frequency resource is assigned to which UE. Additionally, in case of JT or MU-MIMO the scheduler can assign a single time-frequency resource to multiple UEs. Then it also has to decide which scheduling option has to be used for the time-frequency resource.

2.4.2 Overlapping Clusters

Because overlapping clusters share common resources like TPs and radio resources, it is not possible to apply one scheduler per cluster. Instead, the scheduler must have an overview of the whole system. Therefore, a single scheduler could be introduced in the system as suggested by Dai and Yu [DY14] and Kang and Kim [KK16]. The central scheduler has to ensure that the total number of scheduled MIMO streams is below or equal to $N_{\mathrm{TP}} \cdot N_{\mathrm{TX}}$ and also the number of scheduled streams per cluster of size S_C must be below or equal to $S_C \cdot N_{\mathrm{TX}}$. These limits are reasoned, because the number of transmitted MIMO streams in the whole system and per cluster is limited by the number of available transmit antennas. Additionally, the scheduler has to decide how these streams are assigned to the UEs.

As we can see, a central scheduler is required in the case of overlapping clusters, which contradicts the principle of clustering, because the goal of clustering is to reduce the complexity of CoMP by splitting the problem into smaller subproblems.

2.5 Multiple Input Multiple Output (MIMO) and Multi-User MIMO (MU-MIMO) Transmission

An essential building block for the introduction of CoMP and especially Joint Transmission (JT) in cellular networks is the concept of Multi-User MIMO (MU-MIMO). MU-MIMO is an extension of Single-User MIMO (SU-MIMO). For an in depth study on the potential and limitation of MIMO see e.g., [Gol+03]. Even if MIMO and MU-MIMO can be applied for uplink and downlink transmissions in cellular networks, we restrict the following discussion to the downlink case only. Therefore, the functionality of the transmitter is provided by the BS and the UEs are the receivers. SU-MIMO exploits multiple spatial layers to transmit multiple data streams between one transmitter and one receiver on the same time-frequency resource. MU-MIMO extends this principle to one transmitter and multiple receivers. The basic idea of MU-MIMO is to process the transmit signals before sending them such that the effects of the channel are compensated and no interference between the receivers is generated. This is achieved by applying one of the precoding techniques described in Section 2.5.1. One requirement for SU-MIMO is that transmitter and receiver are equipped with multiple transmit and receive antennas, respectively. If the number of transmit and receive antennas is not equal, the additional degrees of freedom can be used to improve the reliability of signal transmission or reception. In the case of

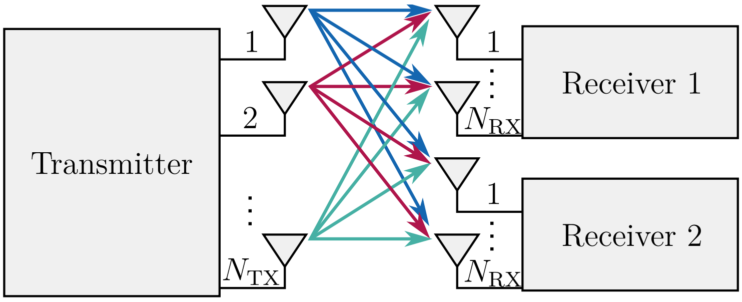


Figure 2.16: MU-MIMO with one transmitter and two receivers

MU-MIMO at least the transmitter has to be equipped with multiple transmit antennas. In both cases the total number of data streams is determined by the minimum of the available transmit and receive antennas. The maximum number of streams per receiver is limited by the available receive antennas. The basic MIMO architecture is depicted in Figure 2.16. Originally MU-MIMO has been introduced for single BSs with multiple transmit antennas [Spe+04; KS10]. JT extends this concept to multiple transmit antennas distributed to different BSs. This also reasons why the concept of JT is sometimes called network MIMO or distributed antenna system [Sti+10; Bas+17].

Power allocation is related to the problem of precoding, because the limited transmit power has to be distributed to the different streams. This problem is discussed in Section 2.5.2. Finally, we show in Section 2.5.3 how LTE networks support MIMO.

Before introducing precoding techniques in the next section, we briefly recap signal transmission over a wireless channel. We use the depicted example in Figure 2.17 with a non-cooperative wireless network with three transmitters ($N_{\mathrm{TP}} = 3$) and three receivers ($N_{\mathrm{UE}} = 3$), each with a single antenna. Therefore, each transmitter can only send a single data stream. In the example, receiver 1 is served by transmitter 1, receiver 2 by transmitter 2 and receiver 3 by transmitter 3. The complex-valued coefficient $h_{u,b}$ describes the channel between transmitter b and receiver u. All channel coefficients can be combined in the channel matrix $\mathbf{H} \in \mathbb{C}^{[N_{\mathrm{UE}} \times N_{\mathrm{TP}}]}$:

$$\mathbf{H} = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{pmatrix} \tag{2.10}$$

Using vector and matrix notation, the signal transmission in the system is described as:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \mathbf{Hx} + \mathbf{n} = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \tag{2.11}$$

The vector $\mathbf{y} \in \mathbb{C}^{[N_{\mathrm{UE}} \times 1]}$ describes the received signals by receivers one, two and three. The vector $\mathbf{x} \in \mathbb{C}^{[N_{\mathrm{UE}} \times 1]}$ is the data signal transmitted by transmitters one, two and three. In the following, we assume a normalized transmit signal, which means that the components of $\mathbf{x}$ have an absolute value of one ($|x_i| = 1$). The received noise is described by the vector $\mathbf{n} \in \mathbb{C}^{[N_{\mathrm{UE}} \times 1]}$. Therefore, the received signal for receiver 1 is:

$$y_1 = \underbrace{h_{1,1} \cdot x_1}_{\text{signal}} + \underbrace{h_{1,2} \cdot x_2 + h_{1,3} \cdot x_3}_{\text{interference}} + \underbrace{n_1}_{\text{noise}} \tag{2.12}$$

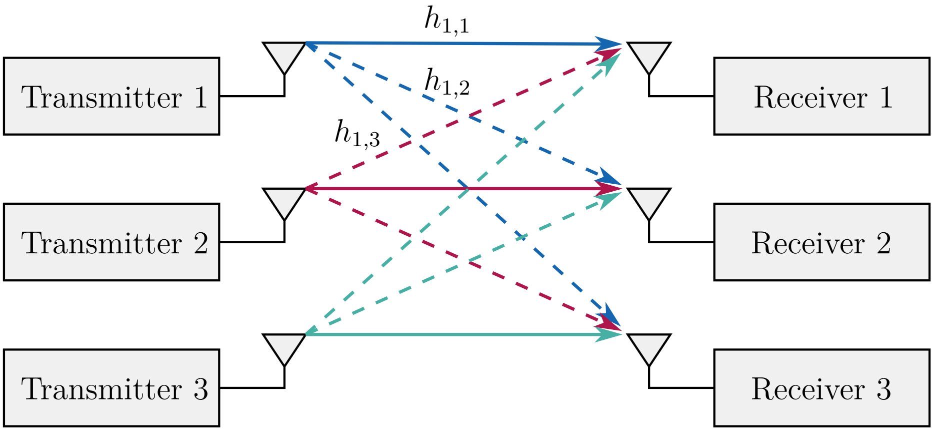


Figure 2.17: Wireless signal transmission

From Equation (2.12) it becomes evident that the received signal is composed of the useful data signal, the interference caused by the other transmitters and the noise. To describe the quality of the received signal the Signal to Interference and Noise Ratio (SINR) γ is a common metric. The SINR is defined as:

$$\gamma = \frac{P_{\text{signal}}}{P_{\text{interference}} + \sigma^2} \tag{2.13}$$

P_{signal} is the power of the received data signal, $P_{\text{interference}}$ is the interference power and σ^2 the noise power. Sometimes also the Signal to Noise Ratio (SNR) is used, which is the ratio between signal and noise power.

The same notation is also applicable for MIMO and MU-MIMO. In the following, we use it to explain the concepts of precoding and power allocation.

2.5.1 Precoding

In general, it can be distinguished between linear and nonlinear precoding techniques. While linear precoding is generally less complex to implement, nonlinear precoding achieves better performance. Nevertheless, for most precoding techniques good channel knowledge is required. For now, we assume perfect CSI knowledge at the transmitter. Later, we show in Section 2.6 how acquisition of the CSI is achieved. Furthermore, we restrict the discussion to single antenna receivers, i.e., $N_{\text{RX}} = 1$. Extensions to multi-antenna receivers are discussed by Viswanathan, Venkatesan, and Huang [VVH03] and Spencer et al. [Spe+04].

Equation (2.14) shows how the received signal vector $\mathbf{y} \in \mathbb{C}^{[N_{\text{UE}} \cdot N_{\text{RX}} \times 1]}$ is obtained from the signal $\mathbf{x} \in \mathbb{C}^{[N_{\text{UE}} \cdot N_{\text{RX}} \times 1]}$ if precoding is applied.

$$\mathbf{y} = \mathbf{H} D(\mathbf{x}) + \mathbf{n} \tag{2.14}$$

Here $\mathbf{H} \in \mathbb{C}^{[N_{\text{UE}} \cdot N_{\text{RX}} \times N_{\text{TP}} \cdot N_{\text{TX}}]}$ is the complex channel matrix and $\mathbf{n} \in \mathbb{C}^{[N_{\text{UE}} \cdot N_{\text{RX}} \times 1]}$ the noise at the receivers. The operation $D(\bullet)$ represents the precoding applied on the signal before transmission. We can further rewrite $D(\bullet)$ as $\mathbf{W} d(\bullet)$, where the function $d(\bullet)$ consists of all nonlinear operations and the multiplication with the precoding matrix

$\mathbf{W} \in \mathbb{C}^{[N_{\mathrm{TP}} \cdot N_{\mathrm{TX}} \times N_{\mathrm{UE}} \cdot N_{\mathrm{RX}}]}$ represents the linear operations. The components of the precoding matrix are $w_{u,r,b,t} = w_{u,b,t}$. The first two indices denote the receive antenna r of UE u and identify a distinct column in the precoding matrix. Because we only consider single antenna receivers, the index for the receive antenna can be omitted. The second last indices denote the transmit antenna t of TP b and identify a row in $\mathbf{W}$.

The goal of precoding is that the received signal becomes:

$$\mathbf{y} = \mathbf{x} + \mathbf{n} \tag{2.15}$$

This means that the channel impairments are canceled out and the received signal is only distorted by noise. For multi-antenna receivers and SU-MIMO, the goal in Equation (2.15) is not strictly necessary, because receiver-side processing can be applied to reconstruct the transmitted signals.

From the precoding point of view there is no difference between SU-MIMO and MU-MIMO, because precoding operates on the signal layer. On this layer it cannot be distinguished if one or several receivers are served. The same also holds for MU-MIMO and JT, which only differ in the locations of the transmit antennas. However, an important difference lies in the structure of the channel matrix. For SU-MIMO the channel coefficients show a relatively high correlation, because the paths between the transmit and receive antenna pairs are similar. With MU-MIMO the correlation between the receive antennas of the different receivers is smaller, because the paths are different. Therefore, the gains achieved with MU-MIMO can be expected to be higher than for SU-MIMO [Dup+11]. For JT the correlation is even smaller, as not only the receive antennas are located at multiple locations, but also the transmit antennas.

2.5.1.1 Linear Precoding

In this section, we focus on linear precoding. So the function $d(\bullet)$ is the identity projection $d(\mathbf{x}) = \mathbf{x}$. To achieve the precoding goal stated in Equation (2.15), we could set the precoding matrix $\mathbf{W}$ to the inverse of the channel matrix $\mathbf{H}^{-1}$. However, inversion of the channel matrix $\mathbf{H}$ is only possible if both dimensions have the same size. For the general case, the pseudoinverse (Moore-Penrose pseudoinverse) $\mathbf{H}^{+}$ has to be used. The product $\mathbf{HW}$ results in a diagonal matrix. This reasons also why this precoding technique is called channel inversion or Zero Forcing (ZF), as the interference is forced to zero. The coefficients of $\mathbf{W}$ directly represent the used transmit power. In the case of limited transmit power, either per BS or per transmit antenna, the coefficients of $\mathbf{W}$ have to be chosen such that the available power is not exceeded. We treat this aspect in Section 2.5.2. Spencer et al. [Spe+04] state that ZF is a good approach in the low noise or high transmit power regime. However, in the other cases, the performance of ZF is reduced because high transmit powers are required to revert the influence of an ill-balanced channel matrix and to suppress interference.

By allowing minimal interference at the receivers, the overall performance of linear precoding can be improved [Spe+04]. An example for this principle is the Minimum Mean Square Error (MMSE) precoder. This leads to the precoding matrix $\mathbf{W} = \mathbf{H}^{H} \left(\mathbf{H}\mathbf{H}^{H} + \alpha \mathbf{I}\right)^{-1}$.

With the loading factor α. The optimal SINR at the receivers is reached for a loading factor of $\alpha = \frac{N_{\mathrm{UE}}}{P}$ [PHS05], where P represents the total available transmit power of the transmitter.

The presented approaches of channel inversion work for arbitrary numbers of receive antennas at the receivers. However, in the case of multiple receive antennas, it is inefficient to enforce complete interference suppression between the receive antennas of a single receiver [She+06]. The reason is that the correlation of the channel between the closely spaced antennas of a single receiver is usually higher than the correlation between antennas from different receivers. Therefore, high transmit powers are required to suppress interference between antennas of a single receiver. If we take into account that each receiver can perform receiver-side signal processing to reduce the interference between its antennas, more transmit power can be used to transmit actually useful signals. Mathematically this means that instead of performing a complete channel inversion, it is sufficient that the matrix product **HW** results in a block diagonal matrix. Equation (2.16) below shows an example of the block diagonalization approach.

$$\mathbf{HW} = \begin{pmatrix} v_{1,1} & v_{1,2} & 0 & 0 & 0 & 0 \\ v_{2,1} & v_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & v_{3,3} & v_{3,4} & v_{3,5} & 0 \\ 0 & 0 & v_{4,3} & v_{4,4} & v_{4,5} & 0 \\ 0 & 0 & v_{5,3} & v_{5,4} & v_{5,5} & 0 \\ 0 & 0 & 0 & 0 & 0 & v_{6,6} \end{pmatrix} \tag{2.16}$$

In this example we can see the result of the block diagonalization, where $v_{i,j}$ represents non-zero values. The three served receivers have different numbers of receive antennas. E.g., the first receiver has two receive antennas, whereas the second and third receiver have three and one receive antennas, respectively. The receivers can then perform receive beamforming to reduce or completely suppress the interference between the different receive antennas.

2.5.1.2 Nonlinear Precoding

The concept of nonlinear precoding was independently introduced by Tomlinson [Tom71] and Harashima and Miyakawa [HM72]. It is therefore also called Tomlinson-Harashima precoding. Originally it has been developed for single dimension constellations, e.g., Amplitude Shift Keying (ASK) or Binary Phase Shift Keying (BPSK) modulation, whereas for efficient communication multi-dimensional constellations, e.g., QAM, are required [TV05]. The theoretical foundation for efficient nonlinear precoding was introduced by Costa [Cos83] as “writing on dirty paper” or also called Dirty Paper Coding (DPC). The remarkable characteristic of the concept is that if the caused interference is known at the transmitter, the same channel capacity can be reached as if there is no interference in the system at all. For the downlink the theoretical results of Costa can be achieved if the transmitter has full channel knowledge. Then the transmitter knows what interference is generated by the transmitted signal for the first receiver on the received signal of a second receiver. The signal for the second receiver can be designed such that the interference caused by the first receiver is avoided [Spe+04]. There are two commonly discussed variants to design the signals if the caused interference is known [KS10].

The first one is based on a QR-decomposition of the channel matrix $\mathbf{H} = \mathbf{LQ}$ [Spe+04], where $\mathbf{L}$ is a lower triangular matrix[11] and $\mathbf{Q}$ is a unitary matrix[12]. Then the precoding matrix is set to $\mathbf{W} = \mathbf{Q}^H$, such that the effective channel $\mathbf{H}_{\text{eff}}$ becomes:

$$\mathbf{H}_{\text{eff}} = \mathbf{HW} = \mathbf{LQQ}^H = \mathbf{L} \tag{2.17}$$

As the effective channel is now described by a lower triangular matrix, the first receiver observes no interference from the signals of the other receivers. Therefore, no additional signal processing is required. However, the second receiver receives interference from the signal for the first receiver. This can be handled, by subtracting the known interference before transmitting the signal. The same applies for all successive receivers that receive interference from all preceding signals.

The second variant, called vector precoding, designs the signals for all receivers in one step [PHS05; HPS05]. Vector precoding is a modified variant of channel inversion precoding, where the transmitted signal power is minimized. For vector precoding, the linear part of the precoding operation $\mathbf{W}d(\bullet)$ becomes $\mathbf{W} = \mathbf{H}^+$. The nonlinear operation is:

$$d(\mathbf{x}) = \mathbf{x} + \tau\mathbf{l} \tag{2.18}$$

The positive real-valued scalar τ is named modulo constant. The value of τ has to be chosen such that it is larger than all possible real or imaginary signal values. One possibility to reach the goal is:

$$\tau = 2|x_k|_{\max} + \Delta \tag{2.19}$$

$|x_k|_{\max}$ is the signal constellation symbol with the highest magnitude and Δ is the distance between neighboring constellation symbols. The complex-valued vector $\mathbf{l}$ of integer offsets is chosen as:

$$\mathbf{l} = \arg\min_{\mathbf{l}'} ||\mathbf{H}^+(\mathbf{x} + \tau\mathbf{l}')|| \tag{2.20}$$

By this the power of the transmitted signal after precoding is minimized. The received signal then becomes:

$$\mathbf{y} = \mathbf{x} + \tau\mathbf{l} + \mathbf{n} \tag{2.21}$$

The integer offset $\mathbf{l}$ is removed at the receivers by applying the modulo operation, so the received signal finally becomes:

$$\mathbf{y} \bmod \tau = (\mathbf{x} + \tau\mathbf{l} + \mathbf{n}) \bmod \tau = (\mathbf{x} + \mathbf{n}) \bmod \tau \tag{2.22}$$

By setting the modulo constant τ as defined in Equation (2.19), the modulo operation becomes transparent and the received signal is:

$$\mathbf{y} \bmod \tau = \mathbf{y} = \mathbf{x} + \mathbf{n} \tag{2.23}$$

2.5.2 Power Allocation

Power allocation deals with the task to distribute the available transmit power at the BSs to the served UEs. Thereby, the power has to be distributed to subcarriers as well

[11] All entries above the main diagonal are zero.

[12] $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$

as to UEs that are served on the same time-frequency resource, but on different spatial layers. In the following, we assume that the available power is distributed equally to the available subcarriers and we focus only on the allocation of power to multiple UEs served on the same time-frequency resource. Due to simplicity of mathematical formulation and later optimization often a sum power constraint per BS is assumed, which means that the available transmit power can be freely distributed to the antennas. However, in real systems, each transmit antenna has its own power amplifier, which also has to be treated in the task of power allocation [YL07].

The allocated power is directly related to the precoding matrix $\mathbf{W}$. Its components $w_{u,b,t}$ reflect the allocated power $p_{u,b,t}$ from antenna t of TP b to UE u. For a normalized transmit signal and linear precoding, the following relation holds:

$$p_{u,b,t} = |w_{u,b,t}|^2 \tag{2.24}$$

Here we discuss two power allocation variants, namely Water Filling (WF) and Equal Power Allocation (EPA). Even if EPA results in non-optimal solutions, it becomes nearly optimal in the high SNR regime [Bjö+13].

2.5.2.1 Water Filling

Water Filling (WF) power allocation is the solution to the following optimization problem [PF05]. We neglect interference caused by neighboring BSs and the achievable performance is only determined by the used transmit power. Also, we only consider a single time-frequency resource.

$$\begin{aligned}
\text{maximize} \quad & \sum_{u=1}^{N_{\text{UE}}} \sum_{b=1}^{N_{\text{TP}}} \sum_{t=1}^{N_{\text{TX}}} \text{ld}(1 + p_{u,b,t} \cdot \lambda_{u,b,t}) && \text{(2.25a)}\\
\text{subject to} \quad & \sum_{u=1}^{N_{\text{UE}}} \sum_{b=1}^{N_{\text{TP}}} \sum_{t=1}^{N_{\text{TX}}} p_{u,b,t} \leq P && \text{(2.25b)}\\
& p_{u,b,t} \geq 0 \qquad \forall u, b, t && \text{(2.25c)}
\end{aligned}$$

N_{UE} is the number of served UEs, $\lambda_{u,b,t} = \frac{h_{u,b,t}}{\sigma^2}$ is the channel gain between UE u and transmit antenna t of TP b, σ^2 is the noise power, $p_{u,b,t}$ is the power allocated to UE u from transmit antenna t of TP b and P is the sum power of all considered transmit antennas. Without loss of generality, we assume that all TPs have the same number of transmit antennas N_{TX}. However, the extension to different numbers of antennas per TP is trivial. The objective function in Equation (2.25a) states that the total data rate should be maximized. Because a UE can be served by multiple TPs, all transmit antennas have to be considered, which is achieved by the sums over N_{TP} and N_{TX} in the objective function. The constraints in Equations (2.25b) and (2.25c) restrict the total used transmit power to the available power P and ensure that no negative power is allocated to $p_{u,b,t}$. The total transmit power P is the sum of the available transmit powers per antenna P_{TX} $\left(P = \sum_{b=1}^{N_{\text{TP}}} \sum_{t=1}^{N_{\text{TX}}} P_{\text{TX}}\right)$. Again we assume without loss of generality the same transmit power P_{TX} for all antennas.

The Water Filling approach yields the following solution for the transmit powers of the individual streams:

$$p_{u,b,t} = \left(\mu - \lambda_{u,b,t}^{-1}\right)^{+} \tag{2.26}$$

The operation $(\bullet)^{+}$ is defined as $(\bullet)^{+} = \max(0, \bullet)$ and μ is the water level satisfying the sum power constraint such that $\sum_{u=1}^{N_{\text{UE}}} \sum_{b=1}^{N_{\text{TP}}} \sum_{t=1}^{N_{\text{TX}}} p_{u,b,t} = P$. The actual power allocation following the WF principle can be performed efficiently using an iterative approach [Yu+01]. For some cases also closed-form solutions exist [SSH04].

While a sum power constraint is questionable in the case of a single TP with multiple transmit antennas, it is not applicable in the case of distributed cooperating TPs as it is the case for JT-CoMP. For this reason extensions to the classical WF approach have been proposed [BH06]. In the optimization problem in Equation (2.25b) the sum power constraint has to be replaced by a per antenna power constraint:

$$\sum_{u=1}^{N_{\text{UE}}} p_{u,b,t} \leq P_{\text{TX}} \qquad \forall b, t \tag{2.27}$$

Scaling the transmit powers on a per antenna basis, as achieved by Equation (2.26), is not suitable as it would result in unbalanced transmit powers, which results in interference between the signals from different transmit antennas. Instead, the transmit powers of all transmit antennas have to be scaled jointly. E.g., Batista et al. [Bat+10] propose to scale the whole precoding matrix $\mathbf{W}$, such that the transmit power of the transmit antenna with largest precoding weights $w_{u,b,t}$ becomes equal to P_{TX}. However, this approach does not use the available transmit power from the other power amplifiers.

2.5.2.2 Equal Power Allocation

While solving the Water Filling problem might be computationally complex, especially for the case of per antenna power constraints, Equal Power Allocation (EPA) distributes the available transmit power equally to the served streams.

$$\sum_{b=1}^{N_{\text{TP}}} \sum_{t=1}^{N_{\text{TX}}} p_{u,b,t} = \frac{P}{N_{\text{UE}}} \qquad \forall u \tag{2.28}$$

In the case of per antenna power constraints, the same principles apply as for the Water Filling approach and the constraint from Equation (2.27) has to be fulfilled.

2.5.3 Standardization of MIMO and MU-MIMO in LTE

LTE supports SU-MIMO and MU-MIMO since release 8 by the standardized Transmission Modes (TMs) listed in Table 2.2. A TM describes how the modulated data symbols are mapped to the available antennas and how the channel is estimated and reported back to the BSs. Channel measurement and reporting is treated separately in Section 2.6.2. To allow vendors more flexibility, the LTE standards do not directly specify which antennas have to be used or how they are placed on the cell towers. Instead, the concept of antenna

TM	Description	maximum MIMO layers	LTE release
1	Single antenna transmission	1	8
2	Transmit diversity	1	8
3	Open-loop codebook-based precoding in the case of more than one layer, transmit diversity in the case of single layer transmission	4	8
4	Closed-loop codebook-based precoding	4	8
5	Closed-loop codebook-based precoding for MU-MIMO	1	8
6	Closed-loop codebook-based precoding for single layer transmission	1	8
7	Non-codebook-based precoding for single layer PDSCH transmission	1	8
8	Non-codebook-based precoding for up to two layers	2	9
9	Non-codebook-based precoding for up to eight layers	8	10
10	Non-codebook-based precoding for up to eight layers with support of CoMP	8	11

Table 2.2: Overview of standardized transmission modes in LTE, [DPS16, page 118]

ports is used [3GPP 36.211]. Antenna ports are defined such that two symbols transmitted via the same antenna port will perceive the same radio channel. E.g., a release 10 BS could use its eight individual transmit antennas to serve legacy release 8 UEs by combining two successive transmit antennas into four virtual antenna ports and serve legacy UEs with up to four spatial layers. This combination enables the BS to use all antennas as well as to increase diversity.

Furthermore, LTE also supports to use the additional degrees of freedom of multiple transmit antennas or antenna ports to increase the reliability of transmissions. This is achieved by transmitting the same information via multiple antennas. Because the channels between different transmit and receive antenna pairs are independent, the likelihood that the information can be decoded correctly by the receiver is increased. This principle is called transmit diversity and is achieved in LTE by Space-Frequency Block Coding (SFBC) for two antenna ports and Frequency-Switched Transmit Diversity (FSTD) for four or more antenna ports.

Multi-antenna transmission can be performed either by using a predefined codebook (codebook-based precoding) or without (non-codebook-based precoding). Furthermore, it can be distinguished between closed-loop precoding and open-loop precoding.

For codebook-based precoding, LTE specifies a set of precoding matrices (the codebook), from which the BS selects an appropriate precoder [3GPP 36.211]. The codebook contains precoding matrices for two antenna ports with one or two MIMO layers and for four antenna ports with one, two, three or four MIMO layers. If closed-loop precoding is used, the UE provides feedback which precoding matrix achieves the best performance. Because the codebook is predefined and known by UE and BS not the complete matrix has to be signaled, but it is sufficient to feed back the index of the matrix in the codebook.

This is called Precoding Matrix Indicator (PMI). As the BS does not have to follow the recommended precoding matrix, the actually used PMI has to be signaled to the UE together with the transmitted data. This is necessary because the UE has to know the effective channel, i.e., the channel the signals observe if precoding is applied, to perform receiver-side signal processing on the received signals. The first LTE release only provides the possibility to measure the channel without applied precoding (refer to Section 2.6), which reasons the necessity to signal the used PMI. Closed-loop precoding is only reasonable if the channel does not vary quickly, such that the reported PMI is still valid when it is used for downlink transmissions. This is not the case if UEs move with high speeds. In that case open-loop precoding is applied. For open-loop precoding the BS selects a sequence of precoding matrices that is also known at the UE. For each subcarrier a different precoding matrix is used. This leads to increased diversity, as the likelihood that one of the used precoders is suitable for the current channel realization is increased.

In contrast to the precoding schemes described before, release 9 introduced non-codebook-based precoding for multiple MIMO layers, which means the BS can select an arbitrary precoding matrix. Because this matrix is not signaled to the UE directly, special predefined reference signals, called Demodulation Reference Signals (DM-RS), are introduced in the transmission. For these signals the same precoding is applied as for the actual data transmission. So the UEs can reconstruct the precoding matrix by comparing the received DM-RS with the expected DM-RS. The BS defines the precoding matrix based on channel measurements provided by the UEs. In LTE not the actual channel coefficients are measured, but only the PMI of the predefined codebook and an overall channel quality are signaled from UE to BS. For more details refer to Section 2.6.2. Therefore, the selection of the precoding matrix could still rely on the predefined codebook [DPS16].

MU-MIMO is available since release 8, however the first LTE release only supports to serve two UEs simultaneously on the same time-frequency resources, while each of them is served with one MIMO layer. The introduction of Transmission Modes 8, 9 and 10 resolved this issue. There, the available MIMO layers can be distributed freely to the served UEs and non-codebook-based precoding is applied.

2.6 Channel Knowledge

This section first deals with the challenges of gaining accurate channel knowledge in Section 2.6.1. Then an introduction of channel measurement in LTE systems is provided in Section 2.6.2. Here we also discuss the overhead caused by channel measurements. Finally, Section 2.6.3 elaborates on the challenges of the channel measurements regarding the requirements of CoMP.

2.6.1 General Channel Measurement Framework

While it is known that having complete channel knowledge at transmitter and receiver yields large performance improvements, this condition is hard to fulfill in real-world systems [Lov+08]. In TDD systems, which use the same channel for forward and backward communication, the channel reciprocity can be exploited to gain channel knowledge by introducing well-defined reference or training signals besides the actual payload data. Reference Signals (RS) are predefined sequences known by both the transmitter and receiver. By comparing the received RS with the expectation, the receiver is able to estimate the channel. In FDD systems this is not possible, because different channels are used for forward and backward communication. Thus, the transmitter cannot directly obtain the channel information, but depends on feedback of the receiver. The usable capacity of the channel is reduced by the additional RS in the forward direction for FDD and TDD systems. In FDD systems the capacity in the backward direction is reduced due to the feedback of channel information from receiver to transmitter. Additional problems rise from the fact that the channel varies over time, such that the measured channel information is only valid for the duration of the coherence time of the channel. Because measurement and feedback of the channel state requires some time, the information at the transmitter is always different from the actual channel, which leads to performance impairments [Man+15].

In case of multiple receive and transmit antennas the channel information for each antenna pair has to be measured. For each transmit antenna orthogonal reference symbols are needed. Orthogonality can be achieved by time-division, frequency-division or code-division multiplexing. The overhead caused by RS generally scales linearly with the number of transmit antennas, which significantly increases the overhead [CSL16].

2.6.2 Channel Measurement in LTE

The general framework for channel measurements was introduced in LTE release 8 and was improved in the following releases. Because the resource allocation in LTE is fully controlled by the eNodeB, uplink and downlink channel information is required in the eNodeB. In the following, we describe the mechanisms used in FDD systems, as they are more widely deployed in Europe and are more challenging in terms of obtaining the channel state.

Besides the general channel measurement framework used for scheduling or handover decisions, LTE also introduces Demodulation Reference Signals (DM-RS). These special RS are transmitted in the uplink and downlink and are precoded with the same precoder as the normal data transmissions. Thereby, the receiver can estimate the effective channel and apply receiver-side signal processing to improve the signal reception. In the following, we only treat the general channel measurement framework.

2.6.2.1 Uplink Measurement

For uplink channel measurements LTE provides Sounding Reference Signals (SRS). SRS can be transmitted periodically or aperiodically. Periodic SRS are transmitted once every 2 ms to 160 ms. Transmission of aperiodic SRS is triggered by the eNodeB by explicit signaling on the PDCCH. The SRS can be configured as wideband SRS transmission that covers a large part of the whole channel bandwidth in a single measurement. Alternatively, the SRS can be configured as narrowband SRS and cover only a fraction of the available bandwidth in a single transmission. In LTE the minimal bandwidth of narrowband SRS equals four RBs. The eNodeB decides which SRS need to be transmitted such that the frequency range of interest is covered. By applying frequency domain multiplexing and orthogonal reference sequences SRS transmissions of multiple UEs on the same subcarriers can be distinguished.

2.6.2.2 Downlink Measurement

Downlink channel information is provided by the UEs to the eNodeB in terms of Channel State Information (CSI) reports [3GPP 36.213]. The complete CSI consists of:

- The Channel Quality Indicator (CQI) represents the index in a table of 16 predefined MCS. If a transmission is carried out using the recommended MCS, Rank Indicator (RI) and PMI the block error probability is at most 10 %.
- The Rank Indicator (RI) recommends how many MIMO layers should be used for a transmission.
- The Precoding Matrix Indicator (PMI) recommends which precoding matrix from a predefined codebook should be used for a transmission, if the number of MIMO layers suggested in the RI is used.
- Introduced with release 13, the CSI-RS Resource Indicator (CRI) recommends which beam should be used to serve the UE if full-dimension MIMO is used.

Which elements are included in the CSI and how detailed they are, depends on the configured reporting mode of the UE. Similar as for the uplink, the downlink CSI can be transmitted periodically or aperiodically. Aperiodic CSI is transmitted on the Physical Uplink Shared Channel (PUSCH), while periodic CSI can be transmitted on the Physical Uplink Control Channel (PUCCH) or PUSCH. Periodic CSI can be transmitted up to every 2 ms. Because the capacity of the PUCCH is scarce, the channel information provided by periodic CSI is only coarse. For more detailed channel information the eNodeB requests aperiodic reports. These always contain a wideband CQI. Additional frequency selective reporting is possible, by dividing the total bandwidth into smaller subbands. Then individual CQI and PMI are reported for each of the subbands. It is possible that the UE or the eNodeB decides which subbands are covered by the CSI.

The CSI is gained by means of RS, inserted by the eNodeB at defined time-frequency resources. Cell-Specific Reference Signals (CRS) are the basic downlink reference signals, already introduced in release 8. CRS are transmitted in every RB. Because up to four different CRS can be transmitted per BS, it is also possible to gain channel information for multiple transmit antenna ports to support MIMO operation. Further, CRS are used for cell-selection and handover decisions. Channel State Information Reference Signals (CSI-RS) have been introduced in release 10, with the goal of more flexibility in CSI acquisition. They are especially important if more than four MIMO layers or coordinated transmissions between multiple cells are intended. The periodicity of CSI-RS is configurable from 5 ms up to 80 ms [3GPP 36.211].

The number of necessary CRS is directly related to the number of individual transmit antenna ports at the BS, e.g., for two transmit antenna ports two CRS are needed. In subframes used for CSI-RS transmissions, 40 REs are reserved in two consecutive RBs to transmit the CSI-RS since LTE release 13. In these reserved REs up to 16 CSI-RS can be transmitted. Each CSI-RS requires one RE. Only in case of a single used CSI-RS, this CSI-RS requires two REs. Which of the reserved REs are used to transmit the CSI-RS is indicated by the so called CSI-RS configuration. Each UE may use a different CSI-RS configuration. For this purpose the UEs have up to four CSI processes each corresponding to an individual CSI-RS configuration. Nevertheless, a single CSI-RS configuration is used in practice by multiple UEs to obtain the CSI [DPS16].

2.6.2.3 Explicit Interference Measurement

By combining multiple CSI-RS configurations and CSI processes, the channel quality for multiple transmit hypotheses can be measured. This concept can be used to introduce CS/CB-CoMP. Measuring transmit hypotheses is possible, because a TP not necessarily has to send a CSI-RS on a RE configured for CSI measurements. Instead, it can also transmit Zero Power Channel State Information Reference Signals (ZP CSI-RS), i.e., not transmitting at all. These REs are then called Channel State Information Interference Measurement (CSI-IM). For this reason CSI processes are defined by one CSI-RS configuration and one CSI-IM configuration. We show the principle using the network of three TPs depicted in Figure 2.18, in which UE 1 is served by TP 1. This example is taken and adapted from [DPS16, chapter 13.2.1].

To measure all transmit possibilities for all three TPs, seven orthogonal CSI-RS configurations are required. The orthogonality is achieved by using the same REs in all three TPs, but using them either as CSI-RS or ZP CSI-RS, as shown in Table 2.3. Then UE 1 uses the following four CSI processes to measure the CSI for all transmit hypotheses.

CSI process 1	CSI-RS: E	CSI-IM: A
CSI process 2	CSI-RS: E	CSI-IM: B
CSI process 3	CSI-RS: E	CSI-IM: C
CSI process 4	CSI-RS: E	CSI-IM: D

In all CSI processes UE 1 uses CSI-RS configuration E, which reflects that only TP 1 transmits. For interference measurements different CSI-IM configurations are used. With CSI process 1 the channel information is obtained under the hypothesis that neither TP 2 nor TP 3 cause interference. Processes 2 and 3 reflect the channel if TP 3 and TP 2 transmit, respectively. Finally, process 4 measures the channel, if both other TPs cause interference. UEs served by TP 2 would use CSI-RS configuration C in all CSI processes and the CSI-IM configurations A, B, E and F for interference measurements. Similarly, UEs served by TP 3 would use CSI-RS configuration B for CSI measurements and the CSI-IM configurations A, C, E and G.

In general, the number of needed CSI-RS configurations is $2^{N_{\mathrm{TP}}} - 1$. With N_{TP} being the number of TPs that are included for interference measurement. Because of the exponential growth of the number of necessary CSI-RS configurations and the limitation to four CSI processes per UE, this approach does not scale for large-scale interference measurements.

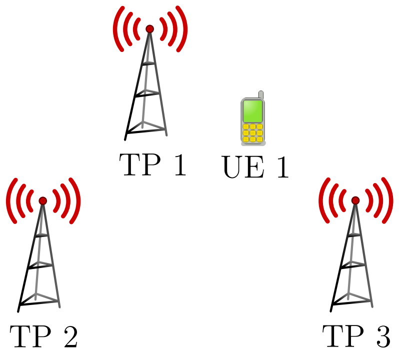


Figure 2.18: Example network to illustrate transmit hypotheses

CSI-RS Config.	TP 1	TP 2	TP 3
A	ZP CSI-RS	ZP CSI-RS	ZP CSI-RS
B	ZP CSI-RS	ZP CSI-RS	CSI-RS
C	ZP CSI-RS	CSI-RS	ZP CSI-RS
D	ZP CSI-RS	CSI-RS	CSI-RS
E	CSI-RS	ZP CSI-RS	ZP CSI-RS
F	CSI-RS	ZP CSI-RS	CSI-RS
G	CSI-RS	CSI-RS	ZP CSI-RS

Table 2.3: CSI-RS configurations for three TPs

Additionally, only 40 RE in two consecutive RBs are usable for CSI-RS transmission, such that the number of orthogonal CSI-RS configurations is limited.

2.6.2.4 Overhead caused by Channel Measurements

The transmission of RS causes significant overhead. Table 2.4 shows the number of used REs and the resulting overhead, if one, two or four CRS are transmitted. The overhead is defined as the fraction of the required number of REs for CRS divided by the total number of REs per RB. In case of a normal cyclic prefix, the total number of REs per RB is 84.

If more than four independent transmit antennas are used, CSI-RS have to be transmitted besides the CRS. Table 2.5 gives an overview of the caused overhead depending on the number of CSI-RS and their periodicity. Here the overhead is defined as the overhead per CSI-RS period. For each transmit antenna at least one CSI-RS is needed. The overhead for one or two CSI-RS is identical, because in these cases the same number of RS is transmitted. E.g., in case of a periodicity of $t_p = 10\,\mathrm{ms}$ and 16 CSI-RS, in total $20 \cdot 84$ REs (20 RBs with 84 REs per RB) are transmitted of which 16 REs are used for CSI-RS transmission. This leads to an overhead of $\frac{16\text{ REs}}{1680\text{ REs}} \approx 0.95\,\%$.

Comparing the overhead for CRS and CSI-RS, it is visible that the transmission of CRS generates a more than seven times larger overhead compared to the worst-case configuration for CSI-RS. For other CSI-RS configurations, the generated overhead is even smaller. Although the capabilities of CRS are limited in comparison to CSI-RS, they have to be transmitted in every RB due to backwards compatibility for legacy release 8 UEs.

2.6.3 Challenges Related to CoMP

Although the challenges of channel measurements required in systems with CoMP are similar to those in traditional networks, we want to highlight some differences. Again we focus only on the downlink direction. A detailed discussion for uplink channel measurement in the context of CoMP is provided by Wild [Wil15].

Number of CRS	Number of REs per RB	Overhead per RB
1	4	4.76 %
2	8	9.52 %
4	12	14.29 %

Table 2.4: Overhead of CRS transmission

Overhead	CSI-RS periodicity t_p				
Number of CSI-RS	5 ms	10 ms	20 ms	40 ms	80 ms
1/2	0.24 %	0.12 %	0.06 %	0.03 %	0.01 %
4	0.48 %	0.24 %	0.12 %	0.06 %	0.03 %
8	0.95 %	0.48 %	0.24 %	0.12 %	0.06 %
12	1.43 %	0.71 %	0.36 %	0.18 %	0.09 %
16	1.90 %	0.95 %	0.48 %	0.24 %	0.12 %

Table 2.5: Overhead of CSI-RS transmission

In networks with CoMP clusters, channel measurements are required on two different timescales. For the definition of clusters long-term average channel information is sufficient, if the clustering is not adapted in timescales close to the channel coherence time. On the other hand, detailed channel knowledge is required for precoding and scheduling. Especially precoding is sensitive to inaccurate or outdated channel information [Man+15]. Also in case of overlapping clusters special care should be taken in the selection of RS, because these have to be orthogonal within a cluster [CHY16]. Selection and assigning RS to clusters further increases the complexity of CoMP.

Often the overhead caused by channel measurement is not considered in the evaluation of CoMP, leading to unrealistic high performance gains in comparison to systems without coordination [RC09]. As shown in the previous section, the number of necessary orthogonal RS to cover all transmit hypotheses grows exponentially with the number of TPs taken into account. Especially large clusters in combination with multiple antennas per TP introduce high overhead, because of the larger number of required RS.

3 Dynamics in Cellular Networks

This chapter discusses dynamic effects in cellular mobile networks and mechanisms to cope with these effects. In Section 3.1, we introduce the effects, which influence the network on timescales ranging from milliseconds to minutes and hours. We focus on effects occurring in the envisioned scenario, i.e., an urban environment with vehicular traffic. Section 3.2 shows how traditional cellular networks adapt to the dynamics. In Section 3.3 we discuss the necessity to respect the network dynamics during the introduction of CoMP with dynamic clustering in cellular networks.

3.1 Sources of Dynamics

We start the overview of sources of dynamics in Section 3.1.1 with effects caused by the wireless transmission channel, which influence the network on relatively short timescales. Effects on longer timescales are caused by the traffic demands generated by the users in the network. These are treated in Section 3.1.2. In Section 3.1.3, we discuss the influence of the user mobility that also plays a role on longer timescales. To some degree these three causes of dynamics are related with each other. E.g., the channel is determined by the position of the user, which in turn is determined by its mobility.

3.1.1 Wireless Radio Channel

The wireless transmission channel between BS and UE is degraded by various influencing effects. These influences are variable in time as well as in frequency and are determined by the path and by the path change between transmitter and receiver, i.e., they are determined by the positions and relative speeds of transmitter and receiver. The stability of the channel is described by the coherence time and coherence bandwidth. The variability over the frequency band is handled in LTE by using OFDM, which divides the overall bandwidth into RBs with smaller bandwidth. Thereby, the channel can be considered as coherent over the bandwidth of a RB. Because the RBs only consist of a few OFDM symbols, the channel can be assumed to be coherent over the duration of a RB, too.

Figure 3.1 illustrates the signal propagation and explains two effects of time variant channel fading. Besides the Line of Sight (LoS) component of the signal the receiver also receives reflections from objects like buildings, vehicles, trees or other obstacles, which form the Non-LoS (NLoS) components of the signal. The different signal components overlap at the receiver and depending on the different path lengths interfere constructively or destructively. As the wavelength in LTE networks is in the range of centimeters (see Equation (3.1), with the speed of light c and the carrier frequency f_c[1]), even small differences in the path

[1] LTE can use multiple frequency bands. E.g., in Germany commercial networks use frequencies ranging from 700 MHz to 3.5 GHz. Therefore, we use the carrier frequency of $f_c = 2\,\text{GHz}$ as a rough average.

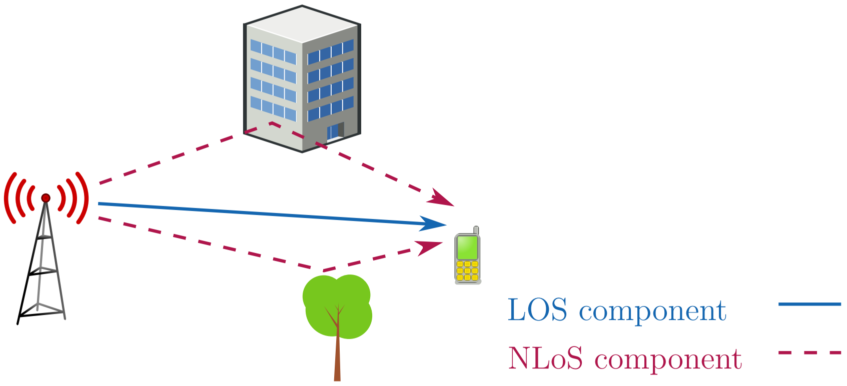


Figure 3.1: Illustration of wireless signal propagation

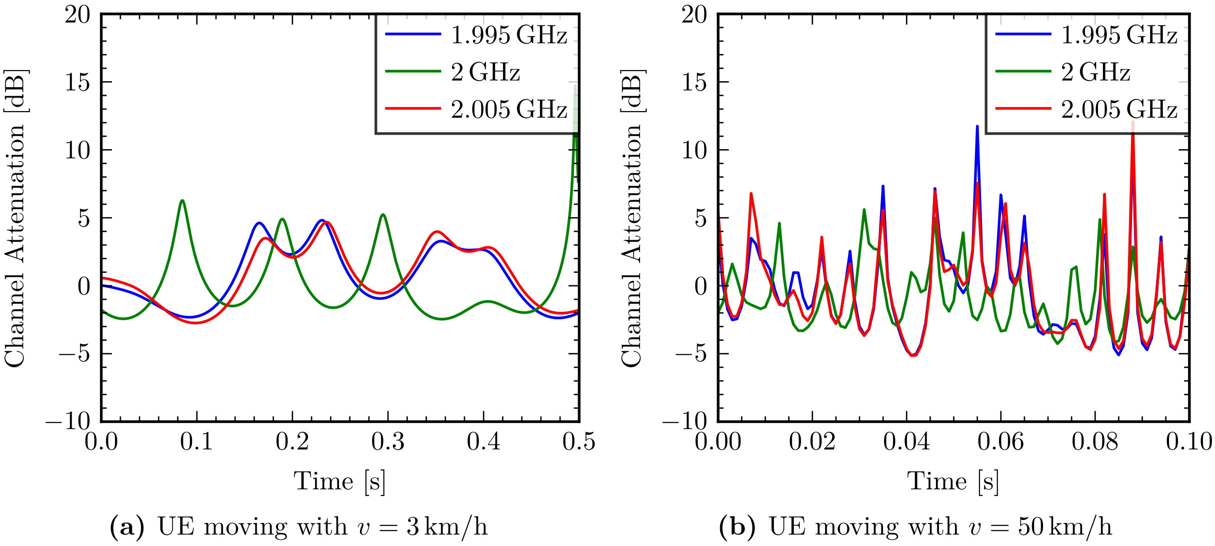
length can change the interference conditions. Because this kind of channel variation is caused by multiple transmission paths, this is also called multipath or fast fading.

$$\lambda = \frac{c}{f_c} = \frac{2.9979 \times 10^8\,\mathrm{m/s}}{2\,\mathrm{GHz}} \approx 15\,\mathrm{cm} \tag{3.1}$$

In urban scenarios it is also common that no direct LoS between transmitter and receiver exists and only reflected signals are received. This is for example the case if a building is between transmitter and receiver. Analogous to shadowing of light, this phenomenon is called shadowing of the wireless signal. Even if a LoS component exists, the transmitted signal is attenuated due to the path loss between transmitter and receiver. Shadowing and path loss are together called slow fading, because both mainly depend on the position of the UE, which only changes comparatively slow.

3.1.1.1 Fast Fading

The effect of fast fading is illustrated in Figure 3.2. The figure shows the channel attenuation for different frequencies and user movement speeds obtained from a system-level simulation. The simulation model is configured as described in Section 5.1.4. In both plots the distance between transmitter and receiver is kept constant such that the effect of fast fading is isolated and slow fading can be ignored. The plot on the left shows the fast fading for a receiver moving with $v = 3\,\mathrm{km/h}$ and the plot on the right for $v = 50\,\mathrm{km/h}$. In both cases the transmitter is stationary. We approximate the speed of pedestrians with $v = 3\,\mathrm{km/h}$ and of vehicles in an urban environment with $v = 50\,\mathrm{km/h}$. For a detailed discussion of vehicular speeds see Section 3.1.3.1. The plots also show the frequency selective fading of the channels for three different frequencies with a spacing of 5 MHz. These frequencies can be interpreted as the RBs with indices 0, 24 and 49 of an LTE network operating at a carrier frequency of $f_c = 2\,\mathrm{GHz}$. It is visible that the effect of fast fading is higher, if the receiver moves with higher speed. It can also be observed that the fading processes are independent for the different frequency indices. The figures also show that the channel attenuation could become negative, which means that the received signal power would be higher than the transmitted signal power. This is however not the case, because fast fading cannot be regarded isolated in real systems as there is always an additional channel attenuation caused by slow fading, such that the overall attenuation is positive.



(a) UE moving with $v = 3\,\mathrm{km/h}$

(b) UE moving with $v = 50\,\mathrm{km/h}$

Figure 3.2: Influence of fast fading for different frequency indices

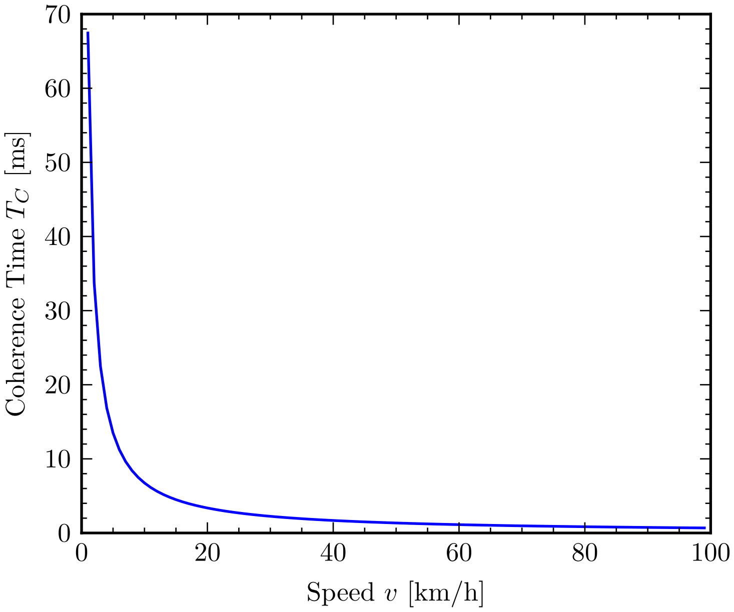


Figure 3.3: Channel coherence time in relation to the speed of the UE, $f_c = 2\,\mathrm{GHz}$

More formally, we can describe fast fading with the coherence time, i.e., the time over which the channel can be considered as not changing. According to Tse and Viswanath [TV05] it can be estimated by the Doppler spread D_s:

$$T_C = \frac{1}{4D_s} = \frac{1}{4 \cdot 2 f_c \frac{v}{c}} \tag{3.2}$$

Note that this is only one possibility to estimate the coherence time besides others. Especially the factor in the denominator varies from author to author, such that the stated coherence times should not be seen as absolute values, but rather indicate the order of magnitude only. Figure 3.3 shows the coherence time of a wireless channel with carrier frequency $f_c = 2\,\mathrm{GHz}$ in relation to the speed of a moving UE. As visible, the coherence

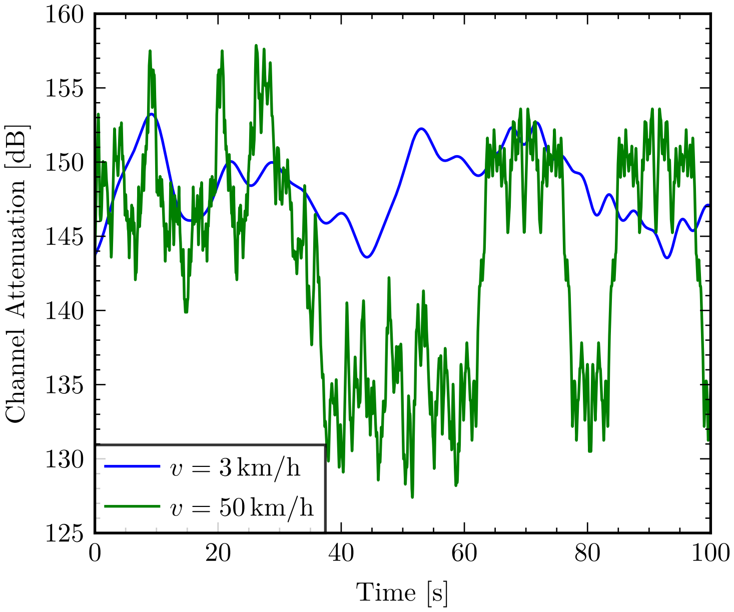


Figure 3.4: Influence of slow fading for different speeds

time for pedestrians is in the range of almost 50 ms, while for vehicles it is only in the range of milliseconds.

We conclude that fast fading can change the channel quality in the range of several dB from one millisecond to the next. Additionally, the fast fading is independent for different frequencies and independent of the position of the UE. However, the fading becomes faster, i.e., the coherence time becomes shorter, if the UE moves faster.

3.1.1.2 Slow Fading

Figure 3.4 illustrates the channel variation caused by slow fading. The plot shows slow fading between a stationary transmitter and two mobile receivers. One receiver is moving with 3 km/h and the other with 50 km/h. Both receivers move on the same trajectory. The channel for the receiver moving with higher speed is distinctly varying faster. Both causes of slow fading, i.e., path loss and shadowing, are shown in the figure. In the following, we discuss them individually.

Path Loss

Path loss describes the signal attenuation between transmitter and receiver due to the distance between them. According to Stüber [Stü01] the channel attenuation in a free space environment is:

$$h_{u,b} = -10 \log \left(\left(\frac{c}{4\pi \cdot d_{u,b} \cdot f_c} \right)^2 \right) \mathrm{dB} \tag{3.3}$$

$d_{u,b}$ denotes the distance between transmitter and receiver. However, cellular systems do not operate in free space, but the signal is at least reflected once from the ground [Stü01]. Thus, a more appropriate model for such a flat surface scenario is:

$$\begin{aligned} h_{u,b} &= -10\log\left(4\left(\frac{c}{4\pi\cdot d_{u,b}\cdot f_c}\right)^2 \sin^2\left(\frac{2\pi\cdot b_h\cdot u_h\cdot f_c}{c\cdot d_{u,b}}\right)\right)\text{dB} \\ &\underset{d_{u,b}\gg b_h\cdot u_h}{\approx} -10\log\left(\left(\frac{b_h\cdot u_h}{d_{u,b}^2}\right)^2\right)\text{dB} \end{aligned} \tag{3.4}$$

With b_h and u_h being the height of BS and UE, respectively. With the approximation shown in Equation (3.4), the path loss becomes independent of the carrier frequency and more notably the attenuation grows with the fourth power of the distance between transmitter and receiver. Besides these theoretical approaches also empirical models for the path loss have been obtained, which are based on measurements in different scenarios [Hat80; COST231]. We present the path loss model used in the evaluations in Section 5.1.4. This model is proposed by the 3GPP and commonly used for the evaluation of LTE networks [3GPP 36.814].

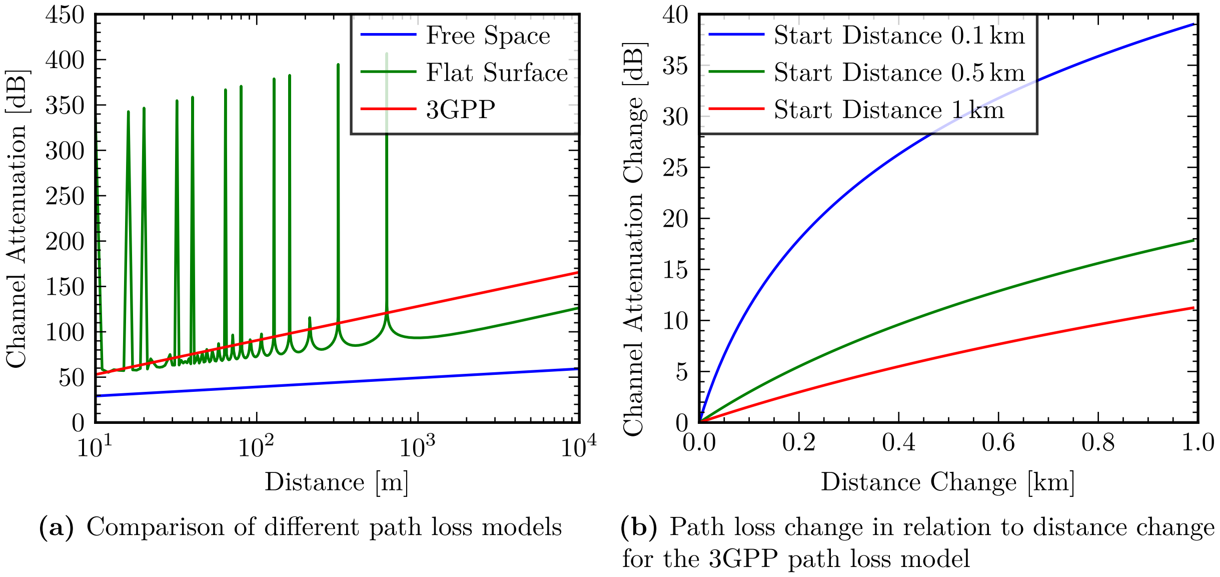
Figure 3.5a shows a comparison of all three models. In accordance to 3GPP recommendations a carrier frequency of $f_c = 2\,\text{GHz}$ and heights for transmitter and receiver of $b_h = 32\,\text{m}$ and $u_h = 1.5\,\text{m}$ are assumed, respectively. The spikes in the flat surface model are caused if the direct signal and the signal reflected from the ground interfere destructively with each other. It can be noted that the 3GPP model results in the highest overall path loss. The reason is that this model is especially tailored for urban environments where generally worse conditions can be expected.

Figure 3.5b shows how the path loss of the 3GPP model changes in relation to a distance change. Because this is influenced by the initial distance, three start positions are included. E.g., if the distance between transmitter and receiver changes from 100 m to 200 m, the path loss increases by 11.32 dB. From 500 m to 600 m it changes by 2.98 dB and from 1 km to 1.1 km only by 1.56 dB. A vehicle moving with 50 km/h can pass a distance of 100 m in approximately 7.2 s, such that the channel attenuation can change by several dB on a timescale of seconds. Nevertheless, the absolute path loss is higher for larger start distances.

Shadowing

Shadowing describes the signal attenuation caused by obstacles in the path between transmitter and receiver. Especially in urban scenarios where LoS conditions are rare, shadowing plays an important role.

Stüber [Stü01] states that the shadowing follows a log-normal distribution with zero mean. The standard deviation of the shadowing distribution is influenced by the environment. In general, the standard deviation is smaller in urban scenarios compared to suburban or rural scenarios. Several values for the standard deviation have been reported. For micro-cell environments, which are typically found in urban scenarios, these values range



(a) Comparison of different path loss models

(b) Path loss change in relation to distance change for the 3GPP path loss model

Figure 3.5: Path loss between UE and BS

from 4 dB to 13 dB [Stü01], while the 3GPP recommends values between 3 dB to 10 dB [3GPP 36.814]. The influence of the carrier frequency on the shadowing is relatively small. E.g., Mockford, Turkmani, and Parsons [MTP90] report that the standard deviation is 0.8 dB higher for a carrier frequency of 1800 MHz than for 900 MHz.

If the UE moves from one location to a nearby location, the shadowing does not change abruptly, but it is correlated between the two locations. A similar argumentation holds for the values of the shadowing components from two transmitters located at different sites to a single receiver. The shadowing of both transmitters is influenced by the same objects in the proximity of the receiver, such that the shadowing is also correlated. Gudmundson [Gud91] reports that the spatial correlation coefficient is 0.82 at a distance of 100 m in suburban environments and 0.3 at a distance of 10 m in micro-cellular, i.e., urban environments. Newer channel models used in the 3GPP standardization process assume a correlation coefficient of 0.5 at a distance of 50 m in urban scenarios [3GPP 36.814].

Figure 3.6 shows an example shadowing map. This example is based on the 3GPP shadowing model used for the evaluations in this thesis (see Section 5.1.4). A single transmitter is located in the center ($x = y = 0$ m) and the shadowing is evaluated in the shown area. In order to isolate the effect of shadowing, other effects like path loss or fast fading are neglected. This is also the reason why the shown attenuations may become negative, which would be prevented by the additional path loss in real systems. It is visible that the shadow fading does not change abruptly, but shadowing of nearby locations is correlated.

From the characteristics of shadow fading we can conclude that it influences the overall channel quality to a similar degree as the path loss. A UE has to move several meters to observe a significant change of the channel attenuation, such that shadowing influences the channel on a timescale of seconds.

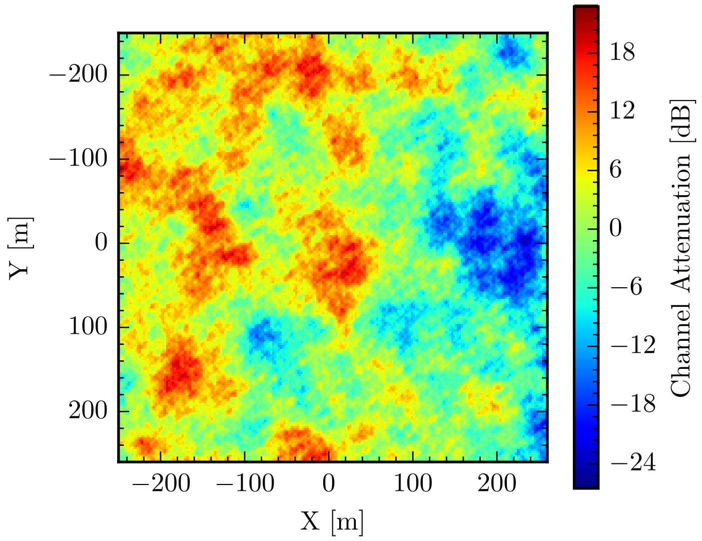


Figure 3.6: Example shadowing map

3.1.2 Traffic Demands

The traffic generated by the users imposes dynamics on multiple timescales. We discuss them in the following, starting with the shortest timescale caused by transport protocols (Section 3.1.2.1), continuing with dynamics caused by the applications (Section 3.1.2.2) and finishing with effects on longer scales caused by different user behavior throughout the day (Section 3.1.2.3).

Figure 3.7 briefly illustrates the relation between user, application and transport protocol. The sending of the individual packets is managed by the transport protocol. The application determines which data has to be transmitted, how many transport layer connections are used and when they are opened or closed. The application in turn is triggered by the user to become active. The example shows a simplified web browsing session, where the user accesses websites. Thus, the application objects represent the website code, images or embedded scripts. The objects are transmitted between server and web browser as individual IP packets, while for the transfer multiple Transmission Control Protocol (TCP) connections are used. The browser decides how many connections are created and which objects are transmitted in which connection. After all data is transmitted, the result is presented to the user.

Lower layers than the transport layer (OSI layer 4) are either transparent, i.e., not influencing the traffic characteristics, or actively managed by the cellular network, so they are not under control by the end user. Therefore, we discuss only layer 4 and above.

3.1.2.1 Protocol Behavior

Today TCP and User Datagram Protocol (UDP) are the most commonly used transport protocols. UDP was first specified in [RFC 768] and provides a connectionless transmission of datagrams between end hosts. Thus, it does not influence the sending behavior. In

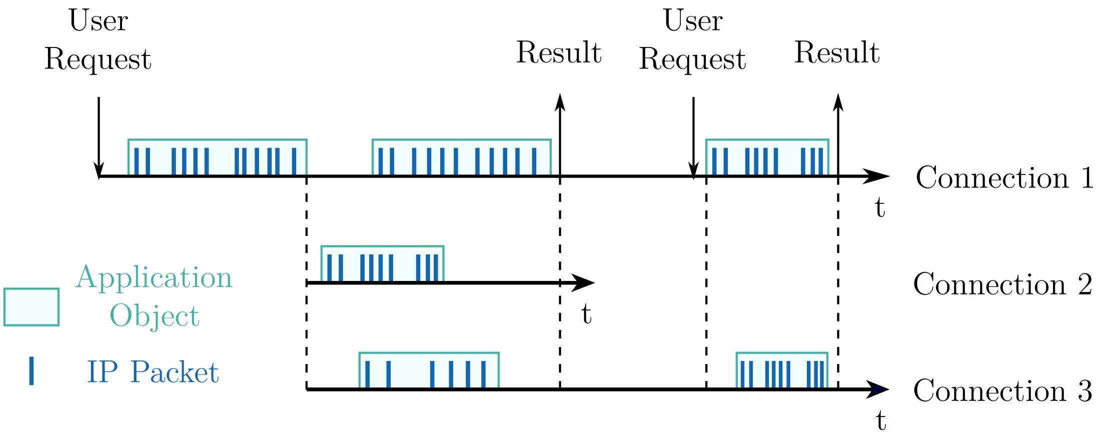


Figure 3.7: Illustration of different influences on the traffic characteristics

contrast, TCP provides a reliable, in-order and error-checked connection-oriented data transmission between two end hosts. Due to its advanced features, it actively controls the sending behavior and therefore has an impact on the traffic characteristics. The basic functionality is specified in [RFC 793]. The TCP congestion control, i.e., the adaption of the sending rate to the available link capacity, is described by [RFC 2581]. For the actual congestion control, several algorithms have been proposed. For an exhaustive discussion refer to [Afa+10]. The algorithms can be classified into two main categories by how they detect network congestion. Either by measuring packet delays or by detecting packet drops. TCP controls the sending rate by adapting the congestion window, i.e., the maximum amount of data the transmitter may send, before it has to wait for an Acknowledgement (ACK) from the receiver. Figure 3.8 shows a typical progression of the congestion window over time. After connection setup TCP starts in the slow-start phase where the congestion window is increased exponentially until congestion is detected. If congestion is detected, the congestion window is reduced and the congestion avoidance phase follows. In the congestion avoidance phase the congestion window is increased linearly until congestion is detected. In both phases, the transmitter increases the congestion window only if it receives an ACK. Therefore, the Round Trip Time (RTT), i.e., the time of a packet from transmitter to receiver plus the time of an ACK from receiver to transmitter, determines how quickly the transmitter adapts the congestion window. Additionally, the receiver controls the sending behavior, because the transmitter may only send as much data as the receiver can handle. Therefore, the receiver reports the receive window, i.e., the amount of data it can handle in its receive buffer, to the transmitter. The transmitter is only allowed to send as much data as the minimum of congestion window and receive window allows.

Many applications, like web browsing including cloud applications, emailing, video streaming or file sharing, use TCP as transport protocol. UDP is mainly used for real-time audio or video transmissions, which includes Voice over LTE (VoLTE), the voice service specified for LTE. Because only TCP controls the sending behavior, we focus the following discussion on the effects TCP introduces to data transmissions in cellular networks.

TCP was initially developed for wired networks, where the link capacity is constant and packet losses are mainly caused by buffer overflows. However, this is not the case in cellular networks, because the available link capacity of a UE is determined by channel attenuation and scheduling decision. As discussed in Section 3.1.1, the attenuation is highly dynamic

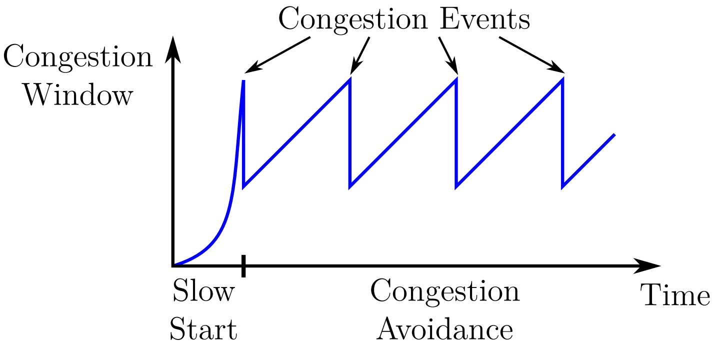


Figure 3.8: TCP congestion window over time

and thus the capacity varies to the same degree. Additionally, the probability of data loss is higher by several orders of magnitude in comparison to wired networks, which in turn leads to a higher variability of the RTT, as most of the errors are handled by the HARQ in LTE. This leads to poor TCP performance in cellular networks and multiple attempts to improve it have been performed.

One way how network operators deal with these problems is the introduction of Performance Enhancement Proxys (PEPs) in the path between the end hosts [FHN12; RFC 3135]. Although the location can be chosen freely, a PEP is often located between P-GW and the Internet uplink [Hua+13]. A PEP splits the TCP connection between server and UE into two connections. One connection between server and PEP and one between PEP and UE. Thereby, the PEP is operating transparently, such that neither UE nor server are aware of it. The benefit of splitting the connection into two connections is the reduction of the RTT for each connection. This allows each connection to recover more quickly from congestion and transmission errors on the wireless link are not propagated to the connection between server and PEP. Additionally, operators can tune the TCP parameters for the connection between UE and PEP to the special requirements of wireless channels.

Li, Chung, and Jiang [LCJ17] compare the performance of different TCP congestion control algorithms in an LTE network while driving on a highway. The main finding is that TCP does not react fast enough to changing link capacities and therefore available capacity is wasted. In some cases it takes up to 4 s for TCP to adjust the sending rate. Due to the variability of the channel, the available link capacity might change again, even before TCP was able to adapt the rate.

Becker, Rizk, and Fidler [BRF14] measure UDP and TCP performance in LTE networks for non-moving UEs. A remarkable finding is that the delays in uplink and downlink direction are significantly different. However, it remains unclear why this effect occurs. While the median delay in the uplink is approximately 30 ms, it is only 10 ms in the downlink. Thus, TCP is able to adapt the congestion window approximately every 40 ms. Besides that, the authors demonstrate that the LTE MAC layer impairs the TCP performance. E.g., the Discontinuous Reception (DRX) feature, which reduces the UE energy consumption by monitoring the control channels only at specific cycles, has an impact on the RTT. Depending on the RRC state, different cycle durations are configurable. In `RRC_IDLE` the DRX cycle ranges from 320 ms to 2560 ms and in `RRC_CONNECTED` the short DRX cycle ranges from 2 ms to 640 ms and the long DRX cycle from 10 ms to 2560 ms [Cox14]. Thus, the cycles are in a similar range as the normal RTT and could therefore affect the TCP behavior. A second impact stems from the HARQ retransmission protocol. If the receiver, i.e., UE or BS, fails to decode the transmitted data, a retransmission is initiated 8 ms after the initial transmission, which is in a similar range as the best case RTT of LTE.

Müller [Mül11] analyzes the influence of TCP on the performance gains achieved by CoMP. The author assumes that it takes some time until coordination between BSs is established for a UE. Until this has happened, the UE is served by its anchor BS only. Consequently, it might be beneficial to pause a transmission that is still in the TCP slow-start phase and only transmit the data if the coordination has been established. The reason is that the TCP connection reaches its maximum throughput more quickly, if the higher bandwidth achieved by CoMP is directly available in the slow-start phase. On the opposite, TCP reacts more slowly to changing bandwidths in the congestion avoidance phase. Therefore, the total duration to transmit a certain amount of data might be shorter even though the beginning of the transmission is delayed.

3.1.2.2 Application Behavior

The wide variety of applications renders it impossible to cover all of them or even a reasonable selection in detail. Instead, we organize them in three categories, which cover most of today's applications. Web browsing traffic comprises traditional web surfing but also traffic generated by many smartphone applications. Streaming media covers buffered audio and video streaming. Finally, real-time communication comprises all interactive audio or video communications. In terms of traffic volume streaming media and real-time communication represent the largest share [San16]. In the following, we describe these categories in more detail and highlight the dynamics of the generated traffic. Newly emerging applications in the context of IoT and MTC can be categorized as web browsing traffic or real-time communication, as their basic communication patterns are very similar. IoT applications often follow a request-response scheme, e.g., for exchanging sensor data or control commands. MTC applications, e.g., machine control applications, show a real-time communication behavior. However, the requirements in comparison to human communication might be higher, because some MTC applications require higher reliability and lower latencies.

Web Browsing Traffic

Web browsing traffic includes a broad range of applications, which are characterized by requests issued by the client, i.e., a user or a machine, and responses generated by a server. This application class contains web browsing, file downloading, emailing and also automatic software updates. Often the HyperText Transfer Protocol (HTTP) is used as application protocol, while TCP is the typically used transport protocol. For the transmission of application objects multiple transport layer connections can be used. E.g., in the case of a website, the browser might use multiple TCP connections to transmit the objects of the website, like embedded images or scripts.

Web browsing traffic is elastic traffic, which completely uses the available bandwidth. Thus, the time to finish a transmission is determined by the available data rate and the size of the application object. Another characteristic of web browsing traffic is its burstiness, which means that phases of activity alternate with idle phases. E.g., during a web browsing session the user downloads a website first and reads it afterwards until another website is accessed. Long-term measurements of the most popular websites show that their average sizes are ever-increasing and now almost reach a size of 3 MB [HTTP Archive]. However, these measurements are not specific for mobile optimized websites, so the average sizes of websites in cellular networks could be smaller. File sizes of automatic software updates might be significantly larger. Own measurements of apps available in the iOS and Android app stores reveal average app sizes of 55.26 MB and 17.08 MB, respectively. Appendix B provides more details of the performed measurements.

The exact behavior of how and when the traffic objects are generated is highly volatile and might change from one day to the other. Also, new applications may emerge while others lose popularity. The application behavior is influenced by many factors like the software version, the Operating System (OS) and even the underlying hardware. Therefore, it is not possible to globally describe the behavior.

Performance issues or degradation for applications belonging to the web browsing traffic category are experienced if the application does not respect the characteristics of cellular networks. Huang et al. [Hua+13] show an example of an application, which downloads a 30 s music sample with a size of 1 MB using a single TCP connection. Even if the download could have been finished within 2.5 s in the ideal case, it takes 9 s due to the following two reasons. First, the advertised TCP receive window is comparatively small (in this case 131.8 kB) and second, the application does not read the received data from the receive buffer fast enough to allow the transmitter to send data continuously. While the first reason could be resolved by a more appropriate TCP configuration, the second reason is a more general problem of the application or the OS.

Streaming Media

Audio and video streaming plays an important role in today's cellular networks. The popular video streaming platform YouTube reported at the end of 2016 that mobile users spent in average 40 min per session watching videos [YT2016]. Besides YouTube, other prominent examples for video streaming platforms, like Netflix or Amazon Video, exist. Although we treat streaming media as a separate category, the same protocols as for web browsing applications can be used. E.g., the above mentioned video streaming platforms use Dynamic Adaptive Streaming over HTTP (DASH) and TCP to deliver their content to the users. Due to the nature of streaming media applications, the majority of the traffic is caused in the downstream direction, while the upstream is mainly used for requests and acknowledgments.

The main feature of streaming media is that the content is not directly played out, but it is buffered at the client first. This allows reacting to varying bandwidths on the one hand. On the other hand, the user can pause the playback at any time. The media is usually delivered by splitting it into smaller chunks and transmitting them as required to maintain a certain buffer level. E.g., YouTube uses chunks of 64 kB and transmits more data at the beginning of a new video to fill up the initial buffer [Ram+14]. The Netflix smartphone app requests chunks of 1 MB to 4 MB in regular intervals of 10 s [Hua+13]. Due to this characteristic, streaming media can be considered as partially elastic traffic. During the transmission of a chunk, the application is able to use the available bandwidth. However, the long-term data rate is limited. Measurements performed in a nationwide cellular network revealed that the median video duration is around 30 s, which corresponds to a median video size of 3 MB [Cas+14]. Because these measurements are relatively old, the bandwidth requirements of streaming media might have increased in the meantime.

Real-time Communication

The main difference between streaming media and real-time communication is that buffering is not applicable for real-time communication. Therefore, real-time communication represents inelastic traffic, which requires a certain bandwidth to operate satisfactory. Additional applications like gaming can also be treated as real-time communication. For this type of applications not only the bandwidth but also the latency plays an important role. E.g., in the LTE standardization, target packet delays of 50 ms to 150 ms are defined for real-time communication [3GPP 23.203]. The other traffic characteristics, i.e., bandwidth requirements and durations, are comparable to those of streaming media transmissions. However, the imbalance between uplink and downlink traffic is smaller for some real-time communication applications, e.g., for video or voice telephony.

3.1.2.3 User Behavior and Diurnal Load Profiles

On the largest considered timescale, the traffic characteristics are influenced by the varying user behavior throughout the day. Additionally, the traffic characteristics on weekdays are different from those of weekends. Load profiles on a BS level are also influenced by the location of the BS. E.g., a BS in a residential area shows a different traffic pattern than BSs in office areas, which is an effect caused by the commuting behavior of the users.

Wang et al. [Wan+15] provide detailed insights into the variation of the traffic load of 9600 BSs and 150 000 UEs collected in Shanghai. The authors show that the traffic loads of BSs located in different areas are related to each other. E.g., the evening peak in residential areas is approximately 3 h later than the peak load observed in areas used for transportation. Similarly, the peak in office regions is between the morning and evening peak of transportation areas. Trasarti et al. [Tra+15] observe similar relations in Paris. When the traffic load at the local airport increases by 10 %, 2 h later the load at the Gare de l'Est, where people coming from the airport by train arrive, also increases by 10 %.

The changing user behavior over the day has been observed by Zhang and Årvidsson [ZÅ12] and Xu et al. [Xu+11]. Both studies report that the smartphone usage is higher during the day compared to nighttime. However, not all applications show similar usage patterns. Weather and news applications are more frequently used in the morning, sports and gaming applications are more often used in the early evening and radio and entertainment applications are also used during the night. In comparison to wired users, the usage behavior of wireless users is more constant throughout the day.

Furthermore, the user generated data traffic is related to the user activity. E.g., the user behavior of a commuting user in public transportation is different from the behavior of a user driving a vehicle. Also, static users show a different behavior compared to moving users. Therefore, we treat the user mobility in the next section in more detail.

3.1.3 User Mobility

User mobility affects the performance in cellular networks on different timescales. As already discussed in Section 3.1.1, changing user locations influence the channel quality. Therefore, we discuss in Section 3.1.3.1, how individual users can be expected to move in urban scenarios. On larger timescales the macro mobility, discussed in Section 3.1.3.2, determines the distribution of users in the considered area. Here we observe diurnal changes of the user distribution, e.g., caused by daily commuting of users.

In the following, we focus the discussion of user mobility on urban scenarios, such that mainly vehicles and to some degree pedestrians are of interest. The effect of user mobility is not only relevant for network planning and optimization, but originally comes from the field of vehicular traffic or transportation engineering. For more details refer to [Leu88] and [GMR01].

3.1.3.1 Micro Mobility

Micro mobility describes the movements of a single user or a small group of users. The movement of a single user is described by location and velocity. By expanding the area of interest, further mobility characteristics can be derived, e.g., the local density of users or the correlation between users. The density determines the number of users per length of road or area.[2] If users move in the same direction with the same speed their movement can be regarded as correlated, whereas random travel directions and speeds would result in no correlation.

User Behavior

Vehicles in a traffic stream can be seen as discrete objects, which are loosely coupled with each other by the drivers reactions and actions [GMR01, chapter 3.5]. This can be illustrated by the following examples. Within cities similar effects occur in front of traffic lights, where a group of vehicles piles up until they can continue their journey. Because the probability to follow the main street after a junction or traffic light is higher than

[2] While it is more common in transport engineering to define the density as the number of vehicles per length of road, for the evaluation of cellular networks the definition according to the number of users per area is more suitable. The reason is that users are not restricted to roads, but can in general be located everywhere.

following a smaller street, the vehicles will also travel in groups when they continue their journey. This behavior leads to a correlated movement. A second example is observed on single lane highways, where a group of vehicles piles up behind a moving bottleneck, like a slower car or truck. If possible, the queued vehicles will overtake the moving bottleneck.[3] This again leads to a correlated movement for the group of vehicles behind the bottleneck, as they travel with similar speeds in the same direction.

Average Speeds

In most regions and countries speed limits regulate the traffic. For cities a speed limit of 50 km/h is used in many European countries. Exceptions of 30 km/h or higher speed limits can be found for certain roads. Thus, the average speeds are upper bounded even if not all drivers comply to the limits. Kim, Sridhara, and Bohacek [KSB09] report that the ratio of the actual speed to the speed limit can be represented by a Gaussian distribution with $\mu = 0.78$ and $\sigma = 0.26$ (see Figure 3.9). The deviation between speed limit and actual speed stems from psychological factors as well as traffic circumstances that do not allow higher speeds. The data for this evaluation was originally gathered by Du and Aultman-Hall [DA04] in Lexington, Kentucky, which represents a medium-sized city of approximately 250 000 inhabitants.

Çolak, Lima, and González [ÇLG16] provide speed distributions for five cities in North America, South America and Portugal. The authors derive average free flow speeds of 50.4 km/h and 36.6 km/h during rush hour traffic conditions, which is a significant decrease of about 27.4 %. For the dimensioning of new roads Köhler [Köh14] reports in accordance to the German Road and Traffic Engineering Society (Forschungsgesellschaft für Straßen- und Verkehrswesen) desired minimum speeds for different road categories. For example on inner city main streets speeds between 30 km/h and 60 km/h should be achievable. To compare these values with actual achievable speeds, we used the Google Directions Application Programming Interface (API) to obtain average speeds in 35 of the largest German cities. On the 13 985 randomly selected routes the average measured speed is 24.96 km/h. Appendix C shows more details of the performed measurements.

Density

In transport engineering the fundamental diagram of traffic flow describes the relation between volume, i.e., the number of vehicles using a street segment per hour, and density, i.e., the number of vehicles per length of street. The fundamental diagram is based on a traffic model that describes the relation between vehicle speed, traffic volume and vehicle density. Speed and density are directly related to each other. The basic rule states that the higher the vehicle density is, the slower the vehicles move.

[3] More formally this example can be described and solved with the methods of queuing theory in form of a M/G/1 system [Leu88, chapter II.3.2.1]. The arrival process is Markovian and the rate is determined by the speed difference of the moving bottleneck and the approaching vehicles. The service process is of general nature, as the possibilities of a queued vehicle to overtake are influenced by multiple aspects, e.g., by the opposing traffic or the street conditions. As only one vehicle can overtake per time, the number of servers is one.

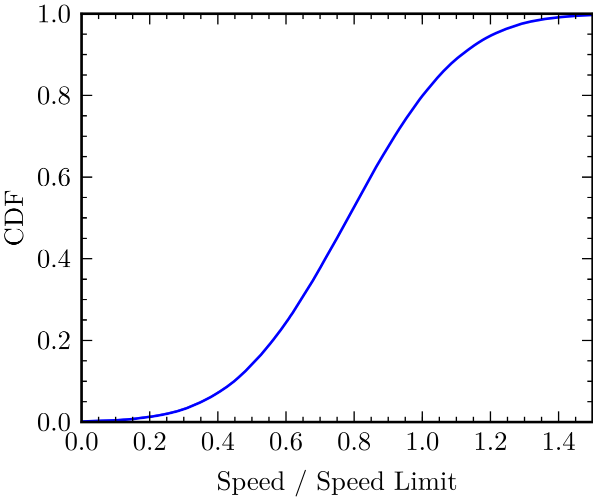


Figure 3.9: Distribution of the ratio between actual driver speed and speed limit according to Kim, Sridhara, and Bohacek [KSB09]

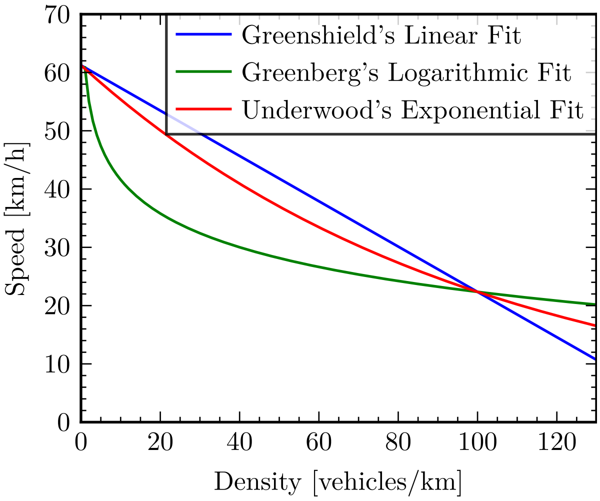


Figure 3.10: Relation of speed and density according to Greenberg [Gre59] and O'Flaherty et al. [OFl+96]

O'Flaherty et al. [OFl+96] present an overview of different studies on the relation between speed and vehicle density. This relation is either derived from traffic theory or is based on actual measurements. Huber [Hub57] reports a linear relation between vehicle speed and density. Besides the linear relation, logarithmic and exponential fits have been suggested, too (refer to Greenberg [Gre59] and the overview by O'Flaherty et al. [OFl+96]). Figure 3.10 shows different relations between vehicle speed and density, proposed by three traffic researchers.

From these observations, we can conclude that the density in urban streets is typically below 30 vehicles/km, which equals a spacing of approximately 33 m between two successive vehicles. However, this average density reveals no insights in how the vehicles are distributed. As we have stated earlier, vehicles are loosely coupled and thus show a group behavior, such that the local density within a group is higher, while the distance between groups might be larger.

3.1.3.2 Macro Mobility[4]

So far we dealt with the movement of individual users. In this section, we consider the movement of all users on a higher level. This provides insights into the distribution of users in the considered area. Also, the timescales are larger and are now in the range of hours to days.

To gather this information, several approaches exist, e.g., by performing surveys, manual vehicle counting or using traffic surveillance cameras. In the recent past, the usage of cell phone data became appealing, because cell phones are ubiquitous and directly related

[4] In traffic and transportation theory the term macro mobility is used if traffic flows are described by sets of differential equations analogously to fluid dynamics. We use this term in the sense that the mobility and location of the individual users are not of interest, but instead the overall distribution of the users in the area of interest is considered.

to their owners [Rat+06; Bec+11; Lou+14]. These studies suggest reusing Call Detail Records (CDRs) to track the movement of individual users. A CDR is generated whenever a voice call or Short Message Service (SMS) message is sent or received by a cell phone. Besides billing information, the CDR also contains the IDs of cell phone and serving cell, and thus allows to coarsely locate the user.

The macro mobility is closely related to the diurnal traffic load profiles in the cellular network, which we discussed in Section 3.1.2.3. However, using CDRs to track the movement of individual users provides more insight in commuting or traveling behavior. E.g., Becker et al. [Bec+11] identify workers and their commuting behavior using CDR data. The authors define users as workers who are located in business districts during business hours, i.e., during Monday till Friday from 9am to 5pm. By evaluating the call and SMS densities the variation of the user distribution throughout the day can be observed [Rat+06]. The user density in the suburban districts is higher from early evening to early morning, while the opposite is the case for business districts.

3.2 Handling of Dynamics

In cellular networks like LTE, the network adapts to dynamic effects on the smallest timescale by adapting the resource allocation, i.e., scheduling including link adaptation, according to the current conditions. We discuss this in Section 3.2.1. On a longer timescale the concept of Self Organizing Networks (SON) adapts to various dynamic effects. SON concepts are introduced in Section 3.2.2. Long-term dynamics are handled by manual network planning, discussed in Section 3.2.3.

3.2.1 Scheduling

In the context of cellular networks, scheduling, sometimes also called dynamic resource allocation, denotes the process of assigning radio resources to queued data to transmit it over the wireless link. This is achieved by internally organizing the waiting data in multiple queues, where a dedicated queue represents an individual user, a data bearer or an application. Scheduling is often accompanied with link adaptation, which selects an appropriate modulation and coding scheme. In LTE networks the scheduler operates on a data bearer level and determines the resource allocation every TTI of 1 ms by assigning RBs to the data bearers. Additionally, the scheduler selects the best suitable of the 16 predefined Modulation and Coding Scheme (MCS) (see Section 2.1.3).

3.2.1.1 Scheduling Principles

Although scheduling is not part of the LTE standardization and the actual implementation is left to the vendors, some common principles can be identified, which we briefly introduce in the following. Basically it can be distinguished between channel aware and non-channel aware scheduling. Channel aware scheduling exploits the channel diversity on different frequencies and for different users, to increase the spectral efficiency by serving the queues on appropriate resources.

Round Robin (RR) scheduling is a simplistic non-channel aware principle as it uses no external information and assigns radio resources to the queued data in a circular order. E.g., the first radio resource is assigned to the first queue, the second radio resource is assigned to the second queue and so on. RR schedulers achieve resource fairness between the queues, because on a longer timescale each queue gets an equal share of the available radio resources. However, due to the variations of the channel quality, the channel capacity or the data rate might not be equal for all queues.

The maximum throughput scheduling principle represents the simplest channel aware principle. Sometimes this scheduling variant is also called Max C/I, where C/I stands for the ratio between carrier and interference, which is in line with the definition of the SINR in this thesis. As the name suggest, this type of scheduler maximizes the total system throughput, but does not guarantee any fairness between the queues. To achieve the goal, a Max C/I scheduler assigns radio resources to the queue with best channel quality on the given radio resource. Therefore, it can happen that queues with bad channel quality are never scheduled. This effect is called scheduling starvation and is generally not desired.

Proportional Fair (PF) scheduling combines the goals of the two previous schedulers to improve the system throughput but at the same time ensures fairness between the queues by preventing scheduling starvation. The concept of proportional fairness was first introduced by Kelly [Kel97]. Mathematically this is achieved by maximizing the sum of the logarithms of the average rates for each queue $\overline{r}_q$:

$$\text{maximize} \sum_q \log(\overline{r}_q) \tag{3.5}$$

By maximizing the sum of the logarithms, not scheduling a queue with high rate can be compensated to some degree by instead scheduling a queue with low rate. Thus, this principle ensures that queues with low rates are also scheduled, but still the higher rates of queues with better channel conditions are exploited. An algorithm to solve this optimization problem has been introduced by Jalali, Padovani, and Pankaj [JPP00]. We discuss this algorithm when we introduce the implementation in the developed simulation model in Section 5.1.7. Margolies et al. [Mar+16] state that PF schedulers are the most common scheduling principle in today's cellular networks.

These three scheduling principles are agnostic to the type of data they handle, which is only reasonable if all data has equal properties. However, this is usually not the case in cellular networks, as the data is generated by different applications with different requirements (for an overview refer to Section 3.1.2.2). Also, the users have different expectations how their data should be handled. Therefore, extensions to the classical scheduling principles have been developed. E.g., Andrews et al. [And+01] introduce a scheduler that is able to handle users with real-time requirements without violating their delay requirements. Another approach is to distinguish the data with respect to its requirements [Nec06]. On a higher layer the scheduler could be provided with information about traffic objects or transactions. Then the scheduler can determine the order in which these transactions are transfered, such that the Quality of Experience (QoE) for the users is maximized [Pro+12].

3.2.1.2 Influences on Scheduling

As scheduling is usually performed at discrete time intervals, e.g., in intervals of 1 ms in LTE, the time the scheduler can react to dynamic effects is lower bounded. On the other hand, the scheduler influences the characteristics of the transmitted data on a timescale of multiples of the scheduling interval, because data is only transmitted if the scheduler decides to transmit it. Other mechanisms like the TCP congestion control react to the scheduling decisions. We discuss the influences of network dynamics on the scheduling behavior in the following.

Channel Knowledge

Because the mostly used schedulers in cellular networks are channel aware, they require accurate knowledge of the channel states. As discussed in Section 2.6, channel measurements introduce overhead and are therefore only performed in intervals with low frequency or upon request. This further increases the reaction time of the scheduler to the range of the CSI reporting time. E.g., in LTE the transmission of the required reference symbols for downlink channel measurement is restricted to intervals of 5 ms to 80 ms. For aperiodic CSI reports, the scheduler has to trigger the UE first, before the current channel conditions are measured and signaled to the scheduler. As we have seen in Section 3.1.1.1, the coherence time of vehicular users could be shorter than the channel measurement period, such that channel aware scheduling is not meaningful.

Traffic Characteristics

Another aspect is the influence of the traffic characteristics on the scheduling. A common assumption for the design and evaluation of scheduling algorithms is a full-buffer scenario where all queues always have data waiting for transmission. However, this is not the case in real systems and the traffic characteristics are influenced by transport protocol, application and user behavior, as discussed in Section 3.1.2. Especially the interaction of TCP and scheduling have been treated in the literature [Zho+13; SGS14; Grø+17].

The main issue between the interaction of TCP and scheduling is that TCP adapts the sending rate according to the perceived link capacity. The link capacity in turn is controlled by the scheduler and depends on the actual channel conditions. Fast fading changes the channel conditions on timescales in the order of milliseconds, which might cause the TCP transmitter to adapt the sending rate. In contrast to wired networks the changing capacities in cellular networks are normal, but TCP is not designed to handle such conditions.

Another aspect is the burstiness of the traffic, which we discussed for web browsing traffic and streaming media. As a consequence, the scheduler must handle situations where application objects of few kilobytes to several megabytes arrive in short time after a longer idle time. If the scheduler does not react quickly to an arriving burst, it will take the TCP transmitter longer to increase the sending rate, which in turn increases the transmission time of the application object.

3.2.2 Self Organizing Networks

The concept of Self Organizing Networks (SON)[5] refers to a variety of techniques to simplify and optimize the operation of cellular networks by automating time-consuming tasks that had to be performed manually [MF11, chapter 7.2.1]. We can identify three major areas where SON concepts are used [Ali+13]. Self-configuration allows network devices to automatically configure themselves after they are switched on or inserted into the network. With self-optimization we describe the ability to automatically tune system parameters to increase the system performance. Self-healing capabilities enhance the resilience and allow the network to react to failures and reconfigure itself such that the network is still operational.

Because self-configuration is only carried out when new devices enter the network, we focus the following discussion on self-optimization. Self-healing can also be seen as a part of self-optimization, as both approaches try to maximize the system performance under the given circumstances. According to Aliu et al. [Ali+13] the purpose of self-optimization is to perform load balancing, interference control, coverage extension or capacity optimization. The operational timescale of SON ranges from seconds up to days or even months.[6] E.g., in the range of minutes, SON reacts to medium-term variations of the radio channel, shadowing or temporary user hotspots by adapting the antenna configuration, like the electronic tilt. If the variations are more permanent, e.g., if new buildings influence the shadowing or new user hotspots arise, SON can rebalance the load between BSs to improve coverage and performance. This can be achieved by adapting the handover conditions or even forcing handovers from an overloaded BS to a less loaded one. Because this requires measurements of load and channel conditions, this aspect of SON can only be adapted in the range of hours to days. The reorganization of BSs works on even longer timescales. E.g., this includes adaptive sectorization algorithms or the assignment of radio resources to BSs. Dynamic clustering for CoMP can be seen as an aspect of SON reacting to medium-term dynamics. The goals of dynamic clustering are similar to those of SON, as in both cases the network is reconfigured such that the overall network performance for the current condition is improved.

With the rise of big data analytics in the recent past, research on big data assisted SON has gained momentum [IZA14; Bas+17]. In contrast to traditional SON concepts, which mainly work reactively, big data analytics allows predicting future network conditions, such that the network can adapt proactively. Big data in the context of cellular networks refers to different information sources, which are already available, but not used for proactive network reconfiguration. This could be subscriber level data like CDRs, cell level data like resource utilization within a cell or core network level data like aggregated statistics of network performance metrics. This can be assisted by external data like timetables of public transportation or social media feeds. By combining the gathered information, the prediction of the wireless channel, the user mobility or the user behavior becomes possible.

[5] This term is not clearly defined. Other authors also use the term Self Optimizing Network.

[6] Sometimes scheduling is also considered as part of SON and thus the timescales range down to milliseconds. However, we treat scheduling separately.

3.2.3 Network Planning

The network planning process is a highly manual process with the goal of defining the network architecture and configuration. It can be carried out when a new network technology is deployed or when an existing network is extended. According to Mishra [Mis07] it consists of the steps pre-planning, planning, detailed planning, verification and optimization.

The pre-planning phase covers the dimensioning of the network, including the derivation of the requirements. E.g., the network has to provide a certain coverage to fulfill QoS and QoE requirements like data rate, latency or reliability. Also, the network must be dimensioned in order to serve a certain number of subscribers. At the same time the necessary radio bandwidth has to be derived. For the dimensioning, expectations about future demands and its development should be taken into account. The focus of the planning phase is to determine the BS locations, which allow fulfilling coverage requirements determined in the previous step. The coverage planning is supported by tools, which take the real topography into account and are able to calculate signal propagation and interference conditions. This step also includes antenna configuration, e.g., the type of antennas or their orientation. In the detailed planning phase the actual configuration and parameterization of the network is determined. E.g., neighbor definitions to allow handovers between BSs, the BS internal scheduler or the used radio resources have to be configured. If CoMP is possible in the network, the definition of a static clustering can be seen as a task during the network planning process. The verification phase has the aim to find errors that occurred in the previous steps, before the newly designed network is installed and goes operational. This could include drive tests to assess if the network meets the targeted design goals. After the launch of the network, the performance and status is permanently monitored, which is referred to as optimization phase. In case of any abnormalities the operator can react and change the network configuration. In this phase also smaller migrations can be performed, which do not require a complete planning cycle.

As can be seen, network planning is a highly complex task and is therefore carried out only infrequently. Especially, it is not possible to react rapidly to changing demands or network conditions.

3.3 Discussion

An illustration of the different effects and how cellular networks cope with them is depicted in Figure 3.11. The causes of dynamic effects are shown above the time axis and the mechanisms to cope with the dynamics are shown below. Note that the depicted timescales are not meant as absolute values but rather indicate the order of magnitude. To complete the picture, Figure 3.11 also contains dynamic clustering for CoMP, which was not treated in this chapter.

We can see that fast fading and partially effects caused by network protocols are already treated by scheduling. Mobility and long-term traffic characteristics can be handled by various SON concepts. Finally, the longest-term mobility and traffic characteristics are

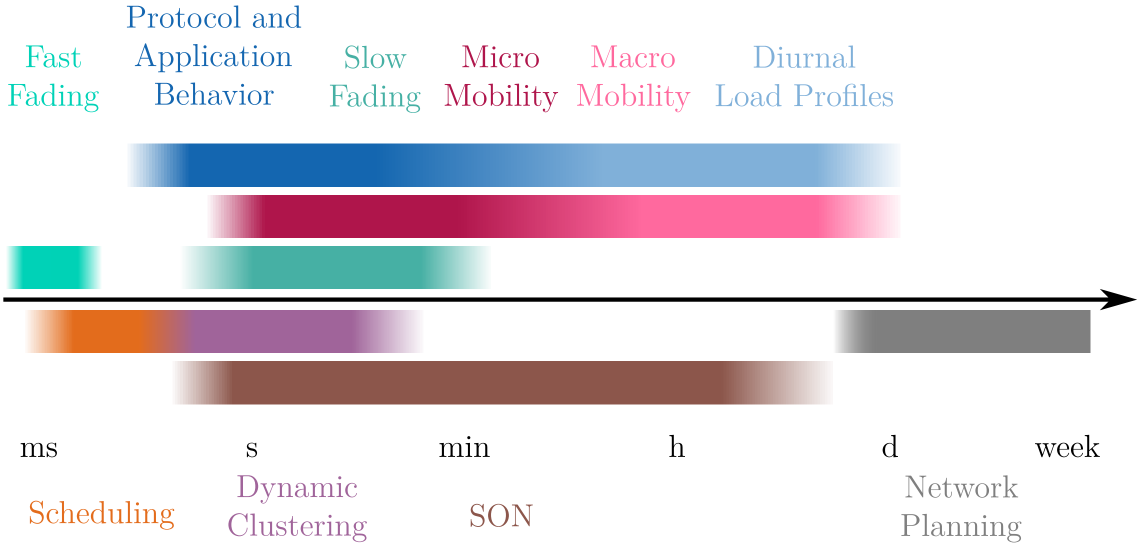


Figure 3.11: Timescale of different dynamic effects

handled by network planning. Dynamic clustering operates on timescales above those of scheduling and it is suitable to deal with dynamics caused by application behavior and user movement. This hints that scheduling and dynamic clustering should interact to allow an adaption of the network to its dynamics on small to medium timescales. We use these findings in the next chapter to develop a suitable dynamic clustering mechanism for CoMP, which takes the occurring network dynamics into account.

4 Dynamic Cloud Clustering and Scheduling Framework (DCCSF)

In this chapter, we introduce the Dynamic Cloud Clustering and Scheduling Framework (DCCSF), which is developed and evaluated in this thesis. DCCSF solves the dynamic clustering problem with an algorithm executing a heuristic. Additionally, it reduces the scheduling effort, which allows introducing CoMP in existing cellular networks. The basic concept and first evaluation results of DCCSF have already been published [Sch17]. We state the design goals in Section 4.1, before Section 4.2 presents the overall system architecture in which DCCSF is implemented. The actual dynamic clustering algorithm, which is a fundamental part of DCCSF, is presented in Section 4.3. Section 4.4 shows how scheduling is included in DCCSF. Finally, Section 4.6 concludes this chapter with a classification of DCCSF and a comparison with other dynamic clustering algorithms.

4.1 Design Goals and Limitations

The main design goal is to develop a solution for the integration of CoMP into existing cellular networks. This is solved by a practical procedure to define and configure clusters of Transmission Points (TPs). This design goal introduces several restrictions and requirements. The most obvious is that the clustering procedure must integrate seamlessly into the existing system, which means the necessary changes should be kept as minimal as possible. Also, the process of defining and configuring clusters shall not disturb normal system operation, e.g., by data loss. As we have seen in the previous chapter, multiple dynamic effects change the network conditions quickly, such that the cluster definition process should dynamically adapt to the current conditions and should be able to operate on similar timescales. Especially the mobility of users should be considered when determining the clusters. Additionally, in real networks the information about the system state is incomplete and to some degree inaccurate, as the overhead of gathering the information prohibits complete system state information [PGH08; Bas+17]. These requirements render some proposed methods to generate the dynamic clustering infeasible. Exhaustive search or optimization methods are hardly possible, because complete knowledge of the system state is required or the complexity is too high. Therefore, we design an algorithm to perform the task of dynamic clustering. By carefully designing the algorithm the overhead is kept small. Additionally, the algorithm is developed such that it can deal with partial information of the system state. The goal of the algorithm is not to find the optimal clustering, but instead to find a reasonably good clustering in reasonable time. The integration of the algorithm in the system operation is handled by executing the algorithm in regular intervals. These intervals are called cluster reconfiguration intervals with a periodicity of T_R. Thereby, the trade-off between overhead and complexity of the cluster reconfigurations on the one side and the benefits of frequent reconfigurations on the other side is adjustable.

Further design goals stem from the fact that the dynamic clustering algorithm shall be integrated into existing networks. We assume especially that the TP positions are already fixed and it is not possible to change them in order to improve the network topology for CoMP. This is reasoned by the high cost and barriers of acquisition of new sites. Also, we consider the locations of TPs in real networks to be already optimized for coverage and throughput during the network planning process. However, we assume a C-RAN architecture as the basis, as this is often seen as the most promising architecture of future cellular networks. Another reason supporting these assumptions is that it might be easier to reuse existing sites and deploy Remote Radio Heads (RRHs) and a high capacity fronthaul network than to acquire new sites.

The C-RAN architecture allows to deploy a Cluster Manager (CM) in the central office, which collects all required information and executes the dynamic clustering algorithm. So the algorithm design is only focusing on a centralized approach. As we have seen in Section 2.2, Joint Transmission (JT) is superior in comparison to Coordinated Scheduling/Coordinated Beamforming (CS/CB)- or Dynamic Point Selection (DPS)-based CoMP. Therefore, the algorithm only supports JT. Additionally, a centralized approach can mitigate the drawbacks of JT, because tight synchronization and data exchange become feasible between the Baseband Units (BBUs) located in the central office.

Because we assume an existing network as the basis, it is not easily possible to introduce new signaling interfaces or measurement possibilities to obtain the system state in the CM. Existing system capabilities should therefore be used where possible and necessary changes should be minimal. However, this applies only for the parts of the system that are covered by standards, e.g., interfaces and protocols between UE, eNodeB and the EPC in LTE. As the architecture of the central office of C-RANs is not covered by any standards, we assume to be relatively free in the architectural design of this system part, while still using reasonable assumptions. The actual implementation within the C-RAN architecture is not part of this thesis. So we do not consider possible hardware or software technologies or the dimensioning of the C-RAN in terms of processing capacity or bandwidth.

The dynamic clustering algorithm is designed to determine non-overlapping clusters, i.e., each TP belongs to exactly one cluster. We show in the next section that this choice is beneficial to integrate CoMP into C-RAN architectures. We already discussed the drawbacks and advantages of overlapping and non-overlapping clusters more generally in Sections 2.3.1 and 2.4. Also, the clustering process should be tightly coupled with resource allocation, because the configured clusters determine which UEs can be scheduled. If goals of clustering and scheduling are not aligned, both processes might work against each other, which should be avoided. This is for example the case if the clustering process aims to maximize the sum capacity of the whole network, while the scheduling aims to achieve fairness between UEs. Therefore, the dynamic clustering algorithm determines the clusters using the same metrics as the scheduler. Non-overlapping clusters also have the benefit that scheduling can be performed for each cluster individually, such that no information has to be exchanged between the schedulers. Because scheduling for CoMP is a complex task, this issue should be addressed when introducing CoMP in cellular networks.

4.2 System Architecture

In this section, we introduce the system architecture in which DCCSF is implemented. Therefore, we first show the general system in Section 4.2.1. Then the necessary steps to gather the information required by the CM to execute the dynamic clustering algorithm are introduced in Section 4.2.2. Section 4.2.3 shows the necessary steps to configure a new clustering. Finally, Section 4.2.4 presents the integration of the dynamic clustering algorithm in the system operation.

4.2.1 Overall Architecture

The overall system architecture follows a typical C-RAN design and is depicted in Figure 4.1. The system comprises RRHs distributed in the field and BBUs located in a central office. The central office is then connected to the EPC. Additionally, the CM, which executes the dynamic clustering algorithm, is deployed in the central office. Similar to the approach from Haberland et al. [Hab+13], we make use of a fine-grained design, which consists of a multitude of small software components. These software components could for example be realized as Virtual Machines (VMs), Operating System (OS) containers or specific applications, depending on the required degree of isolation and the capabilities of the underlying infrastructure. Software components exist e.g., for signal and protocol processing or scheduling. Also, each UE is represented as a software component in the central office by its UE state. The CM is realized as such a software component, too. To transmit data towards a UE, the BBUs execute the appropriate software components using the corresponding UE state. This fine-grained system organization simplifies the reconfiguration of clusters, because in order to reconfigure the clustering, new software instances have to be launched and connected accordingly. The necessary steps to reconfigure the clustering are discussed in Section 4.2.3. In Figure 4.1, the UE state is illustrated by a queue, while in fact the UE state comprises all information required to serve the UE. This is besides queued data in the RLC-buffer also information like Channel State Information (CSI) reports, information about the Discontinuous Reception (DRX) state or uplink scheduling related information like Buffer Status Reports (BSRs) and Power Headroom Reports (PHRs)[1]. To prevent data loss during a cluster reconfiguration also the state of the HARQ processes could be included, which is not the case in a traditional handover in LTE networks.

As visible in Figure 4.1, multiple RRHs, which represent the actual TPs in a C-RAN, are grouped into CoMP clusters. In the central office multiple BBUs are grouped, too, to perform all necessary processing for a cluster. Each group of BBUs is accompanied by the UE states of the UEs served by the cluster and a scheduler that performs the resource allocation within the cluster. The group of BBUs, UE states and scheduler is then called Cluster Entity (CE). Even if not depicted, the necessary software components for signal and protocol processing for each cluster also belong to the CEs. The CEs are pure virtual

[1] In LTE, the UE notifies the eNodeB about buffered data waiting for uplink transmission with a Buffer Status Report (BSR). The Power Headroom Report (PHR) is used to report the available transmit power of the UE to the uplink scheduler, which is required to select an appropriate Modulation and Coding Scheme (MCS).

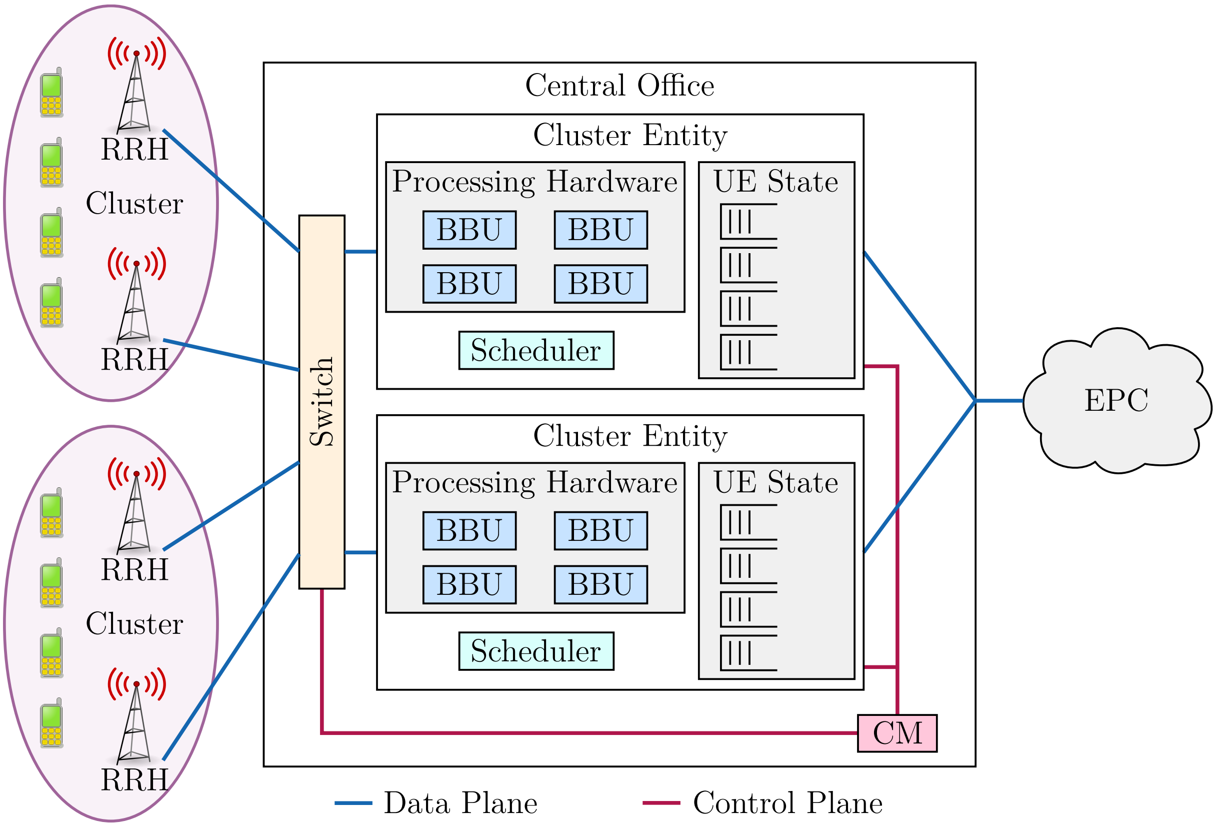


Figure 4.1: DCCSF system architecture

entities, as they only group hardware and software components. They are introduced to simplify the management of the software components within the central office, but do not carry out any functionality to transmit data. The number of BBUs associated with a CE is variable and depends on the expected processing effort required for the cluster. The processing effort depends on the cluster size as well as the traffic load of the cluster [Wer+15; SG16]. But because we do not consider the effect of processing effort in this thesis, we assume that always sufficient processing capacity is available. This assumption is also in line with the dimensioning of current cellular networks, where the processing capacity is dimensioned to the peak effort.

The CM controls the association of RRHs to CEs. This is possible, because RRHs are not directly connected to corresponding BBUs, but they are connected to a central switch inside the central office. The switch provides a direct data plane connection between RRH and BBU. The CM is connected via a control plane interface with the switch. As we do not consider implementation technologies, we simplify the switch as a single entity. In real-world systems the functionality of the switch could also be realized by multiple switching or routing hierarchies. The actual type of the switch depends on the technology used for the fronthaul network. In the case of an Ethernet-based fronthaul, e.g. realized with IEEE P1914.3 [IEEE P1914.3], the switch could be a standard Ethernet switch. Also, for the common fronthaul standards Common Public Radio Interface (CPRI) [CPR15] and Open Base Station Architecture Initiative (OBSAI) [OBS06] a switch between RRHs and BBUs could be introduced.

The CM is connected with a second control plane interface to the CEs. This interface is needed to reconfigure the clustering. Additionally, the functionality of the CM could be extended in real implementation to allocate the required number of BBUs to the CEs. For both control plane interfaces of the CM existing SDN and Network Function Virtualization (NFV) techniques could be used [LC15; Ngu+17]. Here it should be noted that the separation in data plane and control plane in Figure 4.1 is from the view of DCCSF. So the depicted data plane connections transport LTE data plane and control plane traffic.

4.2.2 Obtaining the System State

Here we present how the CM is provided with the information about the current system state. Therefore, we present the necessary extensions of the existing signaling in LTE networks.[2] Figure 4.2 shows the simplified signaling framework. The system state should be updated once per cluster reconfiguration interval T_R. However, if no update is provided, the CM can still work with a cached copy of the previous update at the cost of an impaired network performance. The update of the system state can be accomplished either by periodic updates from the UEs or by polling the UEs once per reconfiguration interval. A similar mechanism is already available in LTE to gather information needed for handover decisions. Both variants have their own advantages and disadvantages, but for the functionality of DCCSF it is irrelevant how the system state is obtained. So in the following we assume that the UEs send periodic updates. The retrieval of the system state is performed in the following three steps.

First Step

All UEs measure the downlink Reference Signal Received Power (RSRP) for all TPs.[3] To do so, the mechanisms provided by the network to measure the RSRP are used. To reduce the overhead of RSRP measurements, it would be possible to perform these measurements for a subset of TPs only. E.g., for neighboring TPs of the currently serving TP or cluster. Similar simplifications are also applied in LTE handover measurements. These measurements only cover the long-term average channel quality, but not the variations caused by fast fading, which is sufficient for the purpose of dynamic clustering. The effects of fast fading can be handled by the scheduler.

Second Step

When the measurements have been performed and the RSRP of all TPs is known by a UE, the UE creates a Cluster Measurement Report (CMR) message m. The CMR message

[2] In LTE release 13 similar extensions have been introduced for the X2 application protocol. Instead of exchanging the information only between eNodeBs, the information could be exchanged between CEs and CM in the proposed system architecture.

[3] Although we only use the RSRP, similar indicators like the Reference Signal Received Quality (RSRQ), the Received Signal Strength Indicator (RSSI) or the Channel Quality Indicator (CQI) could be used, too. Which quality indicator is used depends on the provided options of the cellular network.

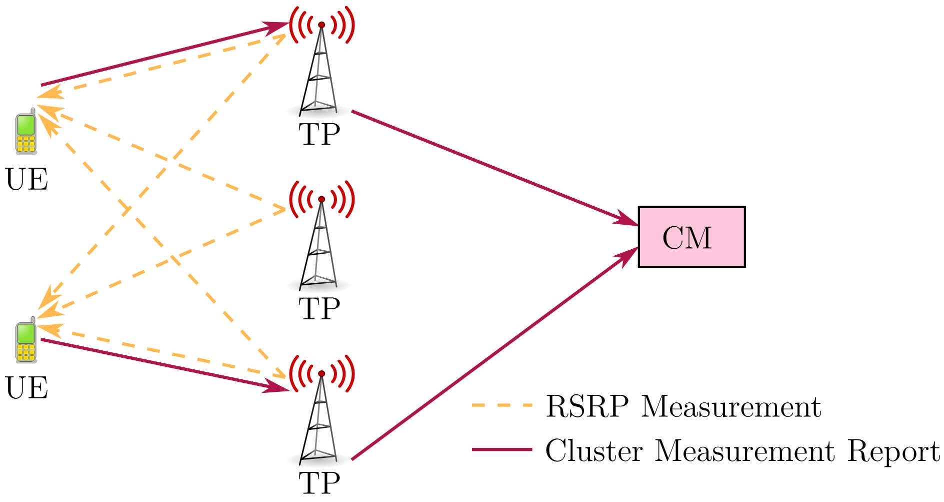


Figure 4.2: Simplified signaling of CMR messages

contains the field m.UE to store the identifier of the UE that issues the message. The message also contains a map of the $S_{C,\max}$ TPs with the highest RSRP and the respective RSRP values. This map is stored in $m.\boldsymbol{\mathcal{B}}$.

As an optional parameter the UE may include a weight, stored in $m.w$. The UE weight can be respected by the dynamic clustering algorithm when generating the clustering. One possibility to define the weight is to set it as inversely proportional to the averaged speed of the UE:

$$m.w = \frac{1}{v_u} \tag{4.1}$$

v_u is the speed of UE u. The idea behind the weight is that slowly moving or static UEs will profit for a longer duration if the CM configures a suitable clustering for the reporting UE. Highly mobile UEs leave the area where cooperation is beneficial or even the coverage area of the cluster more quickly and thus the total gain for the system is smaller. Therefore, slowly moving UEs should have higher weights. In case the speed cannot be measured, the weight is left uninitialized.[4] This can be detected by the CM and handled by performing network-based speed measurements or using a default value instead. E.g., the average speed reported by the other UEs could be used. In the following, we assume that the UEs can measure their speed and set the weight as defined in Equation (4.1).

Finally, the UE transmits the CMR message to its corresponding CE. If no CoMP scheme is used in the uplink, as depicted in Figure 4.2, then the UE transmits the CMR message towards its anchor TP, i.e., the TP with highest RSRP, which is also responsible for normal uplink data transmission. From the TP the CMR message is forwarded to the CE via the fronthaul network.

[4] Most modern devices communicating in cellular networks, e.g., smartphones or vehicles, are already equipped with sensors to measure speeds. Additionally, LTE provides several means to obtain the positions of UEs via the LTE Positioning Protocol (LPP) [3GPP 36.355]. Combining several position measurements also allows to derive the speed of a UE by the CM.

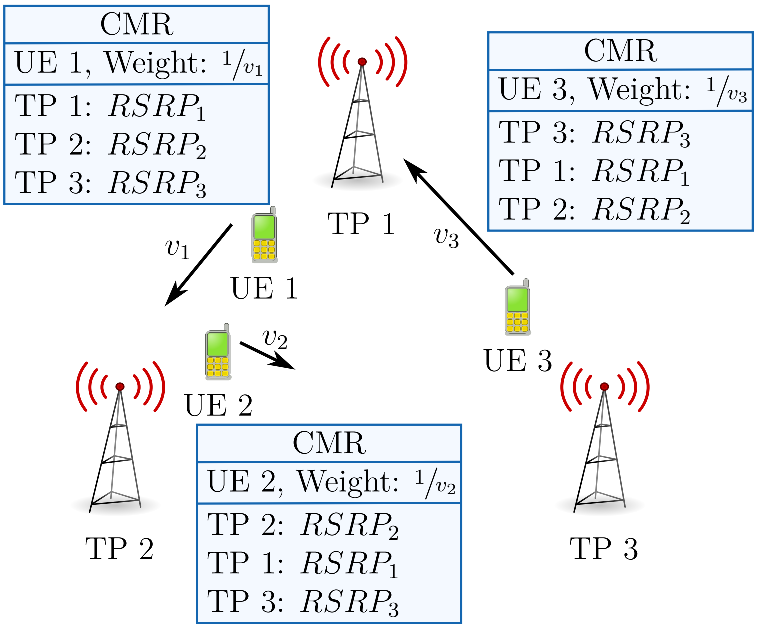


Figure 4.3: Example of CMR messages generated by three UEs in a network of three TPs

Third Step

In the last step, the CMR messages are transmitted from the CEs to the CM. This is done in the central office via the control plane interface between CEs and CM.

In comparison to existing signaling in LTE networks the extensions are minimal, as RSRP reports already exist and only the additional weight of a UE is introduced. Although it would be possible to combine the messages for handover and dynamic clustering measurements in real implementations, specific CMR messages are introduced in this thesis for the sake of clarity.

Figure 4.3 shows the generation of CMR messages in a small network consisting of three UEs and three TPs. We assume in the example that the channel attenuation is directly proportional to the distance between TP and UE. The UEs move with speeds v_1, v_2 and v_3. Because the maximum allowed cluster size is assumed to be $S_{C,\max} = 3$, the CMR messages contain RSRP measurements for all three TPs. In the example the TPs in $m.\boldsymbol{\mathcal{B}}$ are ordered according to descending RSRP, even if this is not necessarily required.

4.2.3 Cluster Reconfiguration

To reconfigure the clustering the following tasks have to be performed by the CM. The first task during the cluster reconfiguration is the creation of new CEs for every cluster in the new clustering. This includes the creation of a new scheduler instance for each CE. The next step is to assign BBUs to the CEs. The actual number of BBUs required per CE depends on various parameters, like the cluster size or the load in the cluster. However, this task is out of scope of this thesis and we assume that always sufficient processing capacity is available. Then the CM assigns the RRHs serving the new cluster to the CE before in the final step the UE states are migrated from the old CE into the new one. This involves the transfer of queued data as well as context information. Additionally, the UEs have to be informed about this reconfiguration. To do so, similar mechanisms as already defined for the handover procedure within LTE networks can be used. Basically a RRC reconfiguration message and the information about the new cell, in this case the new cluster, has to be sent to the affected UEs.

For the instantiation and disposal of the software components several design options exist depending on the underlying technology. E.g., if the software components are realized as VMs or OS containers, new instances could be cloned from a base image and started. For the disposal the VMs or containers are simply turned off. Alternatively, a pool of software components could be created. For the instantiation one entity is taken from the pool and configured accordingly. A non-needed software component is reset and returned to the pool of idle components.

All reconfiguration steps require a certain execution time, which depends on the actual implementation. As we do not consider realization aspects in this thesis, we assume that the cluster reconfiguration can be performed in negligible time compared to the cluster reconfiguration interval T_R. During the cluster reconfiguration, the affected UEs cannot be served. However, data loss can be prevented, because incoming data for downlink transmission is buffered in the UE states until the new CE is created. Uplink data is buffered directly in the UE.

4.2.4 Integration of Dynamic Clustering into System Operation

Dynamic clustering could be performed independently of all other system operations. However, it makes sense to align it to the already existing timescales and intervals. While the dynamic clustering is performed by the CM only once per cluster reconfiguration interval T_R, the schedulers in the CEs determine the resource allocation also in regular intervals. In case of LTE the time step equals a TTI of 1 ms. It is not reasonable to configure T_R to smaller values as the TTI, because in these cases the schedulers cannot make any use of the new clustering. Thus, the value of T_R should be set to multiples of the TTI.

4.3 Dynamic Clustering Algorithm Design

In this section, we present the dynamic clustering algorithm that is executed by the CM. Starting in Section 4.3.1 with the design principles to achieve the stated goals, we introduce the parameters that control the dynamic clustering algorithm in Section 4.3.2. Section 4.3.3 presents the concept of cluster candidates. Finally, we show in Section 4.3.4 how the dynamic clustering algorithm generates a clustering from the available cluster candidates.

4.3.1 Design Principles

To achieve the goals stated in Section 4.1, the algorithm design is guided by the following principles. First, *CoMP should be facilitated for as many UEs as possible*, which means for the cluster definition that the cluster borders should be located where no or only a few UEs are affected. Also, the clusters should be defined such that as many UEs as possible are in the center of a cluster. Together with the goal to *avoid inter-cluster interference*, this leads to clusters that are as large as possible. In this way CoMP becomes feasible for more UEs and the size of the cluster borders in relation to the area of a cluster is reduced. On the other hand, the algorithm should *reduce the complexity of CoMP*, which is achieved by defining the clusters as small as necessary. Therefore, the trade-off between increased performance achieved by larger clusters and reduced overhead by smaller clusters has to be balanced by the algorithm. The dynamic clustering algorithm should be able to *avoid frequent cluster reconfigurations*, as this causes additional overhead and complexity. So the defined clusters should be beneficial for as long as possible, which is enabled by taking the mobility of UEs into account.

The algorithm design must additionally respect the following constraints. The system state is not completely known and only provided in terms of CMR messages. To allow the integration in the chosen system architecture, the defined clusters may not overlap and also it should be possible to tightly couple the cluster definition with scheduling to avoid contradicting optimization goals.

4.3.2 Algorithm Parameters

Two parameters control the definition of clusters. The maximum allowed cluster size is configurable as well as the time interval between cluster reconfigurations. The maximum cluster size is denoted with $S_{C,\max}$ and the cluster reconfiguration interval with T_R. Both parameters are introduced to limit the additional complexity caused by DCCSF. By limiting the cluster size the additional complexity of signal processing, i.e., precoding, is bounded. Similarly, restricting cluster reconfigurations to certain intervals reduces the overhead in the system. On the one hand this reduces the required resources on the radio interface, needed to perform channel measurements and transmit CMR messages. On the other hand, a cluster reconfiguration involves instantiation of the described software components and the transfer of UE states from one CE to another. During the transfer of the UE state the corresponding UE cannot be served. Limiting the frequency of reconfigurations therefore reduces their negative impact.

4.3.3 Concept of Cluster Candidates

Before the dynamic clustering algorithm can be executed, the input information has to be generated from the CMR messages. The input for the dynamic clustering algorithm are possible cluster candidates that are combined to a non-overlapping clustering in the next step. The cluster candidates are generated according to Algorithm 4.1. The idea of the algorithm is to generate cluster candidates for each UE that issued a CMR message. From each CMR message cluster candidates with sizes from one up to $S_{C,\max}$ are generated. The cluster candidates contain the best TPs as seen by the UE, where "best" is defined by the RSRP. A better TP has a higher RSRP. So the cluster candidate with size one only contains the TP with the highest RSRP, the cluster candidate with size two contains the

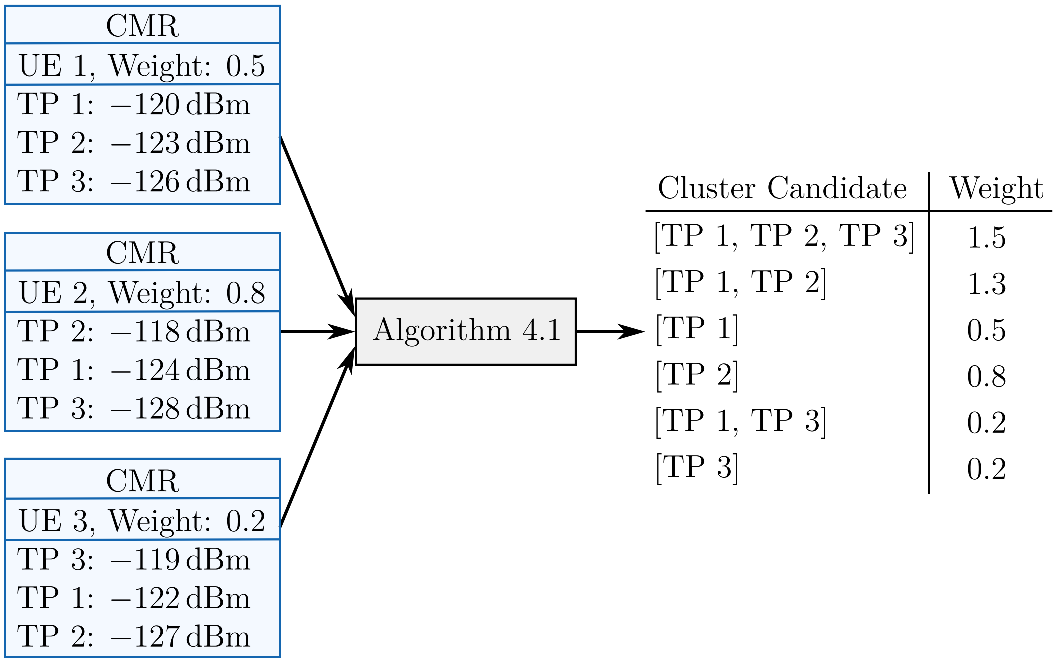
Algorithm 4.1 Generation of cluster candidates

```
 1: function GenerateClusterCandidates(CMR messages $\mathcal{M}$)
 2:     // Initialize
 3:     $\boldsymbol{\mathcal{C}} \leftarrow \{\}$                                   // Create new map used to store
                                                          all cluster candidates
 4:     $\boldsymbol{\mathcal{B}}_{\mathrm{UE}} \leftarrow \{\}$                     // Create new map to store the
                                                          best TP for each UE
 5:     // Process messages
 6:     for all received CMR message $m \in \mathcal{M}$ do
 7:         Sort $m.\boldsymbol{\mathcal{B}}$ according to descending RSRP     // TP with highest RSRP first
 8:         $\boldsymbol{\mathcal{B}}_{\mathrm{UE}}\{m.\mathrm{UE}\} \leftarrow m.\boldsymbol{\mathcal{B}}[0]$
 9:         for $S_C = S_{C,\max} - 1 \ldots 0$ do
10:             $C \leftarrow$ New empty cluster candidate        // Initialize cluster candidate. $C.\mathcal{B}$
                                                          and $C.\mathcal{U}$ are initialized as $\emptyset$
11:             $C.\mathcal{B} \leftarrow [m.\boldsymbol{\mathcal{B}}[0], \ldots, m.\boldsymbol{\mathcal{B}}[S_C]]$    // Add first $S_C$ TPs to new cluster
                                                          candidate
12:             $\boldsymbol{\mathcal{C}}\{C\} \leftarrow \boldsymbol{\mathcal{C}}\{C\} + m.w$
13:         end for
14:     end for
15:     return $\boldsymbol{\mathcal{C}}$, $\boldsymbol{\mathcal{B}}_{\mathrm{UE}}$
16: end function
```

TPs with the highest and second highest RSRP, and so on. To allow the algorithm to select a suitable clustering, a weight is assigned to each cluster candidate, which is derived from the CMR messages. In the case a cluster candidate is generated multiple times, which happens if the same TPs are reported by multiple UEs, they are combined by summing up the individual weights. As an extension, but not considered here, it would be also possible to take further aspects into account to determine the weight of the cluster candidate, like different priorities of the UEs. Another extension that is also not considered, would be to include not only CMR messages received during the current reconfiguration interval, but also include cached messages from previous intervals if an active UE has not transmitted a new CMR message.

The cluster candidate creation in Algorithm 4.1 needs the set of all received CMR messages $\mathcal{M}$ as input. The first steps are then to initialize the maps $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{B}}_{\mathrm{UE}}$ (Lines 3 and 4). The map $\boldsymbol{\mathcal{C}}$ contains for each cluster candidate (the key of an entry) the according weight (the value of an entry). The map $\boldsymbol{\mathcal{B}}_{\mathrm{UE}}$ stores the best TP, i.e., the TP with highest RSRP, for each UE. Thus, the keys of the map are UEs, while the values are TPs.

The actual creation of cluster candidates starts in Line 6 with an iteration over all received CMR messages. Then the map $m.\boldsymbol{\mathcal{B}}$ contained in the CMR message is sorted in descending order such that the TP with the highest RSRP comes first. The best TP is directly stored in $\boldsymbol{\mathcal{B}}_{\mathrm{UE}}$ (Line 8). The cluster candidates are then created in the loop in Lines 9 to 13. The size S_C of the newly created cluster candidates ranges from $S_{C,\max}$ down to one. This is possible because each received CMR message contains exactly $S_{C,\max}$ RSRP measurements and TPs. In the first iteration the new cluster candidate consists of all TPs contained in the CMR message, the second one consists of the first $S_{C,\max} - 1$ TPs, until the final



Cluster Candidate	Weight
[TP 1, TP 2, TP 3]	1.5
[TP 1, TP 2]	1.3
[TP 1]	0.5
[TP 2]	0.8
[TP 1, TP 3]	0.2
[TP 3]	0.2

Figure 4.4: Example for generation of cluster candidates

candidate contains only the best TP. The newly created candidates are inserted into the map $\boldsymbol{\mathcal{C}}$ and the weight of the UE is added to the value of the map entry (Line 12). If $\boldsymbol{\mathcal{C}}$ not already contains the cluster candidate C, we assume that $\boldsymbol{\mathcal{C}}\{C\}$ returns 0, such that the assignment in Line 12 also works if the candidate is created for the first time.

The principle of the algorithm is exemplary shown in Figure 4.4 for the scenario introduced in Figure 4.3. The table on the right indicates the map $\boldsymbol{\mathcal{C}}$, where the left column represents the keys of the map and the right column shows the values. Here and in all following examples we use $C.\mathcal{B}$ to identify clusters or cluster candidates. The algorithm creates one cluster candidate of size three, consisting of the TPs reported in the CMR messages. The weight of this candidate therefore is the sum of the weights reported in the CMR messages. Also, two cluster candidates of size two are created. The cluster candidate [TP 1, TP 2] is a result from the CMR messages issued by UE 1 and UE 2, which both report TP 1 and TP 2 as their best or second best TP. The cluster candidate [TP 1, TP 3] is generated from the CMR message of UE 3. Finally, three cluster candidates of size one are created, which consist of the best TP for each UE.

4.3.4 Generation of a Clustering

The main part of the dynamic clustering algorithm is described by Algorithm 4.2. The algorithm is executed in the CM once per cluster reconfiguration interval T_R. This is expressed by the infinite loop in Algorithm 4.2. The execution of the algorithm can be separated into three steps.

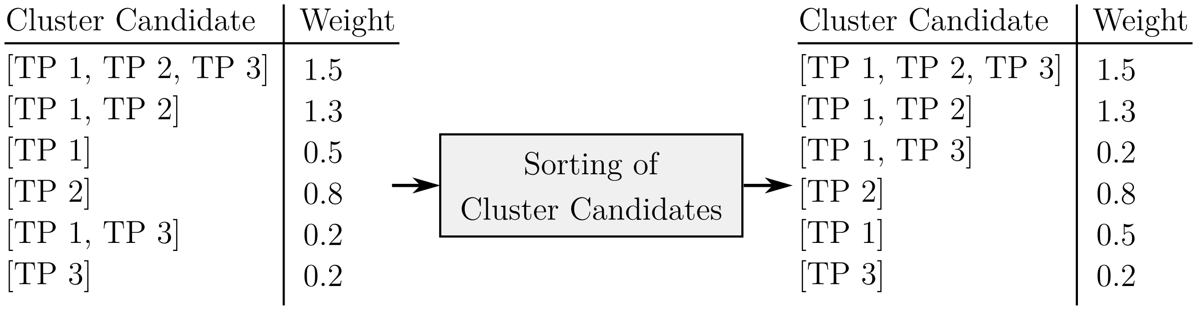
In the first step the maps $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{B}}_{\mathrm{UE}}$ are initialized using the previously defined Algorithm 4.1. Then the entries of the map $\boldsymbol{\mathcal{C}}$ are sorted according to their keys. For the

Algorithm 4.2 Dynamic clustering algorithm

```
while true do
    // Initialize
    $\boldsymbol{\mathcal{C}}$, $\boldsymbol{\mathcal{B}}_{\text{UE}} \leftarrow$ GenerateClusterCandidates($\mathcal{M}$)   // According to Algorithm 4.1
    Sort($\boldsymbol{\mathcal{C}}$, Compare())   // Sort map of cluster candidates using comparator function of Algorithm 4.3
    // Create clustering
    $\mathcal{B}_{\text{considered}} \leftarrow \emptyset$   // Set to store the TPs already added to the clustering
    $\mathcal{C} \leftarrow \emptyset$   // Initialize clustering
    for all Cluster Candidate $C \in \boldsymbol{\mathcal{C}}$ do
        if $C.\mathcal{B} \cap \mathcal{B}_{\text{considered}} = \emptyset$ then
            $\mathcal{C} \leftarrow \mathcal{C} \cup C$   // Add candidate to clustering
            $\mathcal{B}_{\text{considered}} \leftarrow \mathcal{B}_{\text{considered}} \cup C.\mathcal{B}$
        end if
    end for
    // Assign UEs to selected clusters
    for all Cluster $C \in \mathcal{C}$ do
        for all Map Entry (UE $u$, TP $b$) $\in \boldsymbol{\mathcal{B}}_{\text{UE}}$ do
            if $b \in C.\mathcal{B}$ then   // Add UE to the cluster, if it contains the TP with highest RSRP
                $C.\mathcal{U} \leftarrow C.\mathcal{U} \cup u$
            end if
        end for
    end for
    Configure new clustering $\mathcal{C}$ and wait for time $T_R$
end while
```

Algorithm 4.3 Comparator function for two cluster candidates

```
function Compare(Cluster Candidate $C_1$, Weight $w_{C_1}$, Cluster Candidate $C_2$, Weight $w_{C_2}$)
    if $|C_1| = |C_2|$ then   // Cluster candidates have same size
        if $w_{C_1} > w_{C_2}$ then
            return $C_1$
        else if $w_{C_1} > w_{C_2}$ then
            return $C_2$
        else
            return either $C_1$ or $C_2$ randomly   // Ordering is not defined
        end if
    else
        if $|C_1| > |C_2|$ then return $C_1$ else return $C_2$ end if
    end if
end function
```



Cluster Candidate	Weight
[TP 1, TP 2, TP 3]	1.5
[TP 1, TP 2]	1.3
[TP 1]	0.5
[TP 2]	0.8
[TP 1, TP 3]	0.2
[TP 3]	0.2

Cluster Candidate	Weight
[TP 1, TP 2, TP 3]	1.5
[TP 1, TP 2]	1.3
[TP 1, TP 3]	0.2
[TP 2]	0.8
[TP 1]	0.5
[TP 3]	0.2

Figure 4.5: Sorting of cluster candidates using comparator function from Algorithm 4.3

sorting any sorting algorithm can be used, as long as for the comparison of two cluster candidates the comparator function from Algorithm 4.3 is used. This function takes two cluster candidates and their corresponding weights and returns the cluster candidate that should come first. If the sizes of the cluster candidates differ, the larger cluster candidate is returned. In case both cluster candidates have the same size, the weight is used for the comparison and the cluster candidate with higher weight is returned. If both candidates have the same size and weight, the ordering is undefined and one of them is returned randomly. This ordering ensures that larger cluster candidates have a higher priority in the following step of determining the clustering. This fulfills the design goal of selecting the largest possible clusters to avoid inter-cluster interference. Also, sorting the cluster candidates according to their weight ensures that candidates that are beneficial for a longer duration have a higher priority.

The sorted cluster candidates are used in the second step of Algorithm 4.2 (Lines 6 to 13) to determine the clustering. To do so, first the set $\mathcal{B}_{\text{considered}}$ and the clustering $\mathcal{C}$ are initialized as empty sets. $\mathcal{B}_{\text{considered}}$ stores the TPs of all clusters that are already selected. During the selection of new clusters only those candidates are considered, whose TPs are not already contained in $\mathcal{B}_{\text{considered}}$. Thereby, a non-overlapping clustering is guaranteed. Then an iteration over the sorted cluster candidates is started. If the TPs of the currently selected cluster candidate are disjoint from the already considered TPs (Line 9), the cluster candidate is used. This means the cluster candidate is added to clustering $\mathcal{C}$ and the TPs of the cluster candidate are added to the set of already considered TPs $\mathcal{B}_{\text{considered}}$.

In the final step (Lines 15 to 22), the UEs are assigned to the selected clusters and the system is reconfigured to use the new clustering. Therefore, the algorithm iterates over the clustering $\mathcal{C}$. A UE is added to a selected cluster, if the TPs of the cluster contains the TP of the UE contained in $\boldsymbol{\mathcal{B}}_{\text{UE}}$. This means the UE is added to the cluster that contains the best TP for the UE. Finally, the new clustering is configured and the next cluster reconfiguration interval is awaited.

Again we use the example from Figure 4.3 to illustrate the dynamic clustering algorithm. Figure 4.5 shows the sorting of the cluster candidates. As can be seen, first the cluster candidates are sorted according to their size, such that larger candidates come first. Cluster candidates with same size are then sorted according to their weights, such that candidates with larger weights come first.

Iteration	Available Cluster Candidates	Selected Clusters
0	[TP 1, TP 2], [TP 1, TP 3], [TP 2], [TP 1], [TP 3]	-
1	[TP 1, TP 3], [TP 2], [TP 1], [TP 3]	[TP 1, TP 2]
2	[TP 2], [TP 1], [TP 3]	[TP 1, TP 2]
3	[TP 1], [TP 3]	[TP 1, TP 2]
4	[TP 3]	[TP 1, TP 2]
5	-	[TP 1, TP 2], [TP 3]

Table 4.1: Generation of the clustering

To show the generation of the clustering from the available cluster candidates, we assume that the maximum allowed cluster size is $S_{C,\max} = 2$ in the following example. This means in contrast to the previous examples that the UEs only report the RSRPs of two TPs in their CMR messages. As a consequence, the cluster candidate [TP 1, TP 2, TP 3] is not available. The iterations performed by Algorithm 4.2 are shown in Table 4.1. At the beginning, all cluster candidates are still available and no cluster is selected. In the first step the first cluster is selected. In the second iteration, the cluster candidate [TP 1, TP 3] is considered, but cannot be selected, because TP 1 is contained in the already selected cluster [TP 1, TP 2]. So the cluster candidate is skipped without any further action. The same also happens in iterations three and four. Only in the last step, it is possible to select the cluster [TP 3]. The result is the clustering, i.e., the set of selected clusters, which contains all TPs. We can see that by generating cluster candidates with size one, it is always possible to consider all TPs with active UEs in a clustering. Consequently, each active UE can be served by its TP with highest RSRP.

4.4 Scheduler Design

As discussed in Section 2.4, the effort for the scheduler to decide which UEs are served on which resources by which TPs is high. This effort is reduced if clusters are introduced, because fewer scheduling options have to be considered, but it is still significant. However, not all the scheduling options are useful to increase the system performance. So focusing only on beneficial options further reduces the effort of the scheduling process. DCCSF supports this by introducing the concept of partitionings. Together with a greedy scheduling approach, only the most relevant scheduling options have to be considered.

4.4.1 Concept of Partitionings

To reduce the number of possible scheduling options, the concept of partitionings is introduced. Each of the selected clusters in the clustering are further divided into multiple partitionings, which are exclusively used by the scheduler. After the CM has determined the clustering, it generates the partitionings and passes them to the newly instantiated schedulers. This means that partitionings are also generated in regular time intervals of T_R and the schedulers use them for the duration of one cluster reconfiguration interval. The partitionings are generated similarly as the clustering by using the cluster candidates,

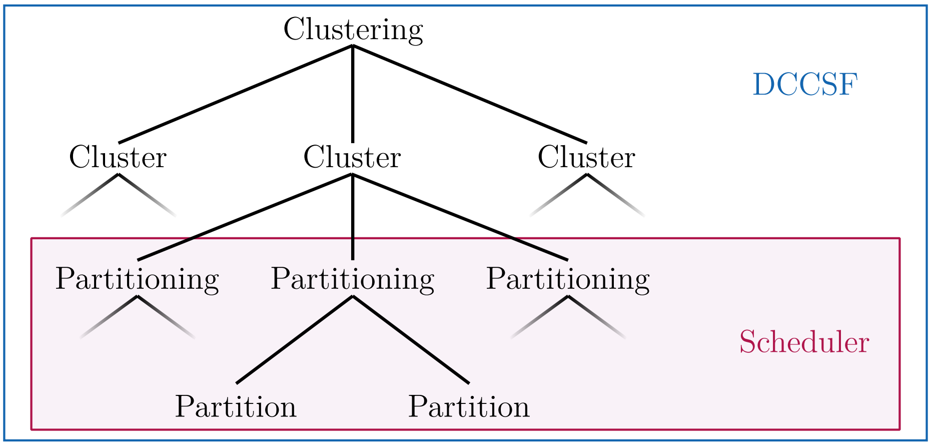


Figure 4.6: Relation of clustering, clusters, partitionings and partitions

created from the CMR messages. The cluster candidates act as prototypes for partitions. Analogous to the clustering, which consists of multiple clusters, a partitioning consists of multiple partitions.[5] Figure 4.6 illustrates the relation between clustering, cluster, partitioning and partition. It also indicates that all are created by DCCSF, but only partitionings and partitions are used by the schedulers. The set of partitionings of a cluster is denoted with $\mathcal{S}_{\mathcal{P}}$ and is stored in the attribute $C.\mathcal{S}_{\mathcal{P}}$ of the cluster. A partitioning is denoted as $\mathcal{P}$ and the symbol P is used for a partition. If a partitioning $\mathcal{P}$ belongs to a cluster C, then C is called the parent-cluster of $\mathcal{P}$. A partition has the attributes $P.\mathcal{B}$ to store its TPs and $P.\mathcal{U}$ to store its UEs.

Using the partitionings in the scheduling process reduces the scheduling effort, because the partitions directly determine which TPs cooperatively apply JT. All TPs belonging to the same partition always serve the scheduled UEs of the partition using JT. This reduces the problem of deciding which UEs are served by which TPs to the problem of which UEs are served by which partition. Because the number of UEs that can be scheduled within a partition is upper bounded by the total number of available transmit antennas in the partition and for all TPs JT is applied, the greedy scheduling approach only considers $\min\left(|P.\mathcal{B}| \cdot N_{\text{TX}}, |P.\mathcal{U}|\right)$ scheduling options in partition P. We assume here single antenna UEs, such that the total number of transmitted MIMO streams is upper bounded by the number of UEs. The greedy scheduler has to decide which UEs are scheduled and how many MIMO streams are used.

The principle of dividing a cluster in partitions is shown using the example from Figure 4.3. In contrast to the previous example in Section 4.3.4, we assume again $S_{C,\max} = 3$ and that the cluster including all three TPs and UEs has been configured during the dynamic clustering process. In this example the cluster candidates [TP 1, TP 2, TP 3], [TP 1, TP 2], [TP 1, TP 3] and the single TP candidates [TP 1], [TP 2] and [TP 3] have been generated. We use the cluster candidates to determine partitionings that contain all TPs and UEs of the parent-cluster, i.e., in the example the cluster consisting of all three TPs. By

[5] Note that in this thesis the term partition is used differently than in combinatorics. In combinatorics a partition is the grouping of a set of discrete elements into subsets. We use the terms partitioning and partition similar as it is used to denote the division of a computer hard drive.

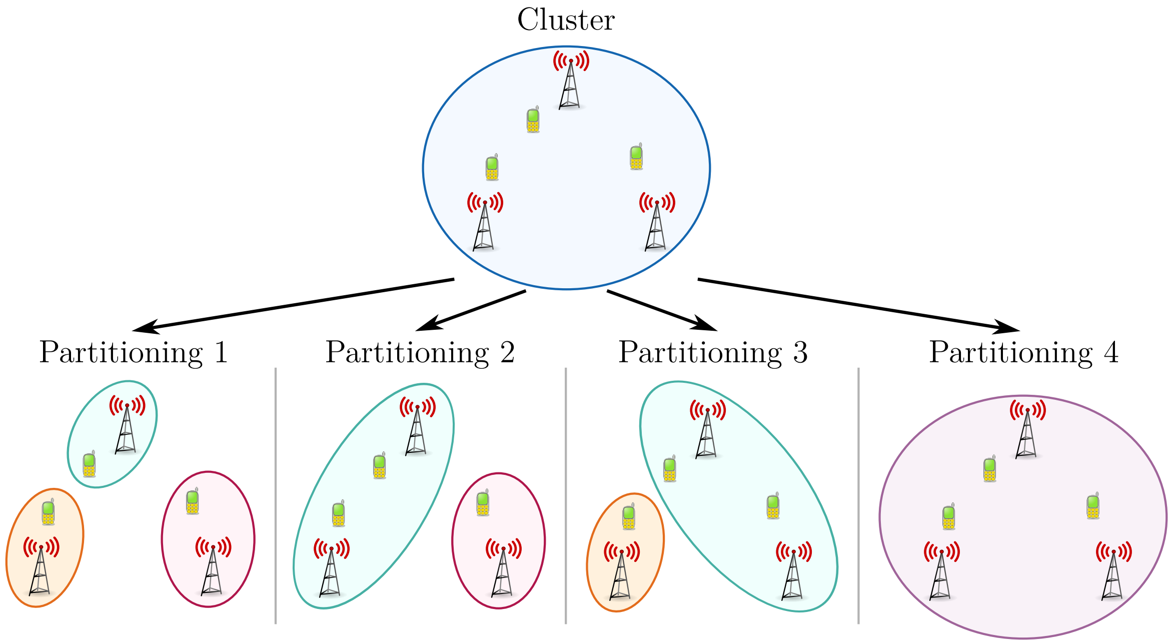


Figure 4.7: Example of generated partitionings

combining the cluster candidates such that all TPs of the parent-cluster are contained in each partitioning, the following partitionings are created:

- $\mathcal{P}_1 = [[\text{TP 1}], [\text{TP 2}], [\text{TP 3}]]$
- $\mathcal{P}_2 = [[\text{TP 1, TP 2}], [\text{TP 3}]]$
- $\mathcal{P}_3 = [[\text{TP 1, TP 3}], [\text{TP 2}]]$
- $\mathcal{P}_4 = [[\text{TP 1, TP 2, TP 3}]]$

Partitioning $\mathcal{P}_1$ contains three partitions, where each partition consists of a single TP. Here it becomes visible that the generated cluster candidates act as prototypes for the partitions. E.g., the cluster candidates with a single TP are the prototypes for the partitions in partitioning $\mathcal{P}_1$. For every partitioning $\mathcal{P}_i$, the UEs are assigned to the partition that contains the TP with the highest RSRP for this UE. The set of partitionings for the cluster is then defined as $\mathcal{S}_\mathcal{P} = [\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4]$. Figure 4.7 depicts the generated partitionings and the association of UEs to partitions. As we can observe, the cluster candidate [TP 2, TP 3] has not been generated, so it is not included in the partitionings, too. Assuming single transmit antennas at the TPs, the number of scheduling options for this cluster would be 40 (refer to Section 2.4 and Appendix A.2). Using the greedy scheduling approach together with the partitionings, the total number of considered scheduling options can be reduced to twelve, which reduces the effort by 70 %. For partitioning 1 as depicted in Figure 4.7, the total number of scheduling options is three, because for each of the three partitions one scheduling option is available. For partitionings 2 and 3, the number of scheduling options is also three, because for the partitions with two TPs two scheduling options have to be considered and for the partitions with one TP one option is considered. Finally, for partitioning 4 three options have to be considered. With this example it becomes evident that using the partitionings during the scheduling reduces the effort. For scenarios with more UEs the effort reduction can be even larger.

Algorithm 4.4 Create partitionings for scheduling

```
 1: function CreatePartitionings(Clustering $\mathcal{C}$, Cluster Candidates $\boldsymbol{\mathcal{C}}$, $\boldsymbol{\mathcal{B}}_{\text{UE}}$)
 2:     for all Cluster $C \in \mathcal{C}$ do
 3:         $\mathcal{C}_C \leftarrow \emptyset$                  // Set of partition candidates for cluster $C$
 4:         for all Cluster Candidate $C_o \in \boldsymbol{\mathcal{C}}$ do
 5:             if $C_o.\mathcal{B} \subseteq C.\mathcal{B}$ then
 6:                 $\mathcal{C}_C \leftarrow \mathcal{C}_C \cup C_o$      // If all TPs of $C_o$ are included in cluster $C$,
                                               cluster candidate $C_o$ is added
 7:             end if
 8:         end for
 9:         $\mathcal{S}_{\mathcal{P}} \leftarrow \emptyset$                  // Set that contains the partitionings
10:         CreateRecursive($C$, $\mathcal{C}_C$, $\emptyset$, $\mathcal{S}_{\mathcal{P}}$)
11:         // Add UEs to newly created partitions
12:         for all Partitioning $\mathcal{P} \in \mathcal{S}_{\mathcal{P}}$ do
13:             for all Partition $P \in \mathcal{P}$ do
14:                 for all Map Entry (UE $u$, TP $b$) $\in \boldsymbol{\mathcal{B}}_{\text{UE}}$ do
15:                     if $b \in P.\mathcal{B}$ then
16:                         $P.\mathcal{U} \leftarrow P.\mathcal{U} \cup u$
17:                     end if
18:                 end for
19:             end for
20:         end for
21:         $C.\mathcal{S}_{\mathcal{P}} \leftarrow \mathcal{S}_{\mathcal{P}}$              // Add partitionings to the cluster
22:     end for
23: end function
```

4.4.2 Generation of Partitionings

The creation of partitionings is performed by Algorithm 4.4. The algorithm needs the clustering $\mathcal{C}$, the map of cluster candidates $\boldsymbol{\mathcal{C}}$ and the map containing the best TP per UE $\boldsymbol{\mathcal{B}}_{\text{UE}}$, all as created in Algorithm 4.2. First, an iteration over all configured clusters is started to generate the partitionings for them. Therefore, a new set is created and filled with cluster candidates that are partitions of the currently considered cluster (Lines 3 to 8). Before the recursive creation of partitionings is started, the set of partitionings $\mathcal{S}_{\mathcal{P}}$ is initialized as an empty set. This set contains the defined partitionings after the termination of the recursive algorithm. The recursive creation of the partitionings is carried out in the function CreateRecursive in Line 10. The last step is then to add UEs to the found partitions. In order to do so two iterations over the found partitionings and the partitions are performed (Lines 12 to 20).

The actual creation of the partitionings is performed by the function shown in Algorithm 4.5, which requires four parameters: the parent-cluster C, the set of partition candidates of the parent-cluster $\mathcal{C}_C$, the helper set $\mathcal{P}$ to store the currently created partitioning and pass it to the next iterations of the algorithm, and finally the set $\mathcal{S}_{\mathcal{P}}$ in which the found partitionings are stored. The first step of the algorithm is to check, if the passed partitioning in the set $\mathcal{P}$ is complete. A partitioning is defined to be complete, if all the TPs of its parent-cluster are contained in the partitioning. The check is carried out with the function shown in

Algorithm 4.5 Recursive creation of partitionings

```
function CreateRecursive(Cluster $C$, Partition Candidates $\mathcal{C}_C$, Partitioning $\mathcal{P}$, Partitionings $\mathcal{S}_\mathcal{P}$)
    if Complete($C$, $\mathcal{P}$) then
        $\mathcal{S}_\mathcal{P} \leftarrow \mathcal{S}_\mathcal{P} \cup \mathcal{P}$    // Add partitioning $\mathcal{P}$ to the set of partitionings for cluster $C$
    else
        for all Partition Candidate $P \in \mathcal{C}_C$ do
            if not Overlaps($P$, $\mathcal{P}$) then
                $\mathcal{C}_{C,r} \leftarrow \mathcal{C}_C \setminus P$    // Create a new set of remaining partition candidates as a copy of $\mathcal{C}_C$ and remove the partition $P$
                $\mathcal{P}' \leftarrow \mathcal{P} \cup P$    // Create a new partitioning as a copy of $\mathcal{P}$ and add partition $P$
                CreateRecursive($C$, $\mathcal{C}_{C,r}$, $\mathcal{P}'$, $\mathcal{S}_\mathcal{P}$)    // Continue with next recursion
            end if
        end for
    end if
end function
```

Algorithm 4.6 Check if a partitioning is complete

```
function Complete(Cluster $C$, Partitioning $\mathcal{P}$)
    for all TP $b \in C.\mathcal{B}$ do
        $contained \leftarrow$ false
        for all Partition $P \in \mathcal{P}$ do
            if $b \in P.\mathcal{B}$ then
                $contained \leftarrow$ true
            end if
        end for
        if $contained =$ false then
            return false
        end if
    end for
    return true
end function
```

Algorithm 4.7 Check if a partition overlaps with the other partitions of a partitioning

```
function Overlaps(Partition $P$, Partitioning $\mathcal{P}$)
    for all Partition $P_o \in \mathcal{P}$ do
        if $P.\mathcal{B} \cap P_o.\mathcal{B} \neq \emptyset$ then
            return true
        end if
    end for
    return false
end function
```

$$\boldsymbol{\mathcal{A}}_C = \{[\text{TP }1, \text{TP }2] : \boldsymbol{\mathcal{A}}_{P_1}, [\text{TP }3, \text{TP }4] : \boldsymbol{\mathcal{A}}_{P_2}\}$$

r	$\mathcal{U}_{\text{scheduled}}$
1	[UE 1, UE 2]
2	[UE 2]
3	[UE 3, UE 4]

r	$\mathcal{U}_{\text{scheduled}}$
1	[UE 5]
2	[UE 5, UE 6]
3	[UE 5, UE 7]

Figure 4.8: Illustration of resource allocation using partitionings

Algorithm 4.6. If the partitioning is complete, it is added to the set of partitionings $\mathcal{S}_{\mathcal{P}}$ and the recursion terminates. In the case of an incomplete partitioning, the next recursion step is started for all partition candidates contained in the set $\mathcal{C}_C$ that are not overlapping with the partitions in the set $\mathcal{P}$. The check if partitions are overlapping is carried out by the function shown in Algorithm 4.7. Before the function CreateRecursive is called recursively, first a copy of the cluster candidates is created and the selected partition is removed from this new set (Line 7). Also, a copy of the set $\mathcal{P}$ is created and appended with the selected partition, which is then passed to the next recursion (Line 8).

Algorithm 4.6 shows the check if a partitioning is complete. The function needs the configured parent-cluster C and the partitioning $\mathcal{P}$ as input parameters. Then an iteration over the TPs of the parent-cluster is started and it is checked if all TPs are contained in one of the partitions contained in $\mathcal{P}$. Only if this is the case, the function returns **true**, otherwise **false** is returned.

The last helper function checks if a partition is overlapping with the other partitions of a partitioning. The function is presented in Algorithm 4.7. It needs the partition P to check and the partitioning $\mathcal{P}$ as input parameters. Then it checks if all partitions of the partitioning are disjoint from P, i.e., the intersection of the sets of TPs is an empty set. If this is not the case, the partition is at least partly overlapping and the function returns **true**, otherwise **false** is returned. The function has to return **true** even if the partition is only partly overlapping, i.e., only some TPs of P are contained in the partitioning, because otherwise the recursive algorithm would define overlapping partitionings.

4.4.3 Using Partitionings in the Scheduling Process

In the following, we demonstrate, how the found partitionings are used in the scheduling process. The goal of scheduling is to define the resource allocation, i.e., which UEs are served on which RBs by which TPs. As discussed, using partitions this problem can be simplified to decide which UEs are served on which RBs by which partition. Nevertheless, this has to be performed each scheduling interval for all defined clusters. The outcome of the scheduling process for a single cluster is described by the resource allocation $\boldsymbol{\mathcal{A}}_C$. Figure 4.8 illustrates the resource allocation. $\boldsymbol{\mathcal{A}}_C$ is a map, where the keys are partitions and the entries are again maps, called $\boldsymbol{\mathcal{A}}_P$. The map $\boldsymbol{\mathcal{A}}_P$ indicates which UEs are served by partition P on which RBs. Therefore, the keys are RBs and the entries are sets of scheduled UEs. In the figure, the maps $\boldsymbol{\mathcal{A}}_P$ are shown as tables on the bottom.

Algorithm 4.8 Using partitionings in the scheduling process

```
 1: function SCHEDULE(Cluster $C$)
 2:     $s_C \leftarrow 0$                       // Stores the best scheduling metric for cluster $C$
 3:     $\boldsymbol{\mathcal{A}}_C \leftarrow \{\}$                      // Stores the best resource allocation for cluster $C$
 4:     // Find best partitioning
 5:     for all Partitioning $\mathcal{P} \in C.\mathcal{S}_{\mathcal{P}}$ do
 6:         $s_{\mathcal{P}} \leftarrow 0$                   // Stores the scheduling metric for partitioning $\mathcal{P}$
 7:         $\boldsymbol{\mathcal{A}}_{\mathcal{P}} \leftarrow \{\}$                  // Stores the resource allocation for partitioning $\mathcal{P}$
 8:         for all Partition $P \in \mathcal{P}$ do
 9:             $s_P, \boldsymbol{\mathcal{A}}_P \leftarrow$ DETERMINERESOURCEALLOCATION($P$)
10:             $s_{\mathcal{P}} \leftarrow s_{\mathcal{P}} + s_P$
11:             $\boldsymbol{\mathcal{A}}_{\mathcal{P}}\{P\} \leftarrow \boldsymbol{\mathcal{A}}_P$
12:         end for
13:         if $s_{\mathcal{P}} > s_C$ then
14:             $s_C \leftarrow s_{\mathcal{P}}$
15:             $\boldsymbol{\mathcal{A}}_C \leftarrow \boldsymbol{\mathcal{A}}_{\mathcal{P}}$
16:         end if
17:     end for
18:     return $\boldsymbol{\mathcal{A}}_C$                 // Return the best found resource allocation
19: end function
```

Algorithm 4.8 creates and determines the resource allocation $\boldsymbol{\mathcal{A}}_C$. Thereby, we make use of the abstract scheduling function DETERMINERESOURCEALLOCATION. This function implements the actual scheduling algorithm and determines the resource allocation for a given partition P. For the scheduling algorithm, well-known variants like Round Robin (RR) or Proportional Fair (PF) can be used. Also, new scheduling algorithms are thinkable, but out of scope in this thesis. When introducing the simulation model in Section 5.1.7, we show how the PF scheduling principle can be used. The function DETERMINERESOURCEALLOCATION determines a resource allocation taking the maximum possible number of MIMO streams into account and returns the performance metric s_P and the resource allocation $\boldsymbol{\mathcal{A}}_P$ for partition P. Using the performance metric two different resource allocations can be compared. A better resource allocation thereby results in a higher performance metric. However, the definition of what better means depends on the applied scheduling algorithm. E.g., in case of a PF scheduler, the performance metric is the sum of the logarithms of the UE rates (see Section 3.2.1.1).

The idea of Algorithm 4.8 is to determine the performance metric and resource allocation for all partitionings of a parent-cluster and then return the resource allocation with the highest performance metric. In order to do so, first the performance metric s_C and resource allocation $\boldsymbol{\mathcal{A}}_C$ are initialized as 0 and empty map, respectively (Lines 2 and 3). Then an iteration over all partitionings is started. Before a second iteration over the partitions of the selected partitioning is started, the variables $s_{\mathcal{P}}$ and $\boldsymbol{\mathcal{A}}_{\mathcal{P}}$ are initialized (Lines 6 and 7). They are used to store the performance metric and resource allocation of partitioning $\mathcal{P}$. In the second iteration, the resource allocation is determined for the selected partition, the performance metric is added to the total performance metric for the partitioning and the resource allocations are combined (Lines 9 to 11). If the total performance metric of the considered partitioning is higher than the already found performance metric, the performance metric and resource allocation of the partitioning are stored (Lines 13 to 16). Finally, the resource allocation with the highest performance metric is returned by the function (Line 18). This resource allocation is then used to assign RBs to the UEs and to transmit data towards the UEs.

4.5 Handling of Special Cases

During normal system operation, it can happen that the concept of dynamic clustering, as presented in the previous sections, has to be extended to be able to handle special cases. In the following, we show these extensions at the example of two special cases. This also demonstrates that DCCSF seamlessly integrates into existing systems, such that it allows future system extensions.

4.5.1 Entering of new UEs

If a new UE wants to enter an LTE network, it has to find a cell first and perform basic synchronization. Then it has to receive and process basic cell information to communicate with the network. Only afterwards it is able to register properly in the system. In a network supporting DCCSF, the basic steps are exactly the same. The UE has to search for the cell with highest RSRP and perform the registration. Because this can happen at any time, the newly arriving UE is added to the already existing cluster that contains the TP with the highest RSRP for this UE. In the central office the corresponding software components have to created and added to the CE. Therefore, the new UE can directly make use of CoMP. However, its CMR message is only taken into account at the next cluster reconfiguration interval. If the TP with highest RSRP is not already contained in any cluster, i.e., it was not serving any UEs when the new UE enters the system, the UE is served without CoMP by this single TP. In this case CoMP is only possible after the next cluster reconfiguration interval when the UE has issued a CMR message.

4.5.2 Interworking with Handovers

Due to the mobility of UEs the necessity of handovers can arise at any time and is not limited to the cluster reconfiguration intervals. Thus, the system has to support independent handovers. This imposes no problem, because as soon as a handover measurement or CMR message is received from a UE and the requirement of a handover is detected, the system can perform this handover. Instead of performing a handover from a single cell into another single cell, the handover is now performed from one cluster into another. In the central office, this means that one UE is migrated from the old CE to the new CE. In case that the TP with highest RSRP for the UE is not contained in a cluster, because it was not serving any UEs, a handover from a cluster into a single TP is performed. Only at the next cluster reconfiguration interval, the system may decide to reconfigure the clustering.

4.6 Discussion

Concluding this chapter, we classify DCCSF according to the scheme developed in Section 2.3.4.1. Further, we analyze the theoretical worst-case and real-world complexities of DCCSF. We also show common properties and differences to other dynamic clustering algorithms.

4.6.1 Classification

According to the classification scheme developed in Section 2.3.4.1, DCCSF is characterized as follows. It makes use of a CM, so the clustering is determined *centrally*. The definition of the clustering is performed by an algorithm executing a *heuristic*. Additionally, the heuristic is *based on cluster candidates* that are generated from measurement reports from the UEs. As each TP belongs to exactly one cluster, the configured clusters are *non-overlapping*. Finally, cluster candidates are further used in the resource allocation process, which means that scheduling is directly *integrated* in the dynamic clustering process.

4.6.2 Complexity Analysis

In the following, we provide a brief discussion of the complexity of DCCSF. The main purpose of the derivation is to show how the maximum cluster size and the size of the scenario, i.e., the number of UEs and TPs, influence the complexity. The theoretical complexities of the dynamic clustering algorithm and the generation of the partitionings are separately analyzed in Section 4.6.2.1. We also show how the theoretical worst-case complexity differs from real-world complexity in Section 4.6.2.2. During the evaluation of DCCSF we analyze the complexity in Section 6.1.5.

4.6.2.1 Worst-case Complexity

In the following we separately analyze the worst-case complexities of the dynamic clustering algorithm and the generation of the partitionings. Thereby we use unrealistic but still possible assumptions.

Dynamic Clustering Algorithm

The worst-case for the number of cluster candidates is obtained if all UEs would report different TPs in their CMR messages, such that the total number of cluster candidates becomes:

$$|\boldsymbol{\mathcal{C}}| = N_{\mathrm{UE}} \cdot S_{C,\max} \tag{4.2}$$

However, this means that more TPs than UEs are available in the system, which is usually not the case. In the course of generating a clustering, the cluster candidates have to be sorted. Using an optimal sorting algorithm the worst-case complexity is $\mathcal{O}(n \cdot \log(n))$ [Knu98]. Therefore, the complexity of sorting the cluster candidates is:

$$\mathcal{O}\left(|\boldsymbol{\mathcal{C}}| \cdot \log\left(|\boldsymbol{\mathcal{C}}|\right)\right) = \mathcal{O}\left(N_{\mathrm{UE}} \cdot S_{C,\max} \cdot \log\left(N_{\mathrm{UE}} \cdot S_{C,\max}\right)\right) \tag{4.3}$$

The actual creation of the clustering is performed by iterating over all cluster candidates, such that the complexity scales linearly with the number of cluster candidates. In the worst-case for each UE $S_{C,\max}$ cluster candidates are generated. Therefore the complexity of the selection of the clustering is:

$$\mathcal{O}(|\boldsymbol{\mathcal{C}}|) = \mathcal{O}\left(N_{\mathrm{UE}} \cdot S_{C,\max}\right) \tag{4.4}$$

The final step of assigning UEs to clusters scales linearly with the number of clusters, i.e., $|\mathscr{C}|$, and also linearly with the number of UEs. The worst-case occurs if an individual cluster is selected for each UE. In this case the complexity is:

$$\mathcal{O}\left(|\mathscr{C}| \cdot N_{\mathrm{UE}}\right) = \mathcal{O}\left(N_{\mathrm{UE}}^2\right) \tag{4.5}$$

The overall complexity of the dynamic clustering algorithm is therefore dominated by the assignment of UEs to clusters in the worst-case as given by Equation (4.5). The complexity scales quadratically with the number of UEs.

Generation of Partitionings

Partitionings are generated for every configured cluster. The worst-case for the number of clusters is obtained if for each UE an individual cluster is generated. The complexity of generating partitionings is maximized if the generated clusters all have a size of $S_{C,\max}$. For each of the selected clusters, the partitionings have to be created. Thereby, the cluster candidates for a cluster have to be generated first. Under the assumption that all cluster candidates have been created, the number of cluster candidates per cluster is as follows. However, this overestimates the number of cluster candidates per cluster, because under the conditions given above, only a single UE is contained in the cluster, such that not all cluster candidates can be proposed.

$$|\mathscr{C}_C| = \sum_{i=1}^{S_{C,\max}} \binom{S_{C,\max}}{i} = 2^{S_{C,\max}} - 1 \tag{4.6}$$

The recursive algorithm that defines the partitionings has the worst-case complexity of:

$$\mathcal{O}(|\mathscr{C}_C|!) \tag{4.7}$$

Because the recursive algorithm has to be executed for all selected clusters, the total complexity of the creation of the partitionings is:

$$\mathcal{O}\left(|\mathscr{C}| \cdot |\mathscr{C}_C|!\right) = \mathcal{O}\left(N_{\mathrm{UE}}\left(2^{S_{C,\max}} - 1\right)!\right) \tag{4.8}$$

The helper functions to check if a partitioning is complete and to check if a partition is already included in a partitioning scale linearly with the size of the parent-cluster and the number of partitions in the partitioning, such that they are not dominating. The overall complexity thus scales linearly with the number of UEs in the system and by $\left(2^{S_{C,\max}} - 1\right)!$ with the maximum cluster size.

4.6.2.2 Real-World Complexity

In the following, we reason why the complexity in real-world scenarios is usually much smaller than the above stated theoretical worst-case complexity. Especially the influence of the maximum cluster size on the complexities of the dynamic clustering algorithm and the generation of the partitionings is in real scenarios much smaller. The reason is that typical values of $S_{C,\max}$ are relatively small. In Chapter 6, we show that values of $S_{C,\max} = 3$ are sufficient to achieve significant performance improvements and that larger clusters lead to relatively small additional gains. Also, the maximum cluster size is independent of the scenario size, which means that also for scenarios with many TPs and UEs the cluster size is limited to relatively small values.

Dynamic Clustering Algorithm

For the derivation of the worst-case complexity of the dynamic clustering algorithm we assumed that all UEs reports different TPs in their CMR messages and that for each UE an individual cluster is selected. This condition requires that the number of TPs is at least $S_{C,\max}$ times larger than the number of UEs ($N_{\mathrm{TP}} \geq S_{C,\max} \cdot N_{\mathrm{UE}}$). However, in real systems the number of UEs is usually much larger than the number of TPs ($N_{\mathrm{UE}} \gg N_{\mathrm{TP}}$). Additionally, multiple UEs usually report the RSRP for the same TPs in their CMR messages, which reduces the total number of cluster candidates. We stated that the complexity of the clustering algorithm is mainly influenced by the sorting of cluster candidates (Equation (4.3)), the selection of the clusters (Equation (4.4)) and the assignment of UEs to clusters (Equation (4.5)). Using the more realistic estimation of the number of clusters $|\mathcal{C}| = \left\lceil \frac{N_{\mathrm{TP}}}{S_{C,\max}} \right\rceil$, the complexities for the three parts of the dynamic clustering algorithm can be estimated as follows.

Under the condition that the number of UEs is much larger than the cluster size, the following approximation[6] holds:

$$N_{\mathrm{UE}} \cdot S_{C,\max} \cdot \log\left(N_{\mathrm{UE}} \cdot S_{C,\max}\right) \underset{N_{\mathrm{UE}} \gg S_{C,\max}}{\approx} N_{\mathrm{UE}} \cdot \log\left(N_{\mathrm{UE}}\right) \tag{4.9}$$

Then the complexity of sorting can be approximated as (compare with Equation (4.3)):

$$\mathcal{O}\left(N_{\mathrm{UE}} \cdot \log\left(N_{\mathrm{UE}}\right)\right) \tag{4.10}$$

A similar reasoning can be applied to approximate the number of required iterations for the cluster selection:

$$N_{\mathrm{UE}} \cdot S_{C,\max} \underset{N_{\mathrm{UE}} \gg S_{C,\max}}{\approx} N_{\mathrm{UE}} \tag{4.11}$$

Then the approximated complexity of cluster selection becomes (compare with Equation (4.4)):

$$\mathcal{O}\left(N_{\mathrm{UE}}\right) \tag{4.12}$$

[6] This approximation is not valid in the strict mathematical sense. However, it is correct in the sense that the complexity of sorting mainly depends on the number of UEs while the influence of the maximum cluster size is comparatively low. The same principle also applies for all following approximations used to derive the real-world complexities.

For the approximation of the assignment of UEs to clusters, we use the reasonable assumption that the number of UEs is much larger than the number of TPs. Additionally, we assume that the number of TP is also much larger than the cluster size. Then the following approximation holds:

$$|\mathcal{C}| \cdot N_{\mathrm{UE}} = \left\lceil \frac{N_{\mathrm{TP}}}{S_{C,\max}} \right\rceil \cdot N_{\mathrm{UE}} \underset{N_{\mathrm{UE}} \gg N_{\mathrm{TP}}/S_{C,\max}}{\approx} N_{\mathrm{UE}} \tag{4.13}$$

Then the complexity of assigning the UEs to clusters can be approximated as (compare with Equation (4.5)):

$$\mathcal{O}\left(N_{\mathrm{UE}}\right) \tag{4.14}$$

We can conclude that the real-world complexity of the dynamic clustering algorithm is smaller than the worst-case complexity. Also, the complexity is mainly influenced by the number of UEs, as the approximations in Equations (4.10), (4.12) and (4.14) reveal.

Generation of Partitionings

For the derivation of the worst-case complexity of the generation of the partitionings we assumed that for each UE an individual cluster is selected. Using the same more realistic assumption for the number of selected clusters as for the real-world complexity of the dynamic clustering algorithm, the number of clusters is $|\mathcal{C}| = \left\lceil \frac{N_{\mathrm{TP}}}{S_{C,\max}} \right\rceil$. Thereby, the number of clusters is independent of the number of UEs. With the realistic assumption that the number of TPs is much larger than the maximum cluster size, the following approximation holds:

$$|\mathcal{C}| \cdot |\mathcal{C}_C|! = \left\lceil \frac{N_{\mathrm{TP}}}{S_{C,\max}} \right\rceil \cdot \left(2^{S_{C,\max}} - 1\right)! \underset{N_{\mathrm{TP}} \gg S_{C,\max}}{\approx} N_{\mathrm{TP}} \tag{4.15}$$

Therefore, the complexity of the creation of the partitionings becomes:

$$\mathcal{O}\left(N_{\mathrm{TP}}\right) \tag{4.16}$$

This estimation is further supported by the fact that the complexity of the recursive algorithm to determine the partitionings is in many cases lower than the factorial scaling. The reason is that it terminates as soon as a complete partitioning is found. E.g., if in the first round the partition candidate that is equal to the parent-cluster is selected, the recursion terminates immediately.

Discussion

We conclude that the real-world complexity of DCCSF is not a hindrance of its introduction in real systems. The complexity of the dynamic clustering algorithm scales with $N_{\mathrm{UE}} \cdot \log\left(N_{\mathrm{UE}}\right)$ with the number of UEs. The complexity of generation of the partitionings scales linearly with the number of TPs in the system.

	Algorithm I by Baracca et al.	Algorithm II by Baracca et al.	Algorithm by Weber et al.	DCCSF
Generated cluster candidates per UE	1	$S_{C,\max}$	≥ 1, depending on applied filtering	$S_{C,\max}$
Using all TPs	no	yes	(yes)	yes
Objective of cluster selection	Weighted Sum-Rate Maximization (WSRM)		utility function / special sorting of cluster candidates	
Interval between cluster reconfigurations	same as for scheduling (TTI)		arbitrary	
Integration of precoding	tight, because necessary for WSRM		no	
Integration of scheduling	tight, because necessary for WSRM		no	independent of scheduling algorithm, but supports schedulers to reduce effort

Table 4.2: Comparison of different dynamic clustering algorithms

4.6.3 Comparison with other Dynamic Clustering Algorithms

DCCSF shares common properties with other dynamic clustering algorithms presented in the literature. In this section, we discuss the similarities and highlight the differences. We only include a selected subset of available dynamic clustering algorithms, which also work with the concept of cluster candidates that are then combined in a Central Unit (CU) to a non-overlapping clustering. In DCCSF the Cluster Manager (CM) fulfills the purpose of the CU. These algorithms have been introduced in Section 2.3.4.3. Table 4.2 shows an overview of the properties of the dynamic clustering algorithms. Properties shared by DCCSF and other clustering algorithms are highlighted in blue.

The first property we compare is the number of cluster candidates generated from the measurements reported by the UEs. Algorithm I by Baracca et al. generates only a single candidate per UE. However, the size of the cluster candidate is variable as the TPs included in the measurement report are either defined by an absolute or a relative configurable threshold of the average measured SNR. In algorithm II by Baracca et al. this concept is extended such that a maximum cluster size is configurable. For each UE cluster candidates of size one up to the maximum cluster size are generated. In the algorithm by Weber et al. the UEs report all TPs matching certain RSRP thresholds and cluster candidates of sizes up to the number of TPs included in the measurement reports are generated. However, before the actual clustering is determined, the cluster candidates are filtered, such that only candidates of configurable sizes are considered. By setting the minimum size equal to one and the maximum size equal to $S_{C,\max}$, the same results as with algorithm II by Baracca et al. can be achieved. DCCSF follows the same principle of cluster candidate creation as algorithm II by Baracca et al.

The next compared property is, if all TPs with active UEs contained in the cluster candidates are used in the clustering. For algorithm I by Baracca et al. this is not guaranteed as the CU only has limited cluster candidates, which cannot always be combined such that all TPs are used in a clustering. The same could happen in the algorithm of Weber et al. if the cluster size thresholds for the candidate filtering are set inappropriately. Because in algorithm II by Baracca et al. and DCCSF cluster candidates of sizes from one up to $S_{C,\max}$ are available, it is always possible to make use of all TPs.

The algorithms differ in terms of the objective they try to achieve with the cluster selection. Both algorithms by Baracca et al. aim to maximize the system's weighted sum rate (Weighted Sum-Rate Maximization (WSRM)). Therefore, a set of clusters is generated from the cluster candidates from which a non-overlapping clustering is selected for each scheduling interval. In contrast, the algorithm by Weber et al. and DCCSF order the cluster candidates and then greedily select as many clusters from the ordered list until a non-overlapping clustering consisting of all TPs is generated. However, both algorithms differ in the way the cluster candidates are sorted. Weber et al. use a cost function (refer to Equation (2.9)), whereas in DCCSF the candidates are directly sorted according to multiple criteria (see Algorithm 4.3). Nevertheless, it would be possible to design a cost function that results in the same ordering as the proposed sorting of the candidates. Thus, the principal approach is similar, even though the sorting of cluster candidates is performed differently.

Both algorithms from Baracca et al. select a clustering per scheduling interval. However, this does not mean that measurements have to be reported by the UEs every scheduling interval, but it is sufficient to report them and generate cluster candidates on a larger timescale. In contrast, the algorithm from Weber et al. and DCCSF work independently of the scheduling interval, such that arbitrary cluster reconfiguration intervals are possible.

The consequence of the WSRM approach of the algorithms from Baracca et al. is that both are tightly integrated with scheduling and precoding. Nevertheless, both algorithms allow using arbitrary scheduling or precoding schemes. In contrast, the algorithm by Weber et al. does not interact with precoding and scheduling at all and any scheduling or precoding scheme can be used. The same holds for DCCSF, with the difference that the schedulers are provided with additional information, to reduce the scheduling effort.

We conclude that DCCSF shares most of the properties with the algorithm of Weber et al. The main difference lies in the objective during the selection of the clustering and that DCCSF provides the schedulers with additional information, generated during the clustering process, to reduce the scheduling effort.

5 Simulation Model and Performance Measures

This chapter is organized into three parts and presents the foundation of the performance evaluation in Chapter 6. First, it describes in Section 5.1 the complete simulation model in which DCCSF is implemented. The used metrics by which the performance of the system is assessed are introduced in Section 5.2. Finally, Section 5.3 explains the evaluation methodology.

5.1 Simulation Model

Although the approach of combining dynamic clustering and scheduling for CoMP is not limited to LTE networks, an LTE system serves as the basis for the following evaluations. Reasons are the well-known system architecture and the availability of simulation guidelines in [3GPP 25.814] and [3GPP 36.814].

The model is implemented in a system-level simulation and represents an urban scenario with vehicular traffic. The simulation is based on the IKR SimLib and IKR RadioLib [SimLib]. These libraries provide basic functionalities for discrete event simulation, like random number generation, simulation control and statistical evaluation of simulation results. Also, higher layer building blocks to model LTE networks are provided. In the following, we describe the exact implementation of the simulation model.

5.1.1 Cellular Network

We use a 3D environment to model the geometry of the scenario. However, the scenario is flat, i.e., it does not contain any elevations like mountains or valleys and the z-dimension only fulfills the purpose to model the height of TPs and UEs. The scenario consists of a homogeneous layout of 19 hexagonal sites depicted in Figure 5.1. The distance between the sites is $D_{is} = 500\,\text{m}$. This results in a scenario size from east to west of approximately 2300 m and from north to south of approximately 2500 m. Each site serves three sectors of a horizontal width of 120°, which are treated as individual TPs. The height over ground of the TPs is set to $b_h = 32\,\text{m}$ and the height of the UEs is $u_h = 1.5\,\text{m}$. The resulting scenario is reasonable large to cover an urban environment. Also, the total number of 57 TPs is sufficiently high to provide enough degrees of freedom for the dynamic clustering algorithm to configure the clusters. All TPs are equipped with the same number of N_{TX} transmit antennas. All antennas have their own power amplifier with the transmit power of $P_{\text{TX}} = 46\,\text{dBm}$.

To avoid border effects, the wrap-around principle is used. This means a user leaving the scenario on the right side, will enter the scenario on the left side again. The same also

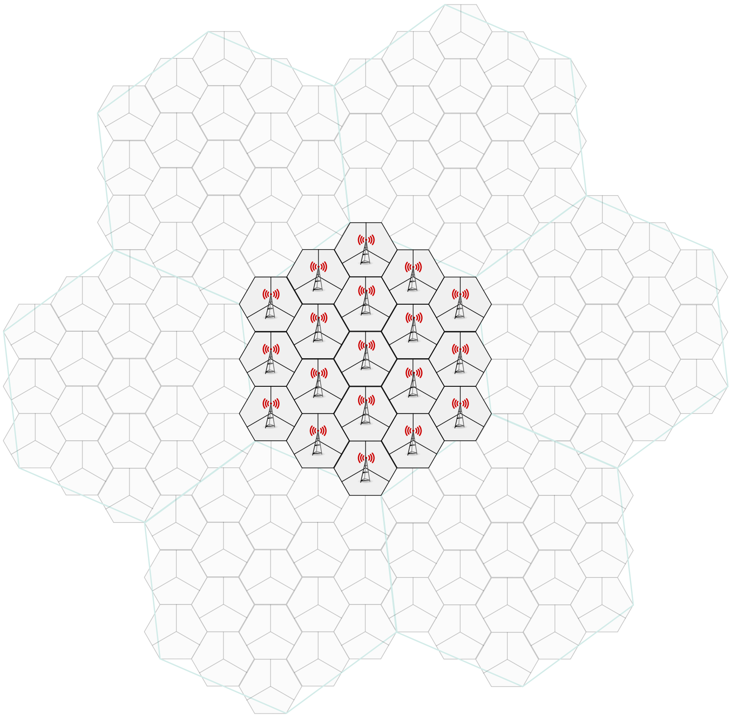


Figure 5.1: Illustration of the wrap-around scenario

holds for wireless radio signals. This is achieved by using seven clones of the scenario and arranging them as depicted in Figure 5.1.

The network uses a total bandwidth of 1.4 MHz at a carrier frequency of $f_c = 2\,\text{GHz}$, which results in six Resource Blocks (RBs) in the frequency domain. The choice of the small bandwidth is justified, because the goal of the model is to show relative performance improvements in comparison to systems without CoMP, while the absolute achievable performance is not as relevant. This is also possible with only six available RBs. This choice also reduces the simulation complexity, such that it is possible to simulate longer durations, which is necessary if the influence of user mobility is evaluated.

5.1.2 Mobility

The task of the mobility model is to describe the location and movement of users in the scenario. For the evaluations the following three models have been implemented. If the user is moving, its new location is calculated every millisecond based on its speed and heading.

5.1.2.1 Static Users Model

For reference purposes, we use a mobility model with static users. The users are initially placed randomly in the scenario, resulting in a uniform user distribution. As soon as a user finishes the transmission of a traffic object (explained in Section 5.1.3), the user is relocated to a new randomly chosen location, where the transmission of the next traffic object takes place. During data transmission the users are stationary however. The relocation of users

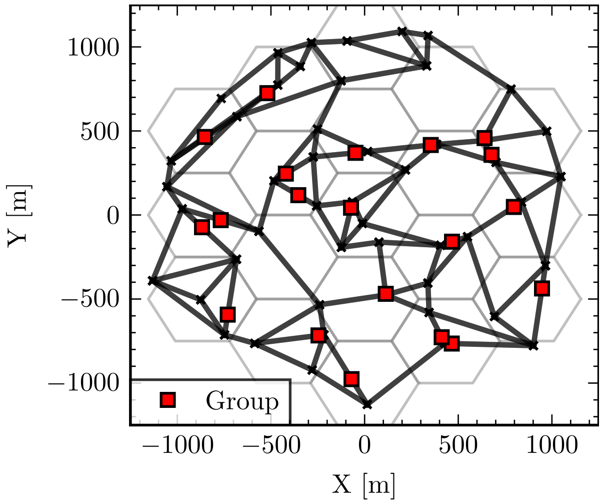


Figure 5.2: Example of a virtual track topology

Property	Average Value
Street Length	284.98 ± 127.36m
Streets per Junction	3.86 ± 0.42
Number of Streets	96.40 ± 1.50

Table 5.1: Average properties of virtual track topologies

ensures that sufficiently many locations are considered during the simulation. Additionally, the changing user locations lead to short-term fluctuations of the user density, such that the effects of dynamic clustering can be shown.

5.1.2.2 Static Hotspot Model

A second model without mobility is the Static Hotspot Model. This model comprises a configurable number of user hotspots. The hotspots are placed randomly on the scenario and the users are assigned to one of the hotspots randomly. The users are not directly positioned at the center of the hotspot, but they are placed uniformly randomly around the center with a maximum distance of 20 m. The positions of users and hotspots are fixed for the whole simulation duration.

5.1.2.3 Virtual Track Model

The Virtual Track mobility model was introduced by Zhou, Xu, and Gerla [ZXG04] and serves to model different user behaviors. E.g., the authors highlight the application for user mobility on highways but also the movement of soldiers in combat. We briefly describe the properties of the model and show how it is used to model the mobility of vehicular users in an urban scenario.

The basis of the model is an undirected graph. The graph vertices are junctions and the edges are tracks on which users move. The users move along the tracks in groups. If a group reaches a junction, the group chooses a new track on which the movement continues.

We use the Virtual Track model to model the mobility of vehicular users in an urban environment. Therefore, the tracks represent streets. The street topology is generated randomly, by placing the junctions randomly on the field. For both coordinates (X and Y) a uniform distribution is used. A minimum distance of 100 m between junctions is ensured. Streets are generated between junctions, if the number of streets per junction does not exceed a threshold. The maximum number of streets per junction is set to four. Also, intersection of streets is prohibited. A speed limit is assigned to each generated street. At simulation start a configurable number of groups is generated and the groups are initially placed on a random position on a randomly selected street. Also, the users are randomly assigned to the available groups. A group is defined by its center, which moves with the speed according to the speed limit of the street where it is located. The users of a group are placed on the street with a random distance from the group center. The maximum distance is thereby ±15 m. In the evaluations in most cases 570 users are distributed to 30 groups, which results in an average group size of 19 users. This leads to an average distance of approximately 1.6 m between the users of a group, which might seem dense. However, in this model a user not necessarily represents a single human, but could instead represent a vehicle with multiple passengers or the vehicle itself, which also transmits data. The speed limit is set in most cases to 50 km/h as we want to model vehicles moving in urban environments. As discussed in Section 3.1.3.1 this speed is often observed within cities.

Figure 5.2 shows an example street topology of 50 junctions. Wrap-around is disabled to simplify the illustration. Because in the evaluations wrap-around is also applied for streets, it is ensured that all streets and junctions are within the coverage area of a site, which is not the case in the shown example. In this scenario 20 groups are deployed, which are indicated by colored dots in the figure. Using the parameters from above results in street topologies with the average properties presented in Table 5.1. The averages are obtained from 10 different topologies generated with the given parameters. The table also includes the standard deviations. Comparing these values with real city topologies, it can be observed that the Virtual Track topologies have longer streets but a lower street density. E.g., Barrachina et al. [Bar+13] report an average street length of approximately 102 m for the city centers for eleven cities on different continents. Mohajeri, Gudmundsson, and French [MGF15] analyzed 41 cities in Great Britain and came up with an average street length of approximately 60 m to 90 m. They report street densities from 100 to 300 streets per km^2, while the Virtual Track topology has an average density of approximately 24 streets per km^2. From the perspective of the following evaluations, these differences are not crucial, as long as the general properties of vehicle movement in urban environments hold, i.e., movement in groups on a limited number of possible streets.

5.1.3 Data Traffic

The purpose of the data traffic model is to generate the traffic transmitted in the simulated network. This includes the instance of time when a new traffic object is generated and the size of the object. This general approach allows different levels of abstraction. E.g., the traffic objects could be individual IP or TCP packets or higher level objects like complete websites or downloads. For the evaluation of wireless networks only few generally accepted models exist. Examples are the models for HTTP and File Transfer Protocol (FTP) traffic specified in [3GPP 36.814], which model the traffic as application layer objects.

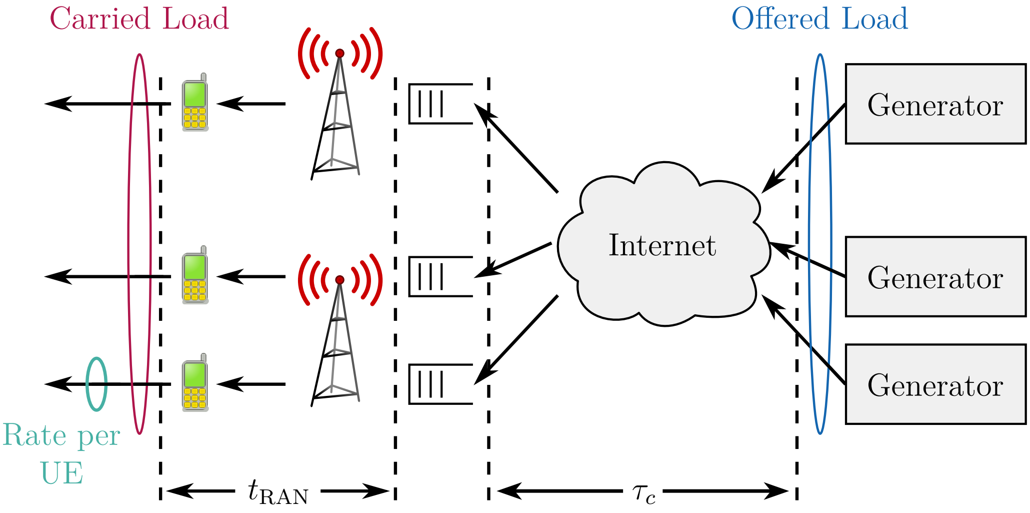


Figure 5.3: Traffic generation and rates in the simulation model

However, the origins of the models reach back to the late 20th century and therefore do not completely represent the characteristics of today's traffic.

Although the share of streaming and real-time entertainment traffic is ever-increasing and caused approximately 38 % of the downlink traffic in cellular networks in 2016 [San16], web browsing traffic still creates the major fraction. We therefore concentrate on web browsing traffic in the following evaluations. Another reason why we only model web browsing traffic is that the sizes of traffic objects of web browsing traffic are small in comparison to streaming media or real-time communications traffic. Therefore, web browsing traffic is more bursty and shows higher dynamics. So, if DCCSF increases the system performance for web browsing traffic, it would be even more beneficial for streaming media or real-time communications.

Figure 5.3 shows the overall architecture for the traffic generation. For each UE an own traffic generator, i.e., server in the Internet, is used. The traffic is modeled on the object level, such that an object represents for example a complete website or file download. The effects of the fronthaul and backhaul network and Internet are considered by a constant propagation delay of $\tau_c = 20\,\mathrm{ms}$ in the downlink direction. Because the bottleneck in cellular networks is the air interface, we assume that all other connections have an infinite bandwidth, which results in no transmission delays. The depicted queues are First In, First Out (FIFO) queues with unlimited capacity and store complete traffic objects. These queues are located in the central office in the real system (compare Figure 4.1) and are connected via the fronthaul network to the RRHs. As we already include the fronthaul delay in the constant propagation delay τ_c, the queues in Figure 5.3 are depicted directly in front of the radio interface to simplify the illustration.

We model the traffic on an object level and thereby abstract from the effects introduced by the IP and TCP layers. The model is based on measurement results from Hernández-Campos et al. [Her+04]. The authors performed application layer object size measurements in a campus network in the early-2000s. The basic finding of the measurements is that the sizes follow a superposition of three log-normal distributions. This is in line with more recent studies, which indicate that the object size distributions are heavy-tailed [MSF10;

Fal+10; IP11]. The reason, why we use relatively old measurements, is that the available data rates in cellular networks and thus also the carried traffic volume is smaller than in fixed networks. The traffic characteristics of these old measurements therefore resemble the characteristics of today's web traffic in cellular networks.

In contrast to the original measurements, we clip the object size distribution at 100 MB This means if the object size drawn from the distribution exceeds 100 MB, new samples are drawn until the sample is below the threshold. We chose this approach to avoid problems with large objects, because it could happen that these are not completely transmitted within the simulated time. Not completely transmitted objects occupy radio resources, but cannot be considered in the evaluation. After clipping the distribution, the average traffic object size is approximately 6.25 kB.

The load is controlled by adjusting the Inter-Arrival Time (IAT) of new traffic objects. The IAT is modeled by a negative-exponential random variable with adjustable mean value. A stable system operation is guaranteed for all IATs, because in case the offered traffic rate exceeds the system capacity, the objects are buffered in the queues as can be seen in Figure 5.3.

For the evaluations it is not necessary to represent the actual behavior of today's applications, but it is sufficient to cover the overall characteristics. There are two main arguments for this reasoning. First, we do not know how the applications will behave when DCCSF will be introduced in real networks. So, it is not necessary to model the behavior of today's applications in great detail, as their characteristics are volatile and might change with any new software version, e.g., application, web browser or OS on UE and server. Second, due to the high diversity of applications generating web browsing traffic, e.g., web browsing, mailing or automatic software updates, it is not possible to capture the characteristics of all applications in a single model. Also, we do not want to model a specific set of applications, but evaluate how DCCSF performs in general.

5.1.4 Channel

In this section, we describe all sources of attenuation and gains between TP and UE. In this thesis, we use models suitable for system-level simulations, which mainly comply to the guidelines from [3GPP 36.814] for urban macro scenarios[1].

The goal of the channel model is to describe the total attenuation $h_{u,b,r}$ between TP b and UE u on RB r, which is composed of path loss, penetration loss, shadowing, fast fading and antenna characteristics. While the path loss, penetration loss, shadowing and the antenna characteristics are constant over the whole channel bandwidth, the fast fading is frequency selective, i.e., also dependent on the considered RB r. The total attenuation is then obtained by summing up all components.

[1] Macro refers to the type of BSs. In the original reference this scenario is also referred to as "Case 1" or "Model 1".

Path Loss

The path loss component captures the effect of signal attenuation due to distance between transmitter and receiver. Path loss does not take any obstacles like buildings or other natural objects into account. The path loss $h_{u,b,\mathrm{PL}}$ is according to [3GPP 36.814] defined as:

$$h_{u,b,\mathrm{PL}} = (128.1 + 37.6 \cdot \log_{10}(d_{u,b}))\,\mathrm{dB} \tag{5.1}$$

With $d_{u,b}$ being the distance between UE u and TP b given in kilometer.

[3GPP 36.814] assumes an additional penetration loss of $h_{u,b,\mathrm{P}} = 20\,\mathrm{dB}$, which takes the transmission through all obstacles between TP and UE into account.

Shadowing

The shadowing component $h_{u,b,\mathrm{S}}$ is modeled according to 3GPP recommendations from [3GPP 36.814] as a log-normal distributed random variable with a standard deviation of 8 dB. The mean value of the log-normal distribution is set to 0 dB. The values of the shadowing components of two TPs from different sites are correlated with a correlation coefficient of 0.5. The correlation of the shadowing component between different locations decreases exponentially with the distance between the two locations. At a distance of 50 m the correlation coefficient decreases to 0.5. According to 3GPP recommendations, the shadowing component is the same for all sectors of a site, because the paths from all TP of the site to the UE are the same.

Fast Fading

Fast fading is caused by multipath propagation and is modeled by Rayleigh fading. This fading model is applicable, if there is no dominant path, i.e., no Line of Sight (LoS) condition, between transmitter and receiver, which is typically the case in urban environments [Stü01, chapter 1.2.1]. For the actual implementation of Rayleigh fading, we use an improved version of the Jakes model [DBC93]. As the multipath propagation is also determined by the Doppler shift, the speed of the UEs and the carrier frequency are needed as inputs for the model. Fast fading is calculated for each RB individually, but is assumed to be constant within a RB.

Transmit and Receive Antennas

For the transmit antennas at the TPs a 3D model is used, such that the antenna gain depends on the vertical and horizontal angles between TP and UE. Figure 5.4 illustrates the involved angles. In this example the main lobe of the TP antenna points in the x-direction and downward by θ_{tilt}. The total antenna gain $h_{u,b,\mathrm{A,TX}}$ is defined according to [3GPP 36.814]. However, we inverted and simplified the original equations, such that the

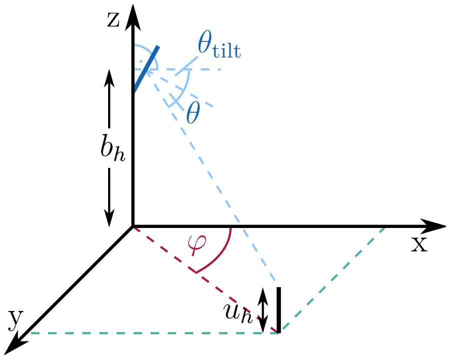


Figure 5.4: Illustration of the antenna gain

antenna gain becomes an equivalent attenuation $h_{u,b,\mathrm{A,TX}}$, which can directly be added to the other sources of channel attenuation.

$$h_{u,b,\mathrm{A,TX}} = -14\,\mathrm{dB} + \min\left(A_H(\varphi) + A_V(\theta), A_m\right) \tag{5.2}$$

With $A_m = 25\,\mathrm{dB}$ and $A_H(\varphi)$ and $A_V(\theta)$ being the horizontal and vertical components, respectively. The BS antenna gain of 14 dB is taken from [3GPP 25.814]. The angles φ and θ are the horizontal and vertical angles between TP and UE antennas. The horizontal component is defined as:

$$A_H(\varphi) = \min\left(12\left(\frac{\varphi}{\varphi_{3\,\mathrm{dB}}}\right)^2, A_m\right) \tag{5.3}$$

The angle $\varphi_{3\,\mathrm{dB}}$ represents the angle where the antenna gain is reduced by 3 dB and is set to $\varphi_{3\,\mathrm{dB}} = 70°$. The vertical component is defined as:

$$A_V(\theta) = \min\left(12\left(\frac{\theta - \theta_{\text{tilt}}}{\theta_{3\,\mathrm{dB}}}\right)^2, A_v\right) \tag{5.4}$$

With $A_v = 20\,\mathrm{dB}$, $\theta_{3\,\mathrm{dB}} = 10°$ representing the angle reducing the antenna gain by 3 dB and θ_{tilt} being the down-tilt of the antenna. The angle θ_{tilt} is not specified in [3GPP 36.814], but for calibration purposes values of 6° and 15° are proposed. In the simulation model we use the value of $\theta_{\text{tilt}} = 10°$. Smaller down-tilts lead to higher interference between sectors facing each other, but tend to improve the achievable CoMP gain, because the signal power at the border between facing sectors is higher.

The UEs have only a single isotropic receive antenna with a noise figure of 9 dB.

5.1.5 Link Abstraction

The link abstraction model is responsible to determine the performance of the wireless transmission. The tasks of the link abstraction model are to determine the capacity of a transmission (Section 5.1.5.1) for a given link quality (Section 5.1.5.2). Additionally, the overhead caused by LTE control signaling has to be modeled, to determine the channel capacity correctly (Section 5.1.5.3).

5.1.5.1 Capacity Abstraction

Instead of using the defined Modulation and Coding Schemes (MCS) in LTE (see Section 2.1.3) to determine the capacity of the channel, we use the Shannon-Hartley theorem [Sha49] as a more general approach. This is motivated by the fact that CoMP was not considered for the definition of the MCS. Also, it has to be noted that using the Shannon-Hartley theorem allows to determine the capacity after the transmission has been carried out, while using a real MCS requires to select the MCS before the transmission. Therefore, it can happen that the selection of the MCS was too optimistic, resulting in a decode failure at the UE. In LTE this is handled by the HARQ mechanism, which is not necessary in the approach with the Shannon-Hartley theorem. Consequently, the chosen approach models an optimal MCS selection.

The Shannon-Hartley theorem yields the amount of data $d_{u,r}$ transmitted to UE u in RB r:

$$d_{u,r} = B_r \cdot T_r \cdot \mathrm{ld}\,(1 + \min(\gamma_{u,r}, \gamma_{\max})) \tag{5.5}$$

B_r is the bandwidth of a RB (180 kHz in LTE) and T_r the duration of one RB (0.5 ms in LTE). The SINR $\gamma_{u,r}$ is bounded to $\gamma_{\max} = 24\,\mathrm{dB}$. With this value a spectral efficiency of approximately 8 bit/s/Hz is achieved, which equals the highest modulation scheme specified since LTE release 12 (256-QAM). This results in a maximal capacity of:

$$d_{u,r}^{\max} = 180\,\mathrm{kHz} \cdot 0.5\,\mathrm{ms} \cdot \mathrm{ld}\left(1 + 10^{24\,\mathrm{dB}/10}\right) \approx 718\,\mathrm{bit} \tag{5.6}$$

The maximum data rate per cell in case of a single MIMO layer transmission is:

$$R_{\mathrm{cell}}^{\max} = 6 \cdot d_{u,r}^{\max}/T_r = 8.616\,\mathrm{Mbit/s} \tag{5.7}$$

Because in total 57 TPs are available, the achievable maximum rate in the system is:

$$R_{\mathrm{system}}^{\max} = 57 \cdot R_{\mathrm{cell}}^{\max} = 491.112\,\mathrm{Mbit/s} \tag{5.8}$$

However, the actually usable rate is lower, because the calculation does not consider overhead caused by control signaling. Also, the cells cause interference to each other, such that the estimation is too optimistic.

5.1.5.2 CoMP Abstraction

We abstract from the transmission techniques required for CoMP, namely precoding and power allocation, and evaluate the resulting performance by considering transmit power and channel attenuation only. This is an often used approach in the literature [MF11, Chapter 7; Web+11; Mah13; AC14; HG17]. In the following, we discuss why this is a suitable abstraction for system-level simulations.

In Section 5.1.5.1, we have seen that the transmission performance, i.e., the data rate, is determined by the SINR. The CoMP abstraction therefore has to provide an accurate calculation of the SINR while being at the same time simple enough to allow large scale simulations. Additionally, the CoMP abstraction has to support CoMP clusters.

In such an abstracted scenario the SINR is determined by the actual signal power, the intra-cluster interference, the inter-cluster interference and the noise power at the receiver. From the discussion in Section 2.5.1 we can conclude that the presented precoding schemes are able to almost revert the effects of the channel within the cluster. Therefore, the intra-cluster interference is suppressed effectively and the useful signals interfere constructively at the receivers. While this is achieved optimally only by using nonlinear precoding, linear precoding techniques achieve inferior performance. However, both approaches are due to their computational complexity and the necessary detailed channel model not suitable for system-level simulations.

In the discussion of power allocation, which is related to precoding, in Section 2.5.2, we could see that optimal power allocation requires to solve an optimization problem. Equal Power Allocation (EPA) is a non-optimal power allocation scheme, but achieves comparable performance in high SNR regimes. Therefore, we use EPA to model power allocation.

From this we conclude that it is suitable to determine signal and interference power only by considering transmit power and channel attenuation and thereby assume idealistic transmissions in the sense that intra-cluster interference is suppressed completely. On the other hand, the chosen power allocation scheme is non-ideal. This is a reasonable choice, because the focus of the thesis is not to evaluate the performance of DCCSF if optimal transmission techniques are applied. Instead, the abstraction of the transmission techniques only has to cover the relevant effects, such that relative gains can be derived. Altogether this leads to the following calculations for signal and interference power. The signal power $P_{\text{signal},u,r}$ for UE u on RB r is modeled according to Equation (5.9):

$$P_{\text{signal},u,r} = \sum_{b \in P.\mathcal{B}} P_{\text{TX},r} \cdot h_{u,b,r} \cdot N_{\text{TX}} \cdot \frac{1}{N_{P,r}} \tag{5.9}$$

This equation models the signal power for UE u on RB r by summing up the transmit power $P_{\text{TX},r}$ for RB r of all TPs serving this UE, i.e., the TPs of the partition P ($P.\mathcal{B}$). This power is multiplied by the channel attenuation $h_{u,b,r}$ between UE and TP on RB r and the number of transmit antennas N_{TX} per TP. Transmit power and channel attenuation must be converted into the linear scale to allow the multiplication. The multiplication with N_{TX} covers the assumed individual power amplifier per transmit antenna, such that more transmit antennas at the TP lead to an increased total transmit power. EPA is modeled by the division by $N_{P,r}$, which is the number of served UEs on RB r within partition P. We also assume that the available transmit power is equally distributed to all RBs. Because six RBs are available in the frequency domain, the transmit power per RB is $P_{\text{TX},r} = {}^{P_{\text{TX}}}\!/\!_{6}$, with the total transmit power per antenna P_{TX}. All TPs have the same number of transmit antennas N_{TX} and each antenna has the same transmit power P_{TX}.

A similar approach is used in Equation (5.10) to calculate the inter-cluster interference power $P_{\text{inter-cluster interf.},u,r}$:

$$P_{\text{inter-cluster interf.},u,r} = \sum_{b \in \mathcal{B} \setminus P.\mathcal{B}} a_{b,r} \cdot P_{\text{TX},r} \cdot h_{u,b,r} \cdot N_{\text{TX}} \tag{5.10}$$

The interference received by UE u on RB r is modeled as the sum of the transmit powers $P_{\text{TX},r}$ of all TPs not serving UE u. If a TP is transmitting, is reflected by the binary variable $a_{b,r}$. It is set to $a_{b,r} = 1$ if TP b is transmitting on RB r and set to $a_{b,r} = 0$ if it is not transmitting. Again the transmit power is multiplied with the channel attenuation $h_{u,b,r}$ between UE u and TP b and the number of transmit antennas N_{TX} at TP b. Using this approach, it is assumed that UE u always receives full interference from all other transmitting TPs, which generally overestimates the interference. If precoding would be considered, the neighboring TPs would steer their transmissions such that most of the transmitted power can be received optimally by their served UEs, which leads to less interference.

This leads to the SINR $\gamma_{u,r}$ as defined in Equation (5.11):

$$\gamma_{u,r} = \frac{P_{\text{signal},u,r}}{P_{\text{intra-cluster interf.},u,r} + P_{\text{inter-cluster interf.},u,r} + \sigma^2} = \frac{P_{\text{signal},u,r}}{P_{\text{inter-cluster interf.},u,r} + \sigma^2} \tag{5.11}$$

Due to the discussion above we assume no intra-cluster interference ($P_{\text{intra-cluster interf.},u,r} = 0$) and the SINR is determined by the received signal power according to Equation (5.9), the inter-cluster interference power according to Equation (5.10) and the noise power σ^2. The noise power is set to $-174\,\mathrm{dBm/Hz}$.

5.1.5.3 LTE Overhead Abstraction

LTE uses a non-negligible fraction of the bandwidth for control and signaling purposes. How this is respected in the simulation model is described in the following. We assume to use the normal cyclic prefix length, so each RB consists of 84 Resource Elements (REs), i.e., twelve subcarriers times seven OFDM symbols. Thus, a TTI consists of 14 OFDM symbols leading to a total of 168 REs. Of these 14 OFDM symbols the first three are reserved for control signaling, e.g., to signal the downlink scheduling assignment in the PDSCH, uplink scheduling grants or HARQ acknowledgements. Although the size of the control region in real systems can be adjusted to match the control load and available bandwidth, we assume due to simplicity a fixed size here. Therefore, the number of REs used to transmit control information is $N_{\text{control}} = 3 \cdot 12\,\text{REs}$. This results in a constant overhead for two consecutive RBs of:

$$\frac{3 \cdot 12\,\text{REs}}{168\,\text{REs}} \approx 21.4\,\% \tag{5.12}$$

Additional REs are used for channel measurements. Depending on the used transmission mode, channel measurement in LTE is performed by Cell-Specific Reference Signals (CRS) and Channel State Information Reference Signals (CSI-RS) (see Section 2.6.2). The overhead for channel measurement within a cluster is determined by three parameters: the number of TPs within the cluster, i.e., the cluster size S_C, the number of transmit antennas located at the TPs (N_{TX}) and the periodicity of channel measurement (t_p). The total needed additional REs caused by transmission of reference signals N_{RS} is defined according to Equation (5.13).

$$N_{\mathrm{RS}} = \underbrace{\sum_{b \in C.\mathcal{B}} N_{\mathrm{CRS}}(N_{\mathrm{TX}})}_{\text{Overhead caused by CRS}} + \underbrace{N_{\text{CSI-RS}}\left(N_{\mathrm{TX}} \cdot S_C\right) \frac{1\,\mathrm{ms}}{t_p}}_{\text{Overhead caused by CSI-RS}} \tag{5.13}$$

As can be seen the total overhead is composed of the overhead for CRS and CSI-RS transmissions. The overhead of CRS depends on the number of transmit antennas per TP and must be calculated for each TP individually.[2] The number of needed REs can be found in Table 2.4. However, some of these REs are transmitted in the control region within the first three OFDM symbols of a TTI, such that they are already covered and may not be counted twice. The number of additional REs is given by Equation (5.14):

$$N_{\text{CRS}}(N_{\text{TX}}) = \begin{cases} 6 & N_{\text{TX}} = 1 \\ 12 & N_{\text{TX}} = 2 \\ 16 & N_{\text{TX}} = 4 \end{cases} \tag{5.14}$$

The number of needed CSI-RS is determined by the total number of transmit antennas in the cluster. Section 2.6.2.4 shows the calculation and Equation (5.15) gives the number of used REs. Because CSI-RS are not transmitted every TTI, we divide this number by the periodicity t_p, which leads to a non-integer number of needed REs. Nevertheless, this gives the average overhead observed on a longer timescale. We use the shortest possible periodicity of $t_p = 5\,\text{ms}$, as this results in the best performance at the cost of the worst-case overhead.

$$N_{\text{CSI-RS}}\left(N_{\text{TX}} \cdot S_C\right) = \begin{cases} 2 \cdot S_C & N_{\text{TX}} = 1 \\ N_{\text{TX}} \cdot S_C & \text{otherwise} \end{cases} \tag{5.15}$$

Combining the overhead of transmitting control signaling, CRS and CSI-RS leads to a reduction of the capacity as shown in the following equation, while the original capacity $d_{u,r}$ of RB r transmitted to UE u is determined by Equation (5.5):

$$d'_{u,r} = \frac{168\,\text{REs} - (N_{\text{control}} + N_{\text{RS}})}{168\,\text{REs}} d_{u,r} \tag{5.16}$$

5.1.6 Clustering

The dynamic clustering and generation of partitionings is implemented as described in Chapter 4. This includes also the exchange of Cluster Measurement Report (CMR) messages. The Reference Signal Received Power (RSRP) measurement is assumed to be ideal. The same holds for the velocity measurement of the UEs, which is ideal, too. In the simulation model CMR messages are transmitted from UE to Cluster Manager (CM) directly before a cluster reconfiguration is triggered by the reconfiguration interval. Thus, the cluster reconfiguration decision is based on recent RSRP and velocity measurements.

As reference, we also implement a system without any cooperation, called No CoMP in the following. Two static clustering schemes (Site Clustering (SC), see Figure 2.13a and Facing Sectors Clustering (FSC), see Figure 2.13b) are implemented to compare the performance between dynamic and static clustering. By definition the traditional system has a cluster size of one, and both static clustering schemes have a constant cluster size of three.

[2] Although it is possible to use the same REs to transmit CRS for multiple TPs, we assume the worst-case where different REs are used.

While for the dynamic clustering the partitionings are created based on the CMR messages, partitionings for both static variants are created statically. Thereby, all possible partitionings of a cluster are created. E.g., for the cluster consisting of TPs 1, 2 and 3 the following partitionings are created:

- [[TP 1], [TP 2], [TP 3]]
- [[TP 1, TP 2], [TP 3]]
- [[TP 1, TP 3], [TP 2]]
- [[TP 1], [TP 2, TP 3]]
- [[TP 1, TP 2, TP 3]]

Similar as in the case of dynamic clustering, UEs are assigned to the partition that contains the TP with highest RSRP for the UE. This gives the scheduler the maximum flexibility to select the best partitioning, but on the other hand increases the effort, because all possible, even probably non-meaningful, partitionings have to be considered.

5.1.7 Scheduler

In the evaluations, we combine the proposed scheduling concept for partitionings with the well-known Proportional Fair (PF) scheduling principle. Algorithm 4.8 performs the task of scheduling and the functionality of the function DetermineResourceAllocation is provided by a greedy PF scheduling algorithm shown in the following. Here it should be noted that the greedy algorithm does not guarantee to achieve optimal performance. This is not an issue for the evaluation results, because we are not interested in obtaining absolute performance values achievable with optimal schedulers. Instead, we want to evaluate the performance of DCCSF and compare it with the performance of systems with static clustering or systems without CoMP. Additionally, the proposed methods should be directly usable in real-world systems.

In the simulation model the schedulers rely on ideal channel knowledge, which is not available in real systems, where channel information is often outdated and only partially available for selected TPs. The ideal channel knowledge generally overestimates the scheduling performance. Nevertheless, this is a reasonable assumption, because the evaluations focus on the assessment of the performance gain achievable by CoMP.

To achieve a resource allocation according to the PF principle as introduced in Section 3.2.1, the scheduler assigns resources to the UE with the highest scheduling weight w_u. The scheduling weight is defined as the ratio between instantaneous rate r_u and long-term average rate $\overline{r}_u$:

$$w_u = \frac{r_u}{\overline{r}_u} \tag{5.17}$$

The long-term average rate is calculated as an exponentially moving average. The time index t thereby represents the scheduling interval, which is here the LTE TTI with a duration of 1 ms.

$$\overline{r}_u(t+1) = \beta r_u(t) + (1-\beta)\overline{r}_u(t) \tag{5.18}$$

For the forgetting factor β and the initial value of $\overline{r}_u$ we follow the recommendations in [3GPP2]. The forgetting factor is set to $\beta = {}^{1}/_{1500} \approx 6.667 \times 10^{-4}$. The initial value

Algorithm 5.1 Weight calculation

```
function CalcWeight(SINR $\gamma_{u,r}$)
    $r_u \leftarrow 180\,\mathrm{kHz} \cdot 0.5\,\mathrm{ms} \cdot \mathrm{ld}\,(1 + \min(\gamma_{u,r}, \gamma_{\max}))$  // Rate estimation based on SINR
    $w_u \leftarrow \frac{r_u}{\overline{r}_u}$  // According to Equation (5.17)
    return $w_u$
end function
```

of $\overline{r}_u$ is set to $\overline{r}_u = 1 \times 10^{-15}$. The calculation of the scheduling weight according to Equation (5.17) is performed by the function CalcWeight in Algorithm 5.1. The instantaneous rate is estimated using the estimated SINR, as presented in Section 5.1.5.1. The long-term average rate is updated after each scheduling decision. In case a UE was not scheduled in a TTI, the instantaneous rate is set to zero and the long-term average is updated accordingly.

The best resource allocation $\boldsymbol{\mathcal{A}}_P$ and the corresponding performance metric s_P of a partition are determined by the function DetermineResourceAllocation as shown in Algorithm 5.2. The first steps of this function are to initialize the performance metric s_P as zero and create the new resource allocation $\boldsymbol{\mathcal{A}}_P$ as an empty map. Also, the maximum number of MIMO layers is calculated (Lines 2 to 4). The maximum number of layers $N_{\mathrm{max},P}$ is determined by the minimum of the total number of transmit antennas and the number of single antenna UEs in the partition. Then for each RB the best UEs are determined in the loop from Lines 5 to 36. Within the loop two local variables are initialized to store the set of scheduled UEs on RB r ($\mathcal{U}_{\mathrm{scheduled}}$) and the performance metric for this RB ($s_{P,r}$). The following loop ranging from Lines 8 to 33 is used to determine the best number of MIMO layers. In each loop iteration a new UE is selected and added to the set of scheduled UEs, if this increases the total performance metric. Inside the loop two local variables are initialized to store the best found UE (u_{best}) and the corresponding scheduling weight (w_{best}). Then all not already selected UEs are evaluated in the loop from Lines 11 to 27. The intention of the loop is to estimate the scheduling weight for each UE and finally select the UE with the highest scheduling weight. Therefore, the signal and interference power is calculated using the transmit powers and channel attenuations (Lines 12 to 20).

For the calculation of the SINR in Line 21 EPA is assumed, which is achieved by dividing the signal power by the number of scheduled UEs. Based on the SINR the scheduling weight of the UE is calculated. Because the set of scheduled UEs $\mathcal{U}_{\mathrm{scheduled}}$ does not yet contain the currently considered UE, the signal power is divided by the number of UEs in $\mathcal{U}_{\mathrm{scheduled}}$ plus one. If the scheduling weight of the currently considered UE is higher than the stored value, both the old best scheduling weight and the best UE are replaced by the newly found values (Lines 23 to 26).

The new-found best UE is only added to the set of scheduled UEs if this increases the total performance of the partition. Therefore, the total performance is calculated with the help of the function CalcPerformance, which is described in Algorithm 5.3. The total performance is defined as the sum of the weights of the scheduled UEs. As can be seen this function performs similar steps as for the selection of the best UE. The difference is that adding new UEs to the set of scheduled UEs also influences the performance of the

Algorithm 5.2 Greedy PF scheduling algorithm

```
function DetermineResourceAllocation(Partition $P$)
    $s_P \leftarrow 0$                                                  // Initialize performance metric
    $\boldsymbol{\mathcal{A}}_P \leftarrow \{\}$                        // Initialize resource allocation
    $N_{\max,P} \leftarrow \min\left(|P.\mathcal{B}| \cdot N_{\mathrm{TX}}, |P.\mathcal{U}|\right)$   // Calculate maximum number of MIMO layers
    for all RB $r \in \mathcal{R}$ do                                   // Determine resource allocation for each RB
        $\mathcal{U}_{\mathrm{scheduled}} \leftarrow \emptyset$          // Initialize set of scheduled UEs
        $s_{P,r} \leftarrow 0$                                          // Initialize performance metric for RB $r$
        for $n = 1 \ldots N_{\max,P}$ do                                // Loop over all MIMO layers
            $u_{\mathrm{best}} \leftarrow$ null                         // Stores best UE
            $w_{\mathrm{best}} \leftarrow -\infty$                      // Stores best scheduling weight
            for all UE $u \in P.\mathcal{U} \setminus \mathcal{U}_{\mathrm{scheduled}}$ do
                $P_s \leftarrow 0$
                $P_i \leftarrow 0$
                for all $b \in \mathcal{B}$ do
                    if $b \in P.\mathcal{B}$ then                       // TP belongs to the partition serving the UE
                        $P_s \leftarrow P_s + P_{\mathrm{TX},r} \cdot h_{u,b,r} \cdot N_{\mathrm{TX}}$   // Add transmit power to signal power
                    else                                                // TP does not belong to the partition serving the UE
                        $P_i \leftarrow P_i + P_{\mathrm{TX},r} \cdot h_{u,b,r} \cdot N_{\mathrm{TX}}$   // Add transmit power to interference
                    end if
                end for
                $\gamma_{u,r} \leftarrow \frac{P_s/(|\mathcal{U}_{\mathrm{scheduled}}|+1)}{P_i+\sigma^2}$
                $w_u \leftarrow$ CalcWeight($\gamma_{u,r}$)
                if $w_u > w_{\mathrm{best}}$ then
                    $w_{\mathrm{best}} \leftarrow w_u$
                    $u_{\mathrm{best}} \leftarrow u$
                end if
            end for
            $s_{P,r,n} \leftarrow$ CalcPerformance($P$, $r$, $\mathcal{U}_{\mathrm{scheduled}} \cup u_{\mathrm{best}}$)
            if $s_{P,r,n} > s_{P,r}$ then
                $s_{P,r} \leftarrow s_{P,r,n}$
                $\mathcal{U}_{\mathrm{scheduled}} \leftarrow \mathcal{U}_{\mathrm{scheduled}} \cup u_{\mathrm{best}}$
            end if
        end for
        $s_P \leftarrow s_P + s_{P,r}$
        $\boldsymbol{\mathcal{A}}_P\{r\} \leftarrow \mathcal{U}_{\mathrm{scheduled}}$
    end for
    return $s_P$, $\boldsymbol{\mathcal{A}}_P$
end function
```

Algorithm 5.3 Calculation of the total performance of the scheduled UEs

```
function CalcPerformance(Partition $P$, RB $r$, Scheduled UEs $\mathcal{U}_{\text{scheduled}}$)
    $s_{P,r} \leftarrow 0$
    for UE $u \in \mathcal{U}_{\text{scheduled}}$ do
        $P_s \leftarrow 0$
        $P_i \leftarrow 0$
        for all $b \in \mathcal{B}$ do
            if $b \in P.\mathcal{B}$ then
                $P_s \leftarrow P_s + P_{\text{TX},r} \cdot h_{u,b,r} \cdot N_{\text{TX}}$
            else
                $P_i \leftarrow P_i + P_{\text{TX},r} \cdot h_{u,b,r} \cdot N_{\text{TX}}$
            end if
        end for
        $\gamma_{u,r} \leftarrow \frac{P_s/|\mathcal{U}_{\text{scheduled}}|}{P_i+\sigma^2}$
        $w_u \leftarrow$ CalcWeight($\gamma_{u,r}$)
        $s_{P,r} \leftarrow s_{P,r} + w_u$
    end for
    return $s_{P,r}$
end function
```

already scheduled UEs, because EPA is applied, which would reduce the transmit power of already scheduled UEs. Thus, the total performance might decrease by adding a new UE. Only in case the total performance is increased, the set of scheduled UEs is updated and the new performance is stored (Lines 29 to 32). Although the loop over the MIMO layers could be terminated, if adding more UEs does not yield an improved performance, this is not shown in Algorithm 5.2, because continuing the loop results in the same resource allocation.

After the loop over the MIMO layers, the best set of UEs to be served on RB r has been found, such that the performance metric and resource allocation can be updated accordingly (Lines 34 and 35). After this procedure has been performed for each RB individually, the performance metric s_P and resource allocation $\boldsymbol{\mathcal{A}}_P$ of partition P are returned in Line 37.

It should be noted that the scheduler is assumed to have complete channel knowledge, so it is able to estimate the hypothetical SINR accurately. In real systems this is not the case, as the scheduler has to operate with the CSI provided by the UEs. The complete channel knowledge in combination with the assumption that all TPs not contained in the own partition always transmit influence the scheduling decisions. While the complete knowledge in general leads to better performance, the opposite is caused by the full-interference assumption, which overestimates the interference, because neighboring TPs do not always transmit on all RBs. Even if both effects influence the scheduling decisions, this does not pose a problem to the following evaluations, because they focus on the overall system performance of DCCSF and not on scheduling in particular.

Scheduling Effort

Due to the greedy scheduling approach and the usage of the provided partitionings, the scheduling effort is significantly smaller in comparison to the total number of scheduling options. For each partition of all partitionings of a cluster, the scheduling algorithm evaluates as many scheduling options as the maximum number of MIMO streams allows:

$$N_{\text{options},P}^{\text{DCCSF}} = \min\left(|P| \cdot N_{\text{TX}}, N_{\text{UE}}\right) \tag{5.19}$$

In typical deployments and also in the following evaluations, the number of MIMO streams is limited by the total number of transmit antennas, because in many cases more UEs are available. The total number of scheduling options per cluster is:

$$N_{\text{options}}^{\text{DCCSF}} = \sum_{\mathcal{P} \in \mathcal{S}_{\mathcal{P}}} \sum_{P \in \mathcal{P}} N_{\text{options},P}^{\text{DCCSF}} = \sum_{\mathcal{P} \in \mathcal{S}_{\mathcal{P}}} \sum_{P \in \mathcal{P}} \min\left(|P| \cdot N_{\text{TX}}, N_{\text{UE}}\right) \tag{5.20}$$

5.2 Metrics and Performance Measures

For the assessment of the considered scenarios, we use two groups of metrics. The first group is presented in Section 5.2.1 and quantifies the user mobility and distribution. The second group of metrics deals with the system performance from the network perspective and is introduced in Section 5.2.2.

5.2.1 User Mobility and User Distribution

As we have seen in Section 3.1.3, the mobility of the users in the network plays an important role for the achievable network performance. Therefore, measures to describe the mobility and location of the users are necessary. Multiple mobility metrics have been proposed from traffic and transportation engineering as well as network engineering, mainly under the aspect of meshed device-to-device communication [BSH03; TTV07; XY13]. However, it has been shown in a supervised research thesis that especially two metrics are well suited to describe the user mobility in the context of cellular networks [Tru17]. In the following, we present these metrics in more detail.

5.2.1.1 Local Density

We use the local density to describe the user distribution in the system. To measure the local user density, we divide the area under consideration into regular squares as depicted in Figure 5.5. For the density calculation only those squares containing at least one user are considered. The local user density ρ is defined as the average user density in those squares:

$$\rho = \frac{1}{|\mathcal{N}_s|} \sum_{n \in \mathcal{N}_s} \frac{N_{\text{UE},n}}{A_s} \tag{5.21}$$

With $\mathcal{N}_s$ being the set of squares containing at least one user, $N_{\text{UE},n}$ the number of users in square n and A_s the area of one square. We use an edge length of 20 m, resulting in an area of $A_s = 400\,\mathrm{m}^2$. The size of the squares must be small enough to detect imbalances in the small-scale user distribution. A real-world example are hotspots at highly visited places like train stations or shopping malls. If the squares would have a size of square kilometers, these imbalances would not be reflected in the local user density. On the other hand, the squares must be large enough such that multiple user are located within them. E.g., if the size of the squares would be only a fraction of a square meter, the local density according to Equation (5.21) would be in the most cases close to one.

Because the local density is only calculated for squares with at least one user, this metric does not reflect the total number of users in the considered area. E.g., the local density is the same for a scenario where only one square contains one user and in a scenario where all squares contain one user each. A solution for this problem is to regard either the total number of users in the considered area in addition to the local density or to consider the total number of squares with at least one user.

5.2.1.2 Similarity of Mobility

The similarity of the users' movements in the scenario, sometimes also called degree of spatial dependence [BSH03], describes how similar the movements are in terms of direction and speed. The similarity between two users has values ranging from -1 to 1. A value of 1 means that both users move in the same direction with same speeds. A value of 0 means that the travel directions of the two users are orthogonal to each other. Values between 0 and 1 are either caused by movements with different speeds or if the angle between the travel directions is between 0° and 90°. A similarity of -1 means traveling in opposite directions with the same speed. The similarity of the travel directions is determined according to the cosine-similarity:

$$\text{similarity}\left(u_i, u_j\right) = \cos\left(\varphi_{u_i,u_j}\right) = \frac{\mathbf{v}_{u_i} \cdot \mathbf{v}_{u_j}}{|\mathbf{v}_{u_i}| \cdot |\mathbf{v}_{u_j}|} \tag{5.22}$$

With $\mathbf{v}_{u_i}$ and $\mathbf{v}_{u_j}$ being the velocity of users u_i and u_j, respectively. Consequently, $|\mathbf{v}_{u_i}|$ and $|\mathbf{v}_{u_j}|$ are the speeds of the two users. The similarity between the speeds is determined by the ratio of the lower speed divided by the higher speed. The leads to the following definition of the similarity of the mobility of the two users:

$$S_{u_i,u_j} = \frac{\min\left(|\mathbf{v}_{u_i}|, |\mathbf{v}_{u_j}|\right)}{\max\left(|\mathbf{v}_{u_i}|, |\mathbf{v}_{u_j}|\right)} \cdot \frac{\mathbf{v}_{u_i} \cdot \mathbf{v}_{u_j}}{|\mathbf{v}_{u_i}| \cdot |\mathbf{v}_{u_j}|} \tag{5.23}$$

The similarity S of the movement of all users in the system is defined as the average of the similarity of all user pairs.

$$S = \frac{N_{\mathrm{UE}}(N_{\mathrm{UE}} - 1)}{2} \sum_{i=1}^{N_{\mathrm{UE}}-1} \sum_{j=i+1}^{N_{\mathrm{UE}}} S_{u_i,u_j} \tag{5.24}$$

5.2.2 Network Performance

The network performance is determined by different rate measurements. Figure 5.3 provides an overview and shows additionally where the rates are measured in the network.

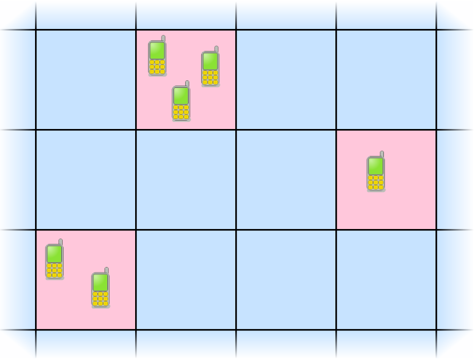


Figure 5.5: Density measurement

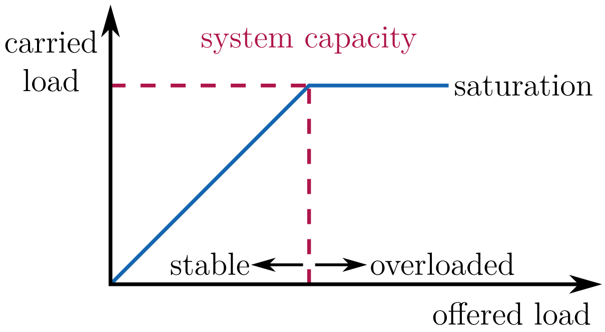


Figure 5.6: Relation between offered and carried traffic

5.2.2.1 System Capacity

The system capacity is used as a basic metric to assess the network performance. A better system configuration results in a higher system capacity. The system capacity describes the maximum data rate the system can handle. This is illustrated in Figure 5.6, which shows the relation between offered and carried traffic load. In case of a stable system operation the carried load is equal to the offered load. When the system capacity is reached, the carried traffic rate saturates and the additional offered traffic is buffered in the queues. In the simulation model, the offered rate is measured by summing up the rates generated by the traffic generators. The carried traffic is measured by summing up the traffic received by the UEs (see Figure 5.3).

Due to the stochastic nature of the offered traffic, the border between the stable region and the overloaded system is not as clearly visible as in the ideal example. Nevertheless, the saturation of the carried load is still visible.

5.2.2.2 Rate per UE

While the system capacity is an important performance indicator for the network operator, it is not relevant from the user's perspective. Therefore, we measure also the rate per UE r_u. This rate is measured at the output of each UE (see Figure 5.3) and is averaged over all users, which is denoted as R_u:

$$R_u = \frac{1}{N_{\mathrm{UE}}} \sum_{u \in \mathfrak{U}} r_u \tag{5.25}$$

5.2.2.3 Rate per Traffic Object

While the total system capacity is a relevant metric for the operator of the network and the rate per UE provides long-term averages, the users are interested in their perceived network performance. In case of web browsing traffic the relevant metric for the users is how long it takes to transmit the traffic objects. Because this absolute time depends on the traffic object sizes, we take the object size into account and define the rate per traffic object r_o as follows:

$$r_o = \frac{\text{traffic object size}}{\text{transmission time}} = \frac{\text{traffic object size}}{t_{\text{RAN}} + \tau_c} \tag{5.26}$$

The transmission time is composed of the actual RAN transmission time t_{RAN} and the core network delay of $\tau_c = 20\,\text{ms}$ as shown in Figure 5.3. The transmission time t_{RAN} depends on the channel quality and scheduling decisions, and denotes the time to completely transmit the traffic object to the UE. We do not include the buffer delay in the transmission time, because we want to measure the rate per traffic object independently of the system load. If the buffer time would be included, the rate per traffic object would go to zero for an overloaded system, even though many traffic objects are transmitted. As a consequence of the definition of the transmission time, the rate per traffic object is only partly influenced by the rate of the wireless link. This approach is similar to the concept of transactions introduced by Proebster et al. [Pro+12]. This rate definition is an indicator for the perceived QoE. A higher rate generally results in higher user satisfaction. The average rate of all transmitted traffic objects is denoted as R_o.

5.3 Evaluation Methodology

The following evaluation results show the outcome of ten independent simulation runs. This means for each parameterization of the simulation model, the simulation is executed ten times, with another initialization of the random number generators. Therefore, for each simulation run, the traffic demands and positions of the UEs differ. After executing the independent replications, the results are combined, such that the overall statistics of the considered metrics are calculated. Additionally, this method allows calculating confidence intervals for the mean values using the Student's t-test. The confidence intervals in the following represent the 95 % confidence intervals.

The simulated time for one independent replication is 500 s. The actual simulation phase is preceded by a transient phase of 400 s, in which the simulation is executed normally but no statistics are gathered. These durations have been determined by calibration runs in which the downlink queue sizes have been observed. While the queue sizes increase in the transient phase, they remain stationary in the evaluation phase as long as the system is not overloaded.

In case that the ratio of two simulation results for two system configurations is shown (e.g., in Sections 6.1.5 and 6.2.1.1), these ratios are calculated by Fieller's method [Fie54], which provides the mean value of the ratio as well as confidence intervals. Even if Fieller's method allows comparing simulation results obtained from arbitrary number of independent replications, we only consider the case where the same number of runs are executed to obtain the numerator and denominator ($N_N = N_D = N$). The overall mean value $\bar{q}$ is intuitively defined as:

$$\overline{q} = \frac{\overline{n}}{\overline{d}} = \frac{\frac{1}{N_N}\sum\limits_{i=1}^{N_N} n_i}{\frac{1}{N_D}\sum\limits_{i=1}^{N_D} d_i} \tag{5.27}$$

n_i and d_i are the mean values of the independent replications, which are to be compared. $\overline{n}$ and $\overline{d}$ denote the overall mean of the two compared system configurations.

We show various performance gains of system configurations with CoMP in comparison to systems without CoMP in the next chapter. The relative gain is defined as:

$$g = \left(\frac{p_c}{p_n} - 1\right) \cdot 100\,\% \tag{5.28}$$

p_c is the considered performance value, e.g., the system capacity, the carried traffic rate or the rate per traffic object, of the system with CoMP and p_n is the corresponding performance value of the system without CoMP. The ratio ${}^{p_c}\!/\!{}_{p_n}$ is calculated with the Fieller's method as described before. This also yields confidence intervals for the relative gain.

6 Performance Evaluation

This chapter is dedicated to the performance evaluation of DCCSF. The evaluation is split into two parts. In Section 6.1, we present the fundamental behavior of DCCSF and introduce the used evaluation methods. The second part in Section 6.2 deals with the influence of different dynamic effects on DCCSF. Finally, we sum up the results and provide implementation recommendations in Section 6.3.

6.1 Fundamental System Behavior

In a first step, we present how CoMP and dynamic clustering affect the system performance. In the course of the following sections, we also analyze the influence of the configuration parameters of the dynamic clustering algorithm, i.e., the allowed maximum cluster size and the cluster reconfiguration interval.

6.1.1 System Capacity

In the first evaluation, we show how the system capacity is influenced by different CoMP clustering schemes and different numbers of transmit (TX) antennas at the TPs. To do so, the offered traffic rate is varied by changing the IAT of the web browsing traffic model. The system capacity is evaluated by measuring the carried traffic rate. We also evaluate the influence of different mobility models. For the simulations the total number of UEs is 570 and 30 groups are configured for the Virtual Track mobility model. This results in an average group size of 19 UEs. In the case of Virtual Track mobility all UEs move with a constant speed of 50 km/h. Although the UEs do not move in the Static Users model, the fast fading is configured as if they would move with 50 km/h. The cluster reconfiguration interval is set to $T_R = 1\,\text{ms}$. The maximum cluster size for DCCSF is $S_{C,\max} = 3$.

Figures 6.1 and 6.2 present the results for Static Users and Virtual Track mobility, respectively. Comparing the results with the theoretical maximum data rate of approximately 491 Mbit/s for a system with a single transmit antenna per TP (see Equation (5.8)), it becomes evident that the achievable rates are significantly smaller. For the system without CoMP the Static Users model achieves a carried traffic rate of approximately 75 Mbit/s and the Virtual Track model of 69 Mbit/s. The difference between theoretic and real rates can be explained by the influence of channel attenuation, interference and signaling overhead. While for the theoretic estimation of the data rate a SINR of 24 dB is assumed, the average SINR in the simulation is approximately 3.6 dB for the Static Users model and 9.2 dB for the Virtual Track model. The reason for the smaller carried traffic rate despite a better SINR for the Virtual Track can be explained by the fact that not all 57 TPs are used by the configured 30 groups.

We can observe in the figures that the system capacity is increased by introducing CoMP in the network. However, the gains depend on the applied clustering scheme and the mobility

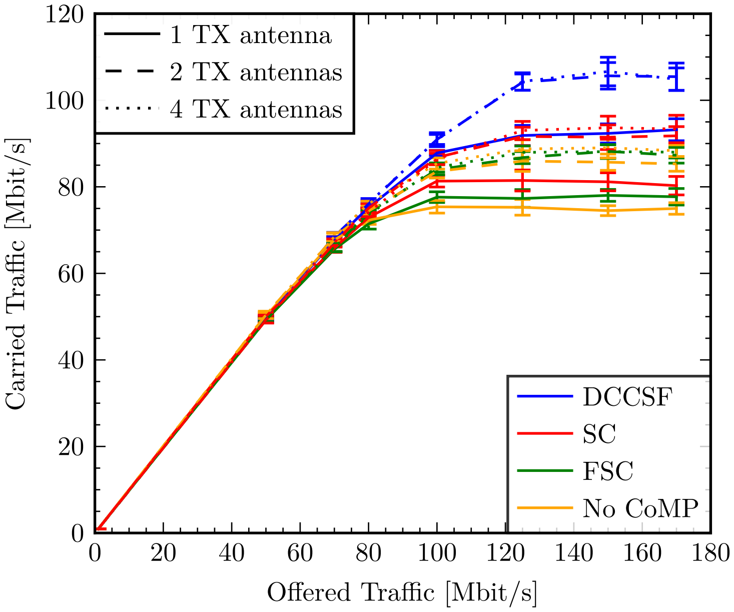


Figure 6.1: System capacity for static users

model. For an easier comparison Table 6.1 provides an overview of the system capacity improvements in comparison to the system without CoMP. The values in subscript and superscript indicate the 95 % confidence intervals. The asymmetry of the confidence intervals is a result of the applied method to calculate the confidence interval for ratios of simulation results. The improvement is defined by comparing the carried traffic for the maximum offered load of 170 Mbit/s for the Static Users model and 200 Mbit/s for the Virtual Track mobility model. Note that the systems with two and four transmit antennas and Virtual Track mobility are not fully saturated at these offered loads, such that the improvements could be even higher. The increasing simulation effort prohibits to evaluate the system at higher offered loads.

Independent of the mobility model and the number of transmit antennas Site Clustering (SC) performs better than Facing Sectors Clustering (FSC), while DCCSF always performs best. The reason for these results is that FSC is mainly beneficial for UEs located at the border between different sites. At these locations the overall channel attenuation is high and therefore the additional CoMP gain is comparatively small. SC is limited to cooperation between the sectors of the same site, such that UEs on the sector borders profit. These UEs have generally lower channel attenuations compared to UEs at site borders such that their achievable data rate is higher and as a consequence also the gain introduced by CoMP. DCCSF adapts the clustering to the actual positions of the UEs, such that the best performance improvements are achieved.

By comparing the results for the different mobility models, it can be seen that the gain of CoMP and especially dynamic clustering is higher for the Virtual Track mobility model. The reason is that in the Virtual Track model distinct groups of UEs are moving through the scenario. The dynamic clustering algorithm defines the clusters such that the groups profit. Because a group consists of several UEs, all UEs of the group profit at the same time. On the other hand, the UEs are distributed uniformly in the scenario in the Static

Mobility	transmit antennas	FSC	SC	DCCSF
Static Users	1	$3.61^{+1.81}_{-1.81}\,\%$	$7.03^{+2.62}_{-2.59}\,\%$	$24.27^{+2.75}_{-2.74}\,\%$
	2	$2.34^{+2.03}_{-2.00}\,\%$	$7.61^{+2.00}_{-1.98}\,\%$	$23.63^{+2.13}_{-2.15}\,\%$
	4	$-0.37^{+1.77}_{-1.74}\,\%$	$5.69^{+3.44}_{-3.43}\,\%$	$18.71^{+2.25}_{-2.25}\,\%$
Virtual Track	1	$23.95^{+2.02}_{-2.06}\,\%$	$38.06^{+4.01}_{-4.15}\,\%$	$55.99^{+1.84}_{-1.84}\,\%$
	2	$13.67^{+1.47}_{-1.44}\,\%$	$29.53^{+3.69}_{-3.73}\,\%$	$50.99^{+3.61}_{-3.40}\,\%$
	4	$1.55^{+1.74}_{-1.68}\,\%$	$15.53^{+2.95}_{-2.95}\,\%$	$29.41^{+3.82}_{-3.53}\,\%$

Table 6.1: System capacity gain for different mobility models and numbers of transmit antennas

Users model. Therefore, the defined clusters are always a compromise and the additional gains are smaller.

Finally, the influence of different numbers of transmit antennas is analyzed. Generally more transmit antennas at the TPs increase the system capacity. However, the relative gains in comparison to the system without CoMP are reduced (see Table 6.1). One reason for this behavior is the increased channel measurement overhead, which reduces the usable capacity. Another reason is that adding more transmit antennas improves the received signal power, but at the same time the interference is increased, too. In the case of Static Users, four transmit antennas and FSC clustering this even reduces the system capacity in comparison to the system without CoMP. By adding transmit antennas to TPs, the total available transmit power increases linearly with the number of antennas, because we assume a separate power amplifier for each antenna. At the same time, more UEs can be served by making use of MU-MIMO and JT. Because in the simulation model the transmit power is distributed equally to the served UEs (see Section 5.1.5.2), the signal power per UE does not change with the number of transmit antennas, if as many UEs are served as the number of transmit antennas allows. On the other hand, we model a pessimistic assumption of the interference, which assumes full interference from all transmit antennas of neighboring TPs. Thus, the interference is increased by adding more transmit antennas. Therefore, we generally underestimate the performance in the simulation and real implementations could achieve even higher system capacities.

From these results we can conclude that introducing CoMP and especially DCCSF improves the system capacity. For network operators this means that they can increase their network's capacity without installing new TPs or acquiring new frequency bands. A couple of further questions are raised from the first evaluations. E.g., how the two parameters of the dynamic clustering algorithm, i.e., cluster size and cluster reconfiguration interval, influence the performance. Also, we considered the performance only from the perspective of the network, but it would be interesting to assess the performance from a user's point of view, too.

6.1.2 Rate Distributions

In the following, we have a closer look on the performance from the user perspective. Therefore, Figure 6.3 shows the Cumulative Distribution Function (CDF) of the rate per

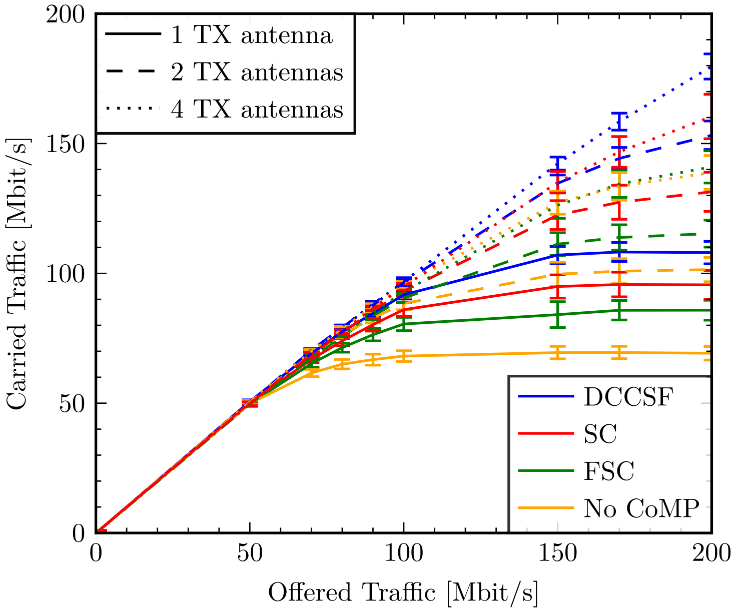
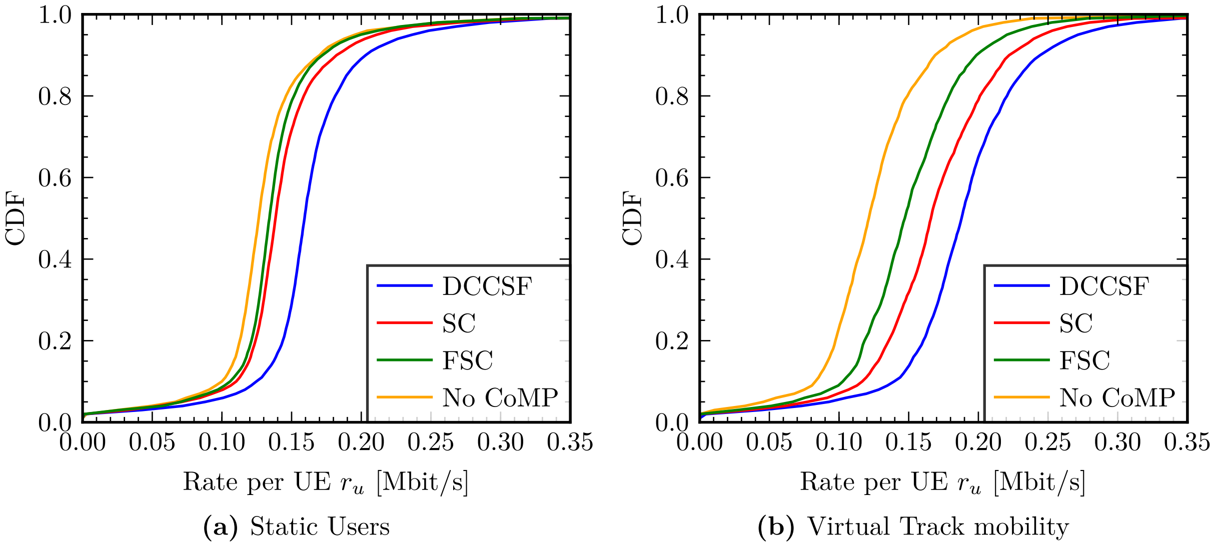


Figure 6.2: System capacity for virtual track mobility



(a) Static Users

(b) Virtual Track mobility

Figure 6.3: Distributions of the rate per UE for one transmit antenna per TP

UE for the Static Users and the Virtual Track mobility models. The system is configured as before, with the difference that the offered traffic rate is set to 150 Mbit/s. Additionally, only systems with a single transmit antenna per TP are considered, because the results for system with four transmit antennas per TP are similar.

With these results it becomes visible which UEs profit from CoMP. UEs with bad channel conditions, i.e., those who experience a low rate in the system without CoMP, benefit from CoMP and especially from DCCSF. These UEs are often called cell-edge users and are defined as those UEs that only achieve a rate of the 5th percentile of the overall rates per UE [3GPP 36.913]. Table 6.2 shows the improvements of the 5th and 10th percentile of the rates per UE if CoMP is applied in comparison to the system without CoMP. Similar as for the system capacity, the gains are higher with the Virtual Track mobility model.

Mobility	Percentile	FSC	SC	DCCSF
Static Users	5th	5.68 %	6.96 %	32.08 %
	10th	5.95 %	10.30 %	24.76 %
	95th	2.06 %	7.98 %	21.04 %
	99th	1.60 %	4.68 %	7.99 %
Virtual Track	5th	45.79 %	68.68 %	102.96 %
	10th	23.73 %	40.51 %	68.10 %
	95th	15.94 %	29.36 %	42.76 %
	99th	16.73 %	30.86 %	44.90 %

Table 6.2: Rate per UE percentile improvement

Also, the improvement of the 5th percentile is higher than for the 10th percentile, which means that UEs with worse channel conditions benefit more from CoMP. For Static Users this is only the case for DCCSF.

Figure 6.3 additionally shows that CoMP achieves an almost constant rate improvement for a wide range between the 10th and 90th percentiles, as the distributions are almost parallel but shifted to the right. This indicates that CoMP is beneficial for a large fraction of the UEs. In contrast, UEs with a good channel quality, i.e., UEs which are already served with a high rate in the system without CoMP, profit less from the introduction of CoMP, as shown in Table 6.2 for the 95th and 99th percentiles. Nevertheless, the gains for those UEs are also higher, if the UEs move in groups, which is the case in the Virtual Track mobility model.

6.1.3 Cluster Size

In the next step, we evaluate how the performance of DCCSF is influenced by the maximum allowed cluster size. For this evaluation we use the same configuration as before, with the differences that the offered traffic rate is fixed to 150 Mbit/s and the maximum cluster size $S_{C,\text{max}}$ is varied. In the following, only the results for the Virtual Track model with 570 UEs and 30 groups are shown, as the behavior of the Static Users model is similar.

6.1.3.1 Relation between Maximum and Configured Cluster Size

First, we compare the maximum allowed cluster size with the sizes of the actually configured clusters in Figure 6.4. Only a single transmit antenna per TP is configured. On the x-axis the maximum cluster size is varied from one to seven and the dashed lines show the average size of the configured clusters on the y-axis. As a reference, the static clustering schemes and the system without CoMP are included, too. Both static clustering schemes are only defined for $S_C = 3$ and the system without CoMP has by definition $S_C = 1$. The results in the figure have been expanded to ease comparison with dynamic clustering.

The figure reveals that the actually configured cluster size increases with increasing maximum cluster size. However, there is a significant difference between allowed maximum

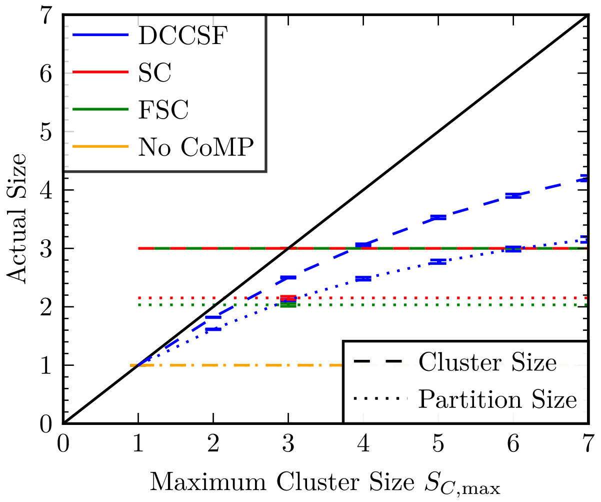


Figure 6.4: Comparison between maximum and configured cluster size

Table 6.3: Probability of partition sizes for $S_{C,\max} = 3$

Clustering	Size 1	Size 2	Size 3
FSC	34.9 %	24.6 %	40.5 %
SC	31.0 %	22.8 %	46.2 %
DCCSF	34.8 %	18.8 %	46.4 %

and configured cluster size, especially for larger maximum cluster sizes. The reasons are twofold. First, as the UEs move in groups, the UEs of the same group report similar TPs in their CMR messages, which reduces the number of available cluster candidates. Second, the reduced set of cluster candidates renders it impossible to select only clusters of the allowed maximum size. Therefore, the dynamic clustering algorithm selects only a limited number of clusters with maximum size and uses clusters with smaller sizes to include all TPs in the clustering.

The dotted lines in Figure 6.4 show the average sizes of the partitions the schedulers select for scheduling. The average partition size must be smaller than or equal to the average cluster size in all cases. A small difference between cluster and partition size indicates that the cluster is actually useful to serve the UEs with CoMP. The equality of cluster and partition size would mean that JT is applied for all TPs within the cluster to serve the UEs cooperatively. On the other hand, if the configured clusters are not beneficial for the UEs, the schedulers would select partitions of size one and thereby serve the UEs without CoMP. For both static clustering variants the average partition size is a little above two (2.03 for FSC and 2.15 for SC). The difference between configured cluster size and average partition size for $S_{C,\max} = 3$ is significantly smaller for DCCSF in comparison to FSC and SC. The difference for DCCSF is 0.39, while it is 0.85 for SC and 0.97 for FSC. In general, the ratio of average partition size and configured cluster size, i.e., the relative difference, is smaller for DCCSF for all considered maximum cluster sizes.

Table 6.3 provides a deeper insight into the size distribution of the used partitions for a maximum cluster size of $S_{C,\max} = 3$. For this maximum cluster size only partitions with sizes of one, two or three TPs are possible. From the results we can conclude that the tendency to use partitions with size three increases from FSC to SC and DCCSF. This means that the clusters configured by DCCSF are actually useful, such that they are used more frequently by the schedulers. Even though the probability of using partitions of size three is similar for SC and DCCSF, it should be noted that the average cluster size of

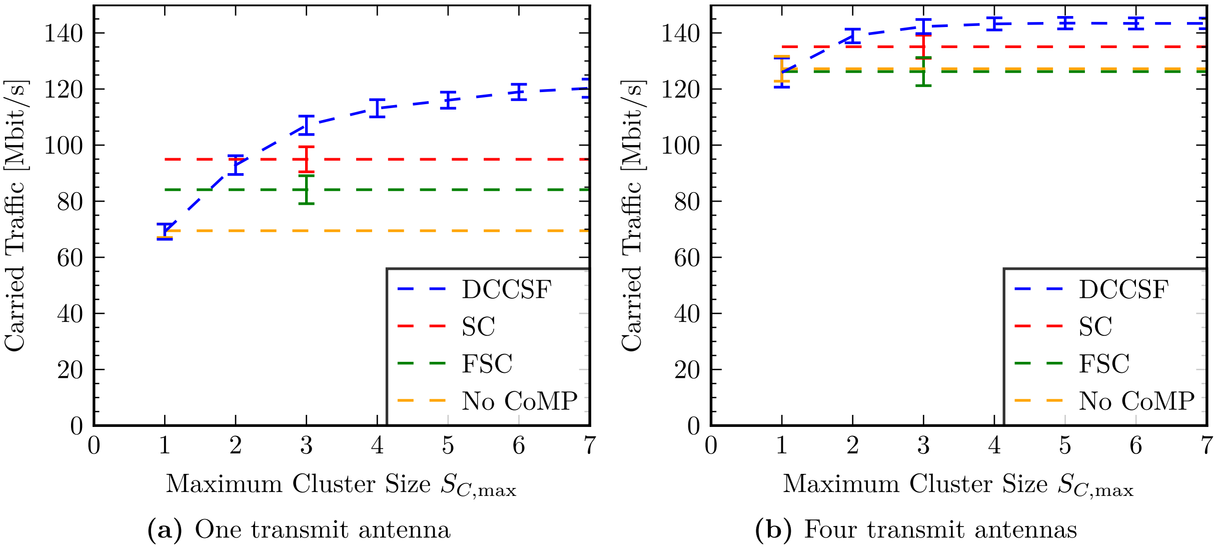
DCCSF with $S_{C,\max} = 3$ is approximately 2.5 and therefore it is often not possible to use partitions of size three. It is also visible that in more than one third of the cases UEs are served by a single TP. This indicates that from a performance point of view it is not always beneficial to apply CoMP to all TPs of a cluster.

6.1.3.2 Influence on Performance

For now, we have seen which clusters and partitions are selected by DCCSF. In this section, we concentrate on the impact of the cluster size on the achievable performance in terms of system capacity and rate per traffic object. Figures 6.5a and 6.5b present the system capacity for system configurations with one and four transmit antennas per TP. Figures 6.6a and 6.6b show the rate per traffic object, again for one and four transmit antennas. The system is loaded with an offered traffic rate of 150 Mbit/s in all cases. We observe that DCCSF outperforms static clustering and the system without CoMP, independently of the number of transmit antennas, if $S_{C,\max}$ is at least three. For DCCSF with a maximum cluster size of $S_{C,\max} = 2$ and a single transmit antenna per TP, the actually configured clusters are smaller than for SC. Nevertheless, DCCSF achieves a similar performance as SC. This indicates that DCCSF adapts the clusters to the actual positions of the UEs to improve the performance. The figures also reveal that the additional gains become lower with increasing maximum cluster sizes. There are two reasons for this behavior. The first is that even if larger clusters would be allowed, DCCSF cannot always make use of this, as discussed in the previous section. The second reason is that the benefit of larger clusters is generally limited, because if a UE would be served by TPs not in the direct vicinity of this UE, the additional improvement of the received signal power is comparatively small due to the larger channel attenuation. Therefore, it is often more efficient to use smaller clusters and serve more UEs simultaneously at the cost of increased interference.

Comparing the influence of different numbers of transmit antennas on the system capacity (in Figure 6.5) and the rate per traffic object (in Figure 6.6), we can remark that the principal behavior is similar for both configurations. However, DCCSF with four transmit antennas outperforms all other systems even with a maximum cluster size of $S_{C,\max} = 2$. The case of FSC shows that combining CoMP with more transmit antennas is not always reasonable, as the system capacity is almost the same as for the system without CoMP and the rate per traffic object is even reduced. Two reasons explain this behavior. The first is the used CoMP and capacity abstraction, which generally overestimates the interference and underestimates the received signal power in the case of CoMP. The second is that FSC is mainly suitable for UEs located on the border between different sites, where the signal power is generally low.

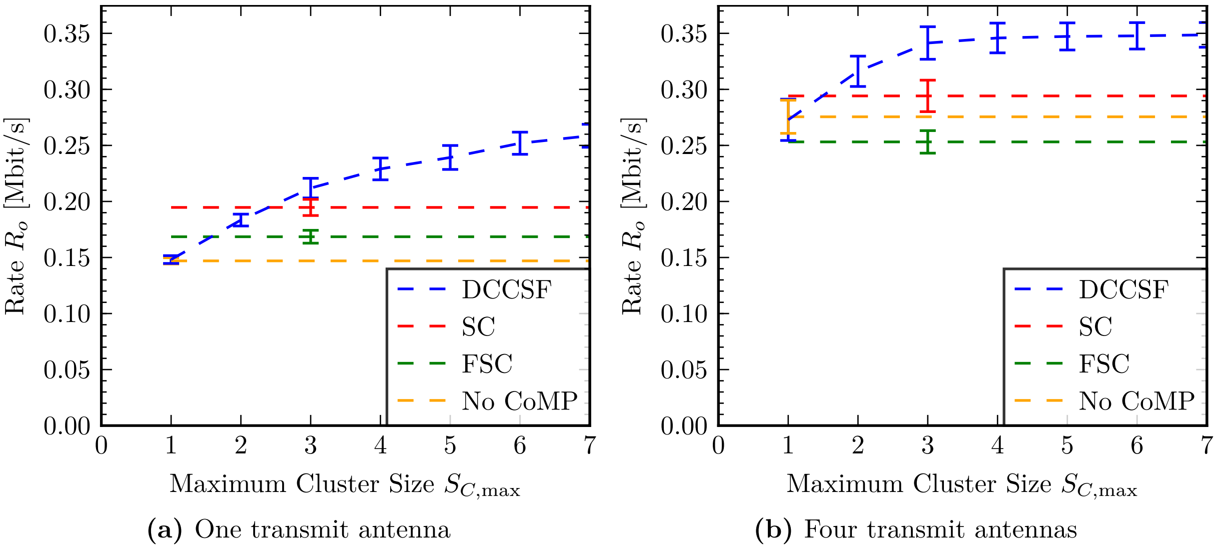
Another difference between the two system configurations with one and four transmit antennas per TP should be noted. In both cases an offered traffic rate of 150 Mbit/s is applied. However, only the system with one transmit antenna is overloaded at this rate (compare Figure 6.2). This explains why DCCSF only leads to minimal improvements of the system capacity in the case of four transmit antennas. Nevertheless, it improves the rate per traffic object, which shows that DCCSF is not only beneficial if a system operates close to the load limit, but it is also advantageous at lower loads.



(a) One transmit antenna

(b) Four transmit antennas

Figure 6.5: Influence of maximum cluster size on the system capacity



(a) One transmit antenna

(b) Four transmit antennas

Figure 6.6: Influence of maximum cluster size on the rate per traffic object

These results are promising from the perspective of introducing DCCSF in real systems, because the clusters do not necessarily have to be large to improve the system performance. Instead, it is sufficient to limit the maximum cluster size, which also limits the overhead of channel measurements and the complexity of precoding and signal processing.

6.1.4 Reconfiguration Interval

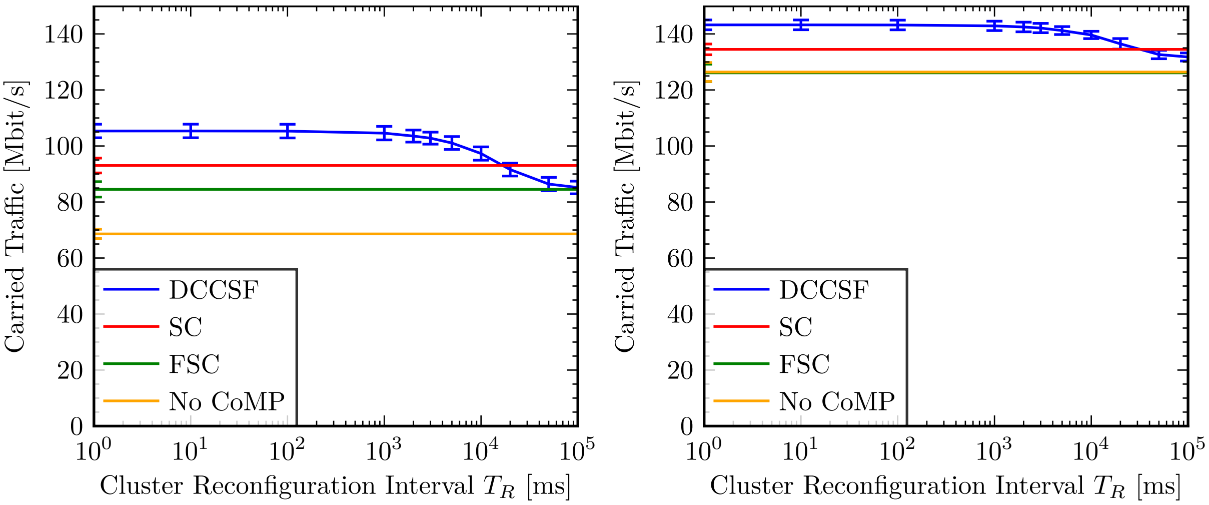
We have raised the question of how the second parameter of the dynamic clustering algorithm, the cluster reconfiguration interval, influences the achievable performance. Until now, we have assumed the optimal case that clusters can be reconfigured every TTI, which might be too optimistic in real systems. Therefore, we use a similar configuration as before, but vary the cluster reconfiguration interval in the range from 1 ms to 100 s. The maximum cluster size for the dynamic clustering is fixed to $S_{C,\max} = 3$. The system is loaded with 150 Mbit/s and the same configuration for the Virtual Track mobility model as before is used. Figures 6.7 and 6.8 present the simulation results of the system capacity and rate per traffic object for one and four transmit antennas per TP. The figures also show results of both static clustering schemes and the system without CoMP. For those configurations no reconfiguration interval is defined, such that the results are expanded to simplify the comparison.

As visible, the carried traffic as well as the rate per traffic object remains constant, if the cluster reconfiguration interval is increased from 1 ms up to 1 s. Only if T_R is further increased, the performance starts to degrade. This observation holds for both configured numbers of transmit antennas per TP, with the difference that the system with four transmit antennas achieves higher overall performances. The reason for the performance degradation is that the UEs move during the reconfiguration interval, such that the configured clusters become outdated and are not beneficial anymore. But even if the clustering is adapted only every 10 s, the performance is still better than the performance of the static clustering schemes or the system without CoMP. For longer reconfiguration intervals, the performance of DCCSF becomes worse than that of the static clustering schemes, because in static clusterings all TPs are contained in a cluster, while in DCCSF only the TPs reported in CMR messages are contained. Consequently, if a group leaves the coverage area of one cluster, it directly reaches the area of another cluster in case of static clustering. If a group leaves the area of a cluster in a system with DCCSF, it can happen that it does not directly reach another cluster, but has to be served by a single TP. This problem could be avoided, if for TPs not treated by the dynamic clustering algorithm a static clustering scheme is applied.

From a system implementation viewpoint these results are encouraging, as they indicate that it is not necessary to adapt the clustering in the range of milliseconds. This relaxes the requirements of periodic channel measurements and transmission of CMR messages. Also, established cloud technologies, like VMs or containers, could be an option to realize the Cluster Entities (CEs), as they allow setup times in the range of seconds. In the second part of the evaluation, where we have a closer look on the influences of dynamic network effects on DCCSF, we also assess the relation between different speeds of the UEs, the mobility model and the cluster reconfiguration interval (see Section 6.2.2).

6.1.5 Scheduling Effort Reduction

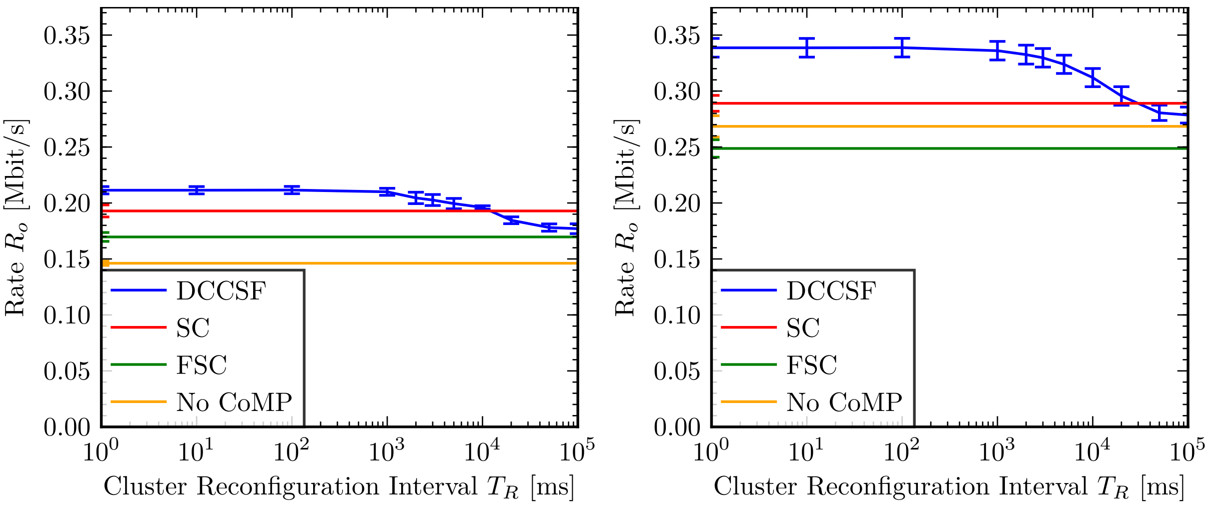
DCCSF provides the schedulers with additional information, which is used to limit the number of scheduling options the schedulers have to evaluate. This reduces the effort at the cost of potentially decreased performance. We only treat the first aspect, because evaluating the reduced performance in comparison to the ideal clustering, would require to find the optimal clustering first. This is not possible due to the complexity of the problem, especially if the cluster sizes become larger. Nevertheless, we have already seen in the previous sections that DCCSF results in significant performance gains. We evaluate the effort reduction by comparing the scheduling options that have to be analyzed in DCCSF (see Section 5.1.7) with the number of scheduling options resulting from exhaustive search



(a) One transmit antenna

(b) Four transmit antennas

Figure 6.7: Influence of the cluster reconfiguration interval on the system capacity



(a) One transmit antenna

(b) Four transmit antennas

Figure 6.8: Influence of the cluster reconfiguration interval on the rate per traffic object

as presented in Appendix A.2. We define the scheduling ratio α_s as the ratio of considered scheduling options to all scheduling options:

$$\alpha_s = \frac{N_{\text{options}}^{\text{DCCSF}}}{N_{\text{options}}^{\text{exhaustive}}} \cdot 100\,\% \tag{6.1}$$

While the number of scheduling options DCCSF considers is directly measured in the simulation, $N_{\text{options}}^{\text{exhaustive}}$ is calculated using the sizes of the configured clusters. Figures 6.9 and 6.10 show the number of evaluated scheduling options and the scheduling ratio for different maximum cluster sizes and mobility models. All blue curves show results of the Virtual Track mobility model while the red curves indicate results of the Static Users model. Due to the high simulation effort the maximum cluster size evaluated for the Static Users model is $S_{C,\max} = 6$. In this evaluation only a single transmit antenna per TP and an offered traffic rate of 125 Mbit/s is configured. For the Static Users model a total number

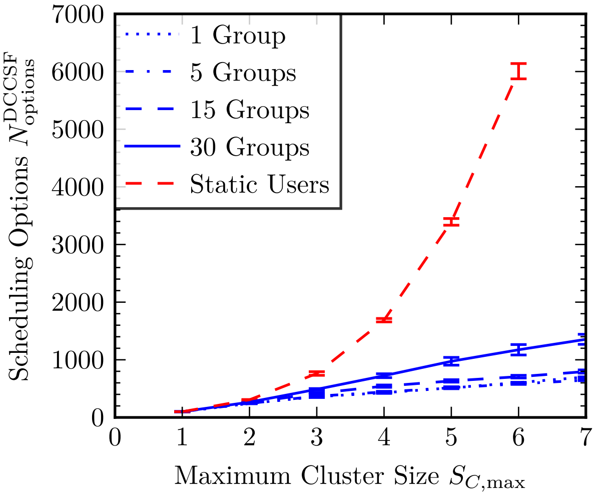


Figure 6.9: Number of scheduling options

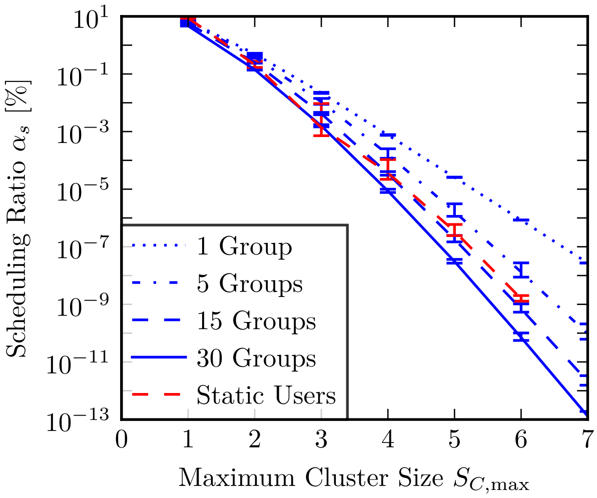


Figure 6.10: Scheduling effort reduction

of 570 UEs is used. In the Virtual Track mobility model, the number of UEs depends on the configured number of groups. Per group 19 UEs are added to the system, such that with 30 groups 570 UEs are in the system, too. The number of scheduling options for one and five groups in the Virtual Track model differ only by less than 8 % even for a maximum cluster size of $S_{C,\max} = 7$, such that the results are hardly distinguishable in Figure 6.9.

The scheduling ratio decreases with increasing maximum cluster size to very small values, which is mainly owed to the fact that $N_{\text{options}}^{\text{exhaustive}}$ grows rapidly with increasing maximum cluster size, while $N_{\text{options}}^{\text{DCCSF}}$ scales approximately linearly in case of Virtual Track mobility (see Figure 6.9). In case of the Static Users model $N_{\text{options}}^{\text{DCCSF}}$ increases superlinear, but still far less than $N_{\text{options}}^{\text{exhaustive}}$. The reason for the different scaling of the scheduling options is that in the Static Users model, the UEs are uniformly distributed, such that increasing the cluster size directly increases the number of UEs within the cluster. On the other hand, this is not the case for the Virtual Track mobility model, because the number of groups is small in comparison to the number of available TPs. As a consequence, the probability that increasing the cluster size also increases the number of UEs within the cluster is lower. This finding provides additional reasoning why the cluster size in real implementations should be kept small, as increasing the cluster size only results in small performance improvements, but comes at the cost of higher scheduling efforts, besides the additional drawbacks mentioned in the previous sections.

Comparing the results for the two mobility models, we observe that the scheduling ratio is in both cases in the same range. However, the number of groups in the Virtual Track model influences the scheduling ratio, as visible in Figure 6.10. More groups result in a lower scheduling ratio, because this also increases the total number of UEs in the system. DCCSF can handle more UEs better than the approach with exhaustive search, because a greedy scheduling approach is used.

In real implementations performing an exhaustive search over all scheduling options is not reasonable and the presented gains in terms of the scheduling ratio are too optimistic. We therefore conduct a second evaluation where $N_{\text{options}}^{\text{DCCSF}}$ is compared to a more realistic

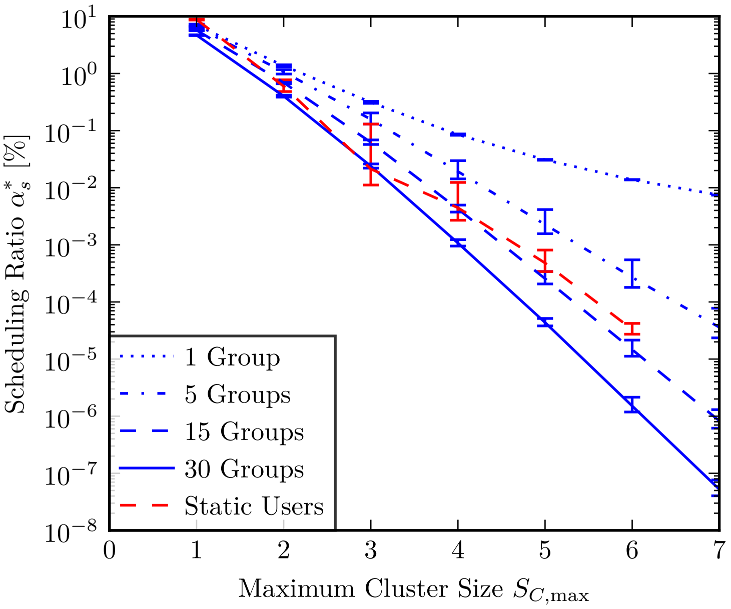


Figure 6.11: Scheduling effort reduction when using the defined partitionings

estimate of the number of scheduling options. This estimation is based on the partitionings as created by DCCSF, but instead of a greedy approach to schedule the UEs within a partition, all combinations to schedule the UEs are determined. For each partition the total number of UE combinations is determined by the following equation:

$$N_{\text{options},P}^{\text{binomial}} = \sum_{i=1}^{N_{\max,P}} \binom{N_{\text{UE}}}{i} \tag{6.2}$$

$N_{\max,P} = \min\left(|P.\mathcal{B}| \cdot N_{\text{TX}}, |P.\mathcal{U}|\right)$ represents the maximum number of MIMO streams possible within the partition. Then the total number of scheduling options $N_{\text{options}}^{\text{binomial}}$ is obtained by summing up the number of options for all partitions and partitionings. We define the modified scheduling ratio α_s^* as:

$$\alpha_s^* = \frac{N_{\text{options}}^{\text{DCCSF}}}{\sum\limits_{\mathcal{P}\in\mathcal{S}_{\mathcal{P}}} \sum\limits_{P\in\mathcal{P}} N_{\text{options},P}^{\text{binomial}}} \cdot 100\,\% \tag{6.3}$$

Figure 6.11 presents the results. In comparison to the previous results of the complete exhaustive search, the scheduling ratio is larger by several orders of magnitude. Nevertheless, for larger maximum cluster sizes, the number of scheduling options DCCSF considers is significantly smaller compared to the case where all scheduling options within the proposed partitions are examined. E.g., for a maximum cluster size of $S_{C,\max} = 3$, DCCSF only has to consider approximately 0.01 % to 1 % of the scheduling options.

6.1.6 Summary of the Fundamental System Behavior

We can summarize the findings of the first evaluations as follows:

- DCCSF significantly improves the system capacity as well as the perceived performance of the UEs in comparison to systems without CoMP and static clustering.

- The required cluster sizes are small and DCCSF even outperforms static clustering schemes, if the allowed maximum cluster size is smaller than in the static clustering schemes. Thereby, the overhead of signaling, channel measurement and CoMP signal processing is kept low.
- DCCSF tolerates cluster reconfiguration intervals in the range of seconds, which facilitates the practical implementation in real systems.
- The additional complexity to introduce DCCSF is kept low, because neither larger clusters nor frequent cluster reconfigurations are required. Also, the effort of scheduling is bearable, as it is sufficient to only consider few but reasonable scheduling options and still achieve significant performance gains.

6.2 Influence of Dynamic Effects

So far, we have seen the fundamental behavior of DCCSF. In the following, we examine the influence of different dynamic effects on the system performance. Therefore, we make use of the Virtual Track mobility model and assess the influence of mobility patterns, like group behavior, the density of UEs and different speeds. We also have a closer look on how CoMP influences the performance for traffic objects of different sizes.

In the previous evaluations, we have seen that the number of available transmit antennas influences the system capacity and rate per traffic object. Nevertheless, the principal behavior is similar for all considered configurations. As a consequence, we only consider systems with a single transmit antenna per TP in all following evaluations. This choice significantly reduces the simulation effort, but still allows to assess the system. Also, we restrict the evaluations to a maximum cluster size of three, because this offers a good trade-off between achievable CoMP gain and generated overhead. If not otherwise noted the users move with a speed of 50 km/h and the cluster reconfiguration interval is set to $T_R = 1\,\text{ms}$.

6.2.1 User Mobility Behavior

This section discusses the influence of the mobility behavior on the network performance. To allow a separation of the individual effects, we consider in Section 6.2.1.1 first purely static users, but vary their locations on the considered area. This allows to evaluate the influence of imbalances of the user density. Then we allow mobility again and make use of the Virtual Track model to analyze the relation of group behavior and the performance gains in Section 6.2.1.2.

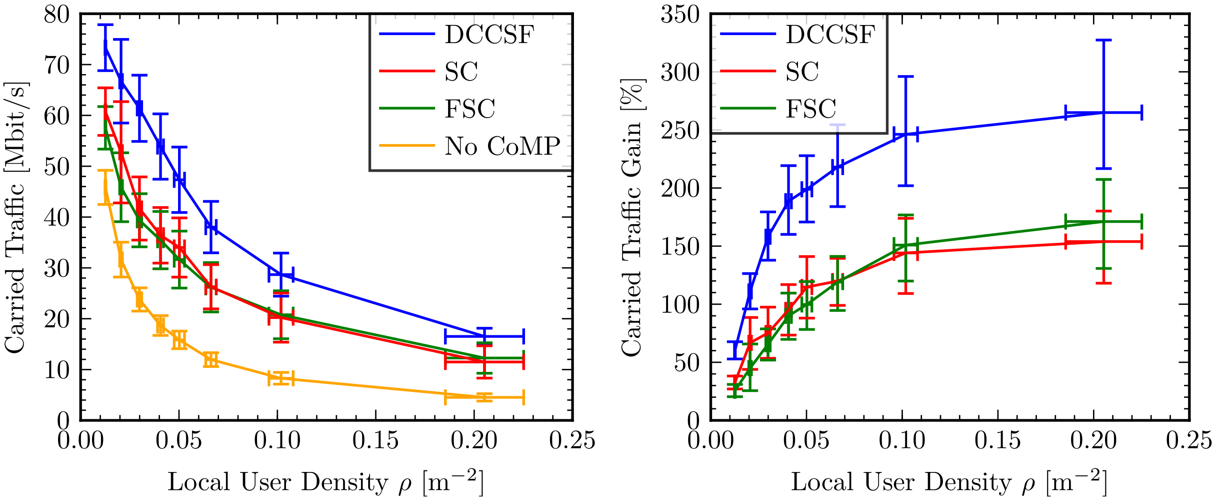
6.2.1.1 Local User Density

For the evaluation how the user density influences the system performance, we use the static hotspot mobility model (see Section 5.1.2.2). Even though the users are static, dynamics are introduced by the traffic model. These dynamics and the random locations of the hotspots allow DCCSF to configure multiple clusterings during the evaluation. The total number of users is kept constant while the number of hotspots is varied in the range from 1 to 19. The variation of the number of hotspots directly influences the local user density. The density is measured according to the definition from Section 5.2.1.1. A total uniform user distribution would result in a low local density, while a single hotspot results in a higher density of ${}^{N_{\mathrm{UE}}}\!/_{A_s}$. The user distribution also affects the system performance. Totally uniformly distributed UEs can make use of all available TPs and therefore the system performance is higher. If all UEs are located in a single hotspot on the other hand, only a single TP or in the case of CoMP the TPs around the hotspot are used to transmit data, which reduces the performance. In this section, we answer the question of how well DCCSF deals with the performance degradation if the users show a higher group behavior in comparison to static clustering or a system without CoMP.

Because the UEs do not move in this evaluation, the fast fading component of the channel model is disabled in the simulation model. The total number of UEs is set to 570. Additionally, the offered load is set to 150 Mbit/s.

In the results shown in Figures 6.12 and 6.13, the local user density is visible on the x-axis and is varied by changing the number of hotspots. The y-axis shows the carried traffic or the rate per traffic object. For better insights, we also include the relative gains in comparison to a system without CoMP in Figures 6.12b and 6.13b. Even though the confidence intervals for the absolute simulation results are relatively small, the confidence intervals for the ratios are large, which is a known property of the method we apply for calculating these ratios. This effect is even amplified, because the compared values are similar in magnitude. Because the density is not directly configurable, but it is also a simulation result, these plots show confidence intervals in the x- and y-direction. Here it should be noted that the user distribution is independent of the data transmission. Therefore, the confidence intervals of the local user density for a given number of hotspots are exactly the same for all considered clustering schemes. Note that the method we apply to measure the local user density has the tendency to underestimate the density. E.g., in the case of a single hotspot all users are located within a radius of 20 m, resulting in a density of $\rho = {}^{570}\!/_{\pi 400\,\mathrm{m}^2} \approx 0.45\,\mathrm{m}^{-2}$. However, the users could also be distributed over nine squares used for density measurement, which results in a density of $\rho \approx 0.16\,\mathrm{m}^{-2}$.

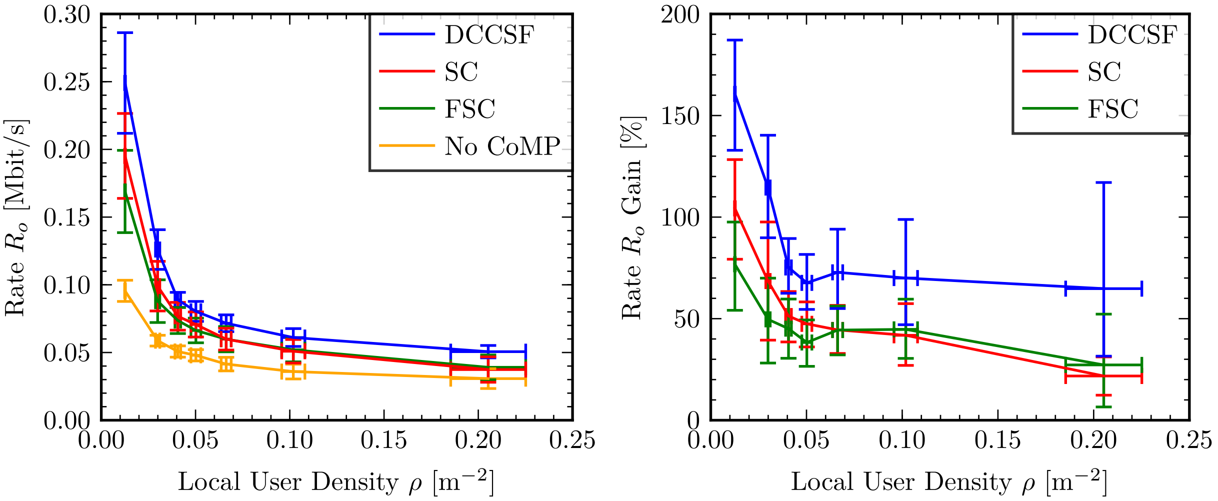
Considering the absolute carried traffic rates in Figure 6.12a, we observe that the rates decrease with increasing user density. This can be explained, because the UEs are served by a decreasing number of TPs. The same also holds for the rate per traffic object as visible in Figure 6.13a. If the relative system capacity of the systems is compared by considering the gain in relation to the system without CoMP in Figure 6.12b, it can be seen that using any clustering scheme leads to relative improvements if the user density increases. The relative gains of more than 250 % for the carried traffic rate for DCCSF seem to be high, however they can be explained by the fact that by using CoMP multiple TPs serve the UEs cooperatively. E.g., in the case of a single hotspot, in the system without CoMP all UEs are served by only a single TP in most cases. If static clustering or DCCSF with $S_{C,\max} = 3$ is applied, three or even more TPs serve the UEs, depending on the location of the hotspot. The results for both static clustering schemes are similar for higher local user densities. For densities above $0.1\,\mathrm{m}^{-2}$ FSC even outperforms SC and achieves gains of up to 170 % while the gain of SC is approximately 153 %. This can be explained that with a smaller number of hotspots the interference in the system is reduced such that coordination between sites becomes more effective.



(a) Absolute carried traffic rate

(b) Relative gains

Figure 6.12: Influence of the local user density on the system capacity



(a) Absolute rates per traffic object

(b) Relative gains

Figure 6.13: Influence of the local user density on the rate per traffic object

The relative gains of the rate per traffic object decrease for increasing local user densities, as visible in Figure 6.13b. The reason is that in a configuration with higher local user density more UEs are served by a reduced number of TPs, such that the transmission of a traffic object takes longer. While the PF scheduler makes use of the channel diversity between different UEs, which improves the system capacity, the rate for single traffic objects does not benefit if more UEs are served by the same scheduler. Nevertheless, DCCSF still achieves gains of approximately 65 % for a single hotspot. Additionally, this gain is almost constant for local user densities larger than approximately 0.05 m^{-2}, which is again an effect caused by the larger number of UEs served by one scheduler. Comparing the results for both static clustering schemes, it can be observed that their performance is similar for higher local user densities. Again FSC outperforms SC for local user densities above 0.1 m^{-2}.

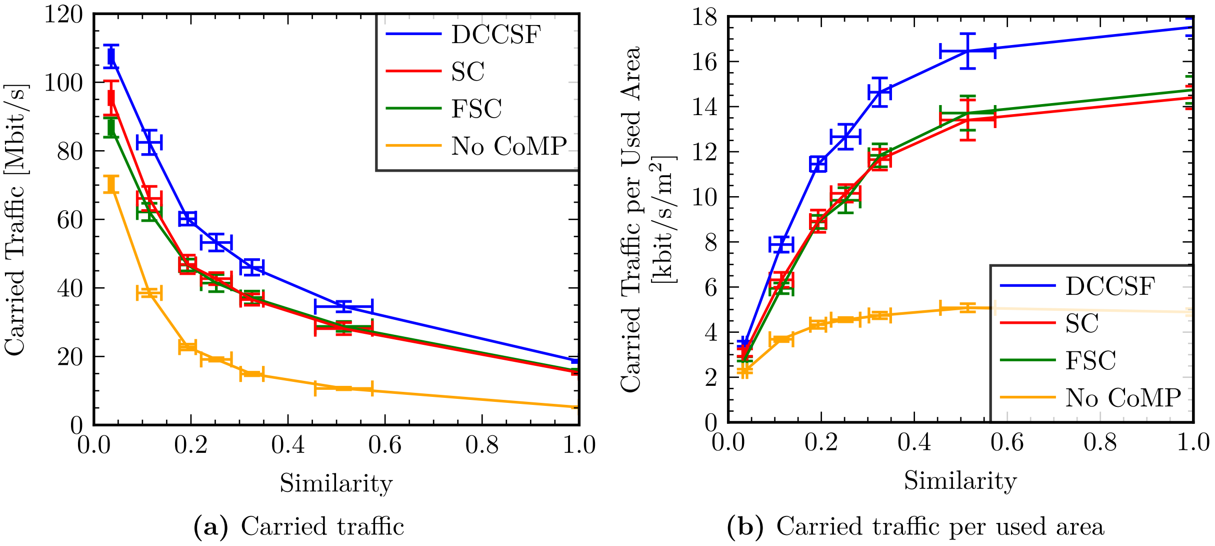
From the results, we conclude that the relative performance gains of DCCSF increase if the UEs are non-uniformly distributed. In the envisioned scenario of an urban environment users are usually located with a higher probability at certain hotspots. Also, these hotspots are not necessarily fixed, but can move over time. Therefore, the clustering should be dynamically configurable, which is provided by DCCSF.

6.2.1.2 Similarity of User Movement

We now focus on group mobility using the Virtual Track model. In this section, we follow the opposite approach than in the previous evaluation. Instead of having a fixed number of users, we now have a fixed number of 19 users per group and vary the number of groups in the scenario, which is directly related to the similarity of the mobility. If only one group is configured, the similarity is $S = 1$, because all users always move in the same direction with the same speed. For two groups the similarity is $S = 0.5$, because the users within the same group have a similarity of $S_{u_i,u_j} = 1$ and the movement of the two groups is uncorrelated, which results in a similarity of $S_{u_i,u_j} = 0$ for all users u_i from group one and all users u_j from group two. In general, the similarity is $S = {}^1\!/_{N_G}$ if N_G groups are configured. The configuration with 30 groups used in the first part of this chapter, results in a similarity of approximately 0.03. As we will see, such a small similarity is generally less beneficial for DCCSF and CoMP in general.

Figure 6.14a shows the influence of the similarity on the carried traffic. The offered traffic rate is set to 150 Mbit/s. As visible, the carried traffic rate decreases with increasing similarity. This is explained by the fact that with increasing similarity, fewer groups and therefore less UEs are in the system. As a result fewer TPs are actively serving UEs, such that the total carried rate is reduced. Note that the carried traffic rate would also decrease if the total number of users in the system would be kept constant while reducing the number of groups, because this effect is mainly caused by the reduced number of active TPs. To take this into account, Figure 6.14b shows the carried traffic rate normalized to the used area. The used area is defined similarly as the local user density, by counting the squares containing at least one UE. Thus, if less UEs are in the system, the used area is smaller. When applying this normalization, it becomes visible that in the system without CoMP, the influence of the similarity between the users on the carried traffic rate is comparatively small. On the other hand, CoMP significantly improves the carried traffic rate for increasing similarity. It should be highlighted that FSC performs even better than SC for similarities above 0.33. However, this is not a pure effect of the increased similarity, but mainly caused by less UEs in the system, such that the overall interference becomes smaller and it is more beneficial to allow cooperation between TPs located at different sites. Nevertheless, SC and FSC show comparable performances.

Figure 6.15 provides a direct comparison between the different systems. The baseline is the system without CoMP and the plot shows the relative improvements of the carried traffic rate. Here we can observe again that the gain in terms of the carried traffic increases with increasing similarity. DCCSF achieves gains of approximately 260 %, which is similar to the gain seen in the previous scenario with static user hotspots. Compared to the static clustering schemes, this is a further improvement of almost 60 percentage points.



(a) Carried traffic **(b)** Carried traffic per used area

Figure 6.14: Influence of similarity on system capacity

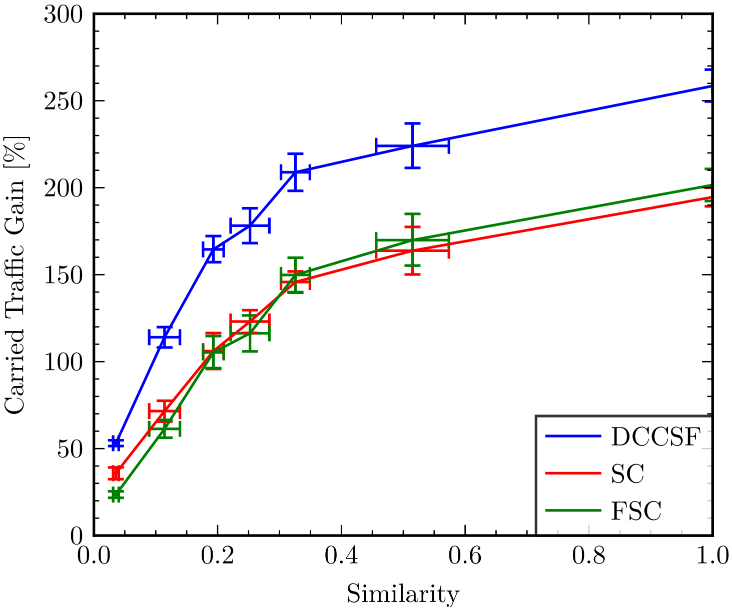
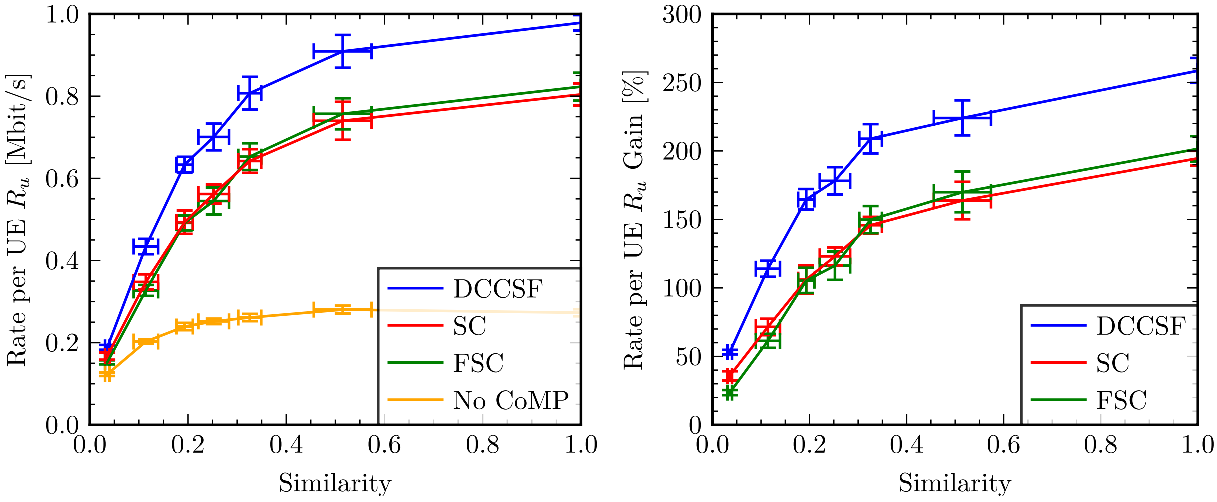


Figure 6.15: Relation between similarity and system capacity gain

Because in this evaluation the number of UEs changes, we also have a look on the average rate per UE, which is presented in Figure 6.16a. From the results it becomes evident that a higher similarity results in improved rates per UE. However, not only the higher similarity, but also the reduced interference if less TPs are transmitting increases the rate. This is the reason, why even the system without CoMP benefits from the increasing similarity. Nevertheless, CoMP leads to significant gains, while DCCSF outperforms the static clustering schemes. This is also visible when directly comparing the different systems in Figure 6.16b. DCCSF achieves gains of the rate per UE of approximately 260 % and in comparison to the static clustering schemes gains of up to 60 percentage points.

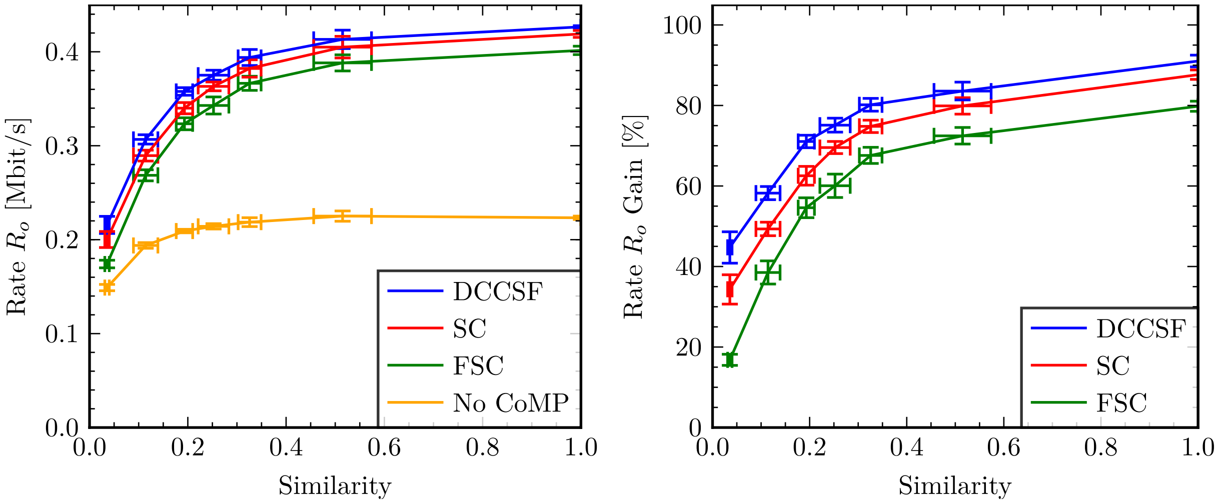
Finally, we assess the rate per traffic object. Figure 6.17a shows the absolute rates and Figure 6.17b presents the gains in comparison to a system without CoMP. As visible,



(a) Rate per UE

(b) Rate per UE gain

Figure 6.16: Influence of similarity on rate per UE



(a) Rate per traffic object

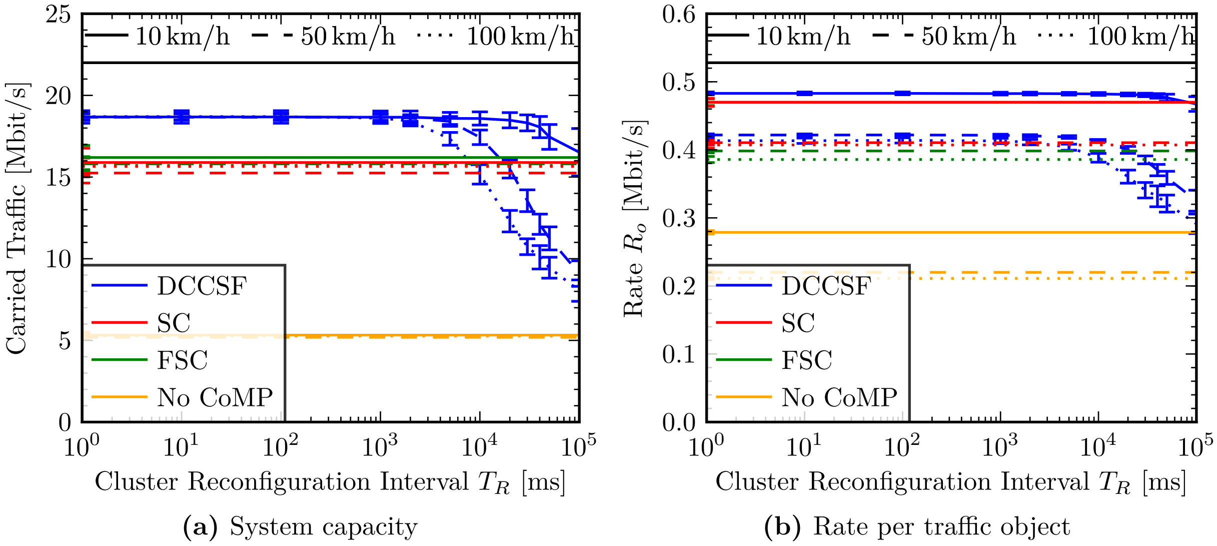
(b) Rate per traffic object gain

Figure 6.17: Influence of similarity on rate per traffic object

introducing CoMP results in smaller gains of the rate per traffic object as for the gains or the carried traffic and rate per UE. The performance of the different clustering schemes is close together, too. Nevertheless, DCCSF performs best in comparison to all other system configurations. The reason for this result is that the rate per traffic object is only partly influenced by the rate of the wireless link as explained in Section 5.2.2.3. The perceived performance of the UEs is still improved in the range of 40 % to 90 %, if DCCSF is introduced in the system.

6.2.1.3 Summary of User Mobility Behavior

To sum up the findings of the evaluations regarding group mobility, we can note that DCCSF performs especially well, if the mobility of the UEs shows a group behavior for both static user hotspots and moving groups. Stronger group behavior, i.e., higher local



(a) System capacity

(b) Rate per traffic object

Figure 6.18: Influence of cluster reconfiguration interval for one group

user densities or higher similarity, leads to increased gains in comparison to a system without CoMP. From the perspective of introducing DCCSF in real networks, this is an encouraging result, because in urban environments, users often move in groups, e.g., vehicles or users in public transportation systems. However, the expected gains of using DCCSF for pedestrian users are lower, because these show generally a less correlated mobility. As a consequence DCCSF could be primarily introduced in selected network slices, e.g., those used for vehicular users, if the network supports slicing.

6.2.2 User Movement Speed

This section deals with the influence of different user speeds on the system performance. In particular, we have a detailed look on the relation between movement speed and the cluster reconfiguration interval. To do so, we use the Virtual Track mobility model as before with a single group of 19 UEs. The system is loaded with an offered traffic rate of 50 Mbit/s. The group moves either with a speed of 10, 50 or 100 km/h. The results in terms of the system capacity and the rate per traffic object are shown in Figures 6.18a and 6.18b. As references also results for static clustering schemes as well as for a system without CoMP are included. However, in these system configurations no reconfiguration interval exists, such that the results have been expanded for easier comparison.

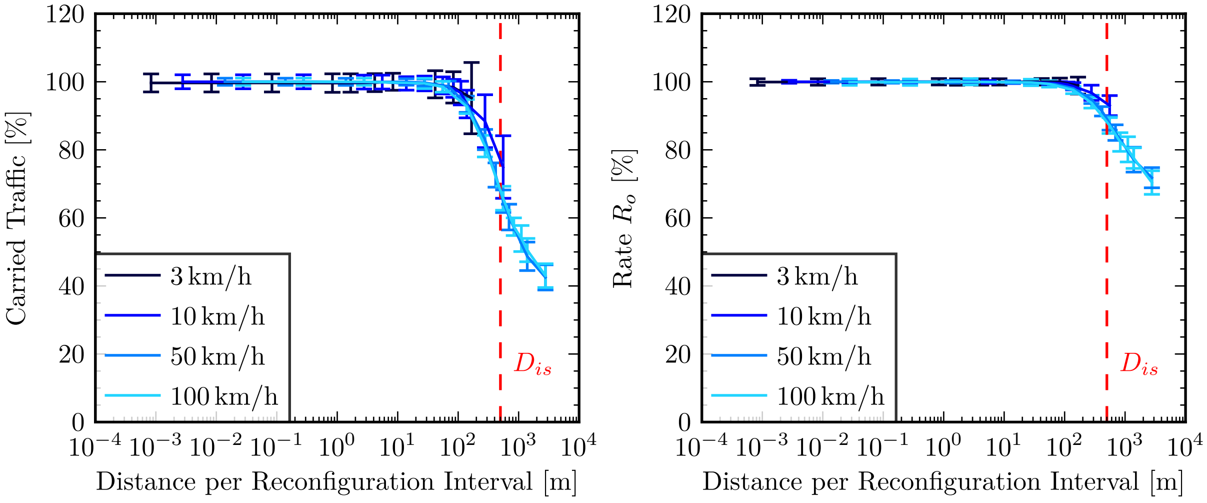
For reconfiguration intervals below 1 s, DCCSF always performs best and in comparison to the system without CoMP achieves a more than three times higher system capacity. This is in line with the results from Sections 6.2.1.1 and 6.2.1.2, because only a single group of UEs is in the system that is served without CoMP by only a single TP and in the case of CoMP by up to three TPs. Comparing the results for different speeds, we observe that the carried traffic rate is almost not affected. However, the rate per traffic object is generally smaller if the UEs move faster. The reason is that the speed determines the fast fading. The higher the speed the larger the effect of fast fading becomes. The schedulers react to the current channel conditions and schedule these UEs with a good channel, such that the overall system performance remains more or less constant. On the other hand, it takes longer to transfer a complete traffic object, because a single UE is scheduled less often if the fast fading is higher, which results in a decreased rate per traffic object.

If the cluster reconfiguration interval is increased to values above 1 s, the performance of DCCSF degrades. Nevertheless, the performance of DCCSF for speeds of 10 km/h is better than the performance of both static clustering schemes for reconfiguration intervals up to 100 s. For speeds of 50 km/h the performance of DCCSF is still better for intervals up to 20 s. Only if the speed is 100 km/h the performance of DCCSF drops below the performance of static clustering for reconfiguration intervals even smaller than approximately 10 s. However, DCCSF always guarantees a better or at least equal performance compared to a system without CoMP, because all UEs can always be served by their best TP.

As we have seen from these results, the relation between the cluster reconfiguration interval and the speed of the UEs influences the performance of DCCSF. In the following, we examine this relation in more detail. Therefore, we apply two normalizations to the simulation results. The first is to normalize the carried traffic rate and the rate per traffic object, such that the results for different speeds become directly comparable. To do so, we define the performance achieved for a reconfiguration interval of 1 ms as 100 %. The second normalization directly reflects the relation of reconfiguration interval and speed, which is achieved by showing the distance the UEs travel per reconfiguration interval on the x-axis. The results are shown in Figures 6.19 to 6.21 for 1, 15 and 30 groups of UEs. Here we include speeds from 3 km/h to 100 km/h and reconfiguration intervals from 1 ms to 200 s, which results in moved distances per reconfiguration interval ranging from less than 1 mm to approximately 2.78 km. For one group an offered load of 50 Mbit/s and for 15 and 30 groups 150 Mbit/s is used. These values are chosen such that the system always operates beyond the load limit. In the figures the dashed red line indicates the inter-site distance of $D_{is} = 500\,\mathrm{m}$.

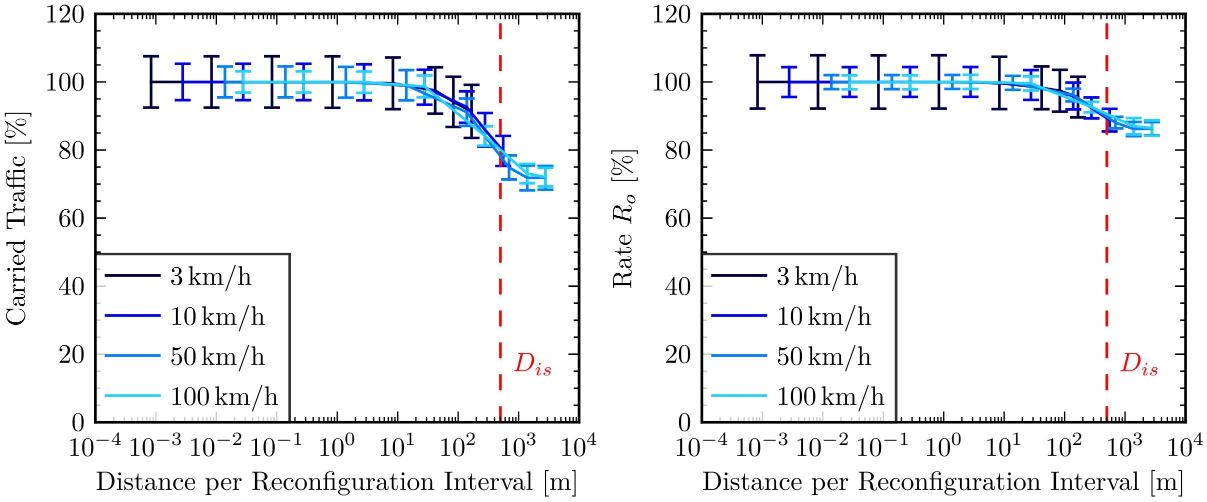
It is visible that the normalized results for different movement speeds closely overlap. This means that the performance DCCSF achieves depends on the relation between the speed of the UEs and the cluster reconfiguration interval. Performance starts to degrade if the moved distance per reconfiguration exceeds approximately 20 m to 50 m, depending on the number of groups. It is also visible that the performance degradation for increasing moved distances per reconfiguration interval is higher if fewer groups are in the system. The reason is that if more groups are available, also more clusters are generated by DCCSF. Thus, even if the clusters are not reconfigured as often, it could happen that a group moves from one area of a cluster to another cluster, such that it can be served using CoMP. This is not possible if only one group is in the system, because only one cluster is configured in this case. If the group leaves the coverage area of the cluster, the UEs are only served by a single TP without CoMP until the next cluster reconfiguration. To overcome this problem in real implementations, it would be possible to combine static with dynamic clustering. TPs that are not treated by DCCSF, because they are not reported in CMR messages, could be clustered using a static clustering scheme. Additionally, the different numbers of groups in the Virtual Track model influence the similarity between the UEs. As already seen in Section 6.2.1.2, the similarity determines the performance gain of DCCSF.

From the comparison of the inter-site distance and the moved distance per reconfiguration interval, it can be observed that the performance degrades even if the users move less than



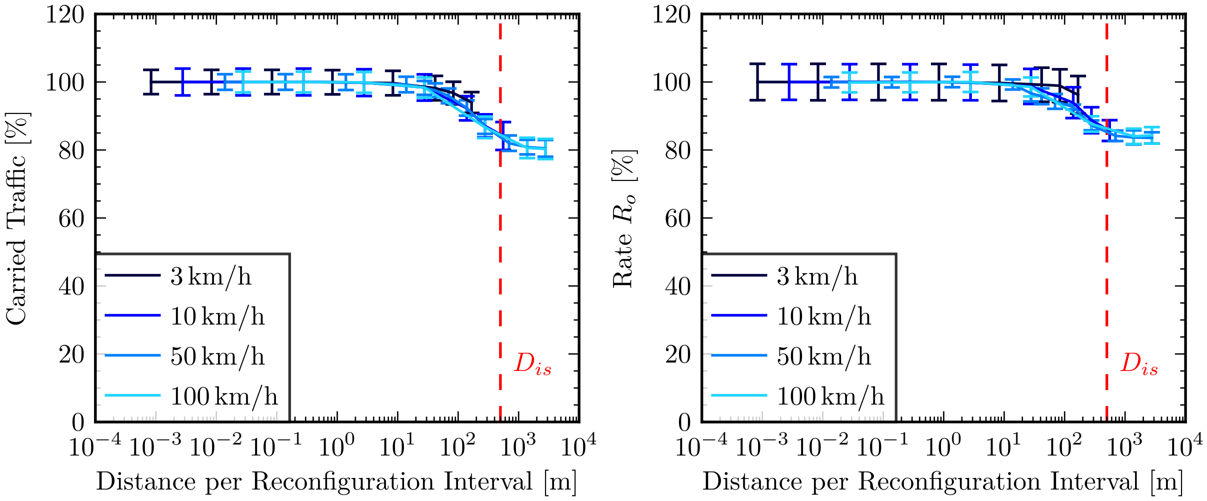
(a) System capacity **(b)** Rate per traffic object

Figure 6.19: Moved distance per reconfiguration interval, 1 group, 50 Mbit/s offered load



(a) System capacity **(b)** Rate per traffic object

Figure 6.20: Moved distance per reconfiguration interval, 15 groups, 150 Mbit/s offered load



(a) System capacity **(b)** Rate per traffic object

Figure 6.21: Moved distance per reconfiguration interval, 30 groups, 150 Mbit/s offered load

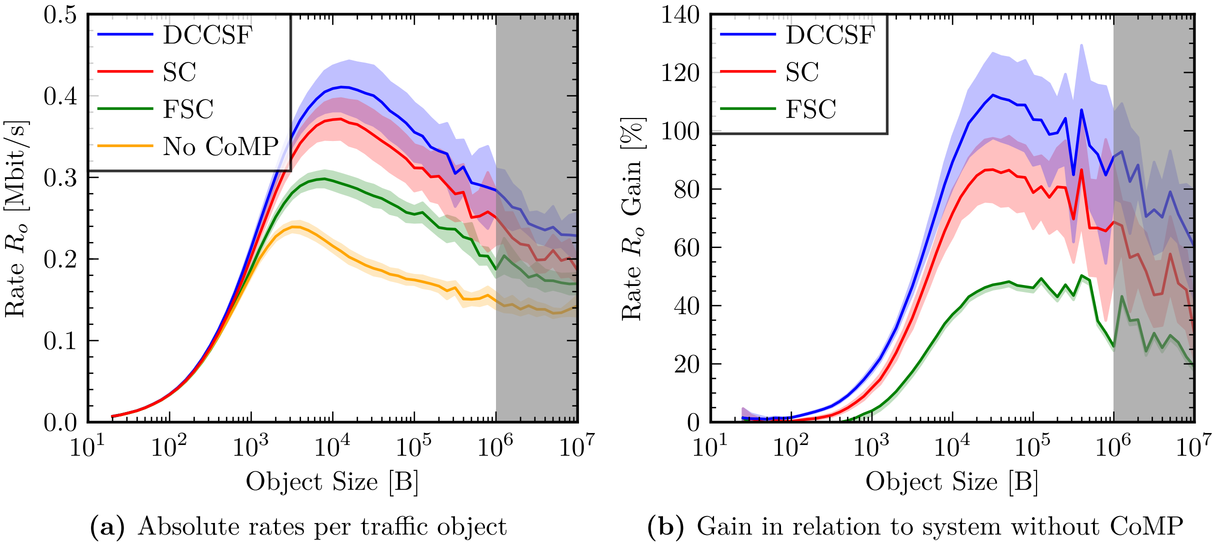
the inter-site distance per reconfiguration interval. However, in the case of 15 and 30 user groups the degradation stagnates if the users move longer distances than the inter-site distance. Therefore, we conclude that the inter-site distance influences the achievable performance, as the inter-site distance determines the physical extent of the clusters.

The results can be used in two ways. The first is to determine which reconfiguration intervals DCCSF has to support, such that performance gains are achievable for a given maximum speed of the UEs. This could be useful to select the technology on which DCCSF could be implemented. On the other hand, for a given reconfiguration interval, the maximum speed of the UEs the system supports can be derived. This is useful, if DCCSF is implemented in a 5G system with several network slices. Then it could be decided in which slices DCCSF is introduced based on the mobility characteristics of the devices in the slices and the used implementation technology.

6.2.3 Traffic Objects

In the previous evaluations, we focused on system capacity and average rates per traffic object. In this section, we have a detailed look on the relation between the user generated traffic and the performance of CoMP. Therefore, we evaluate which traffic objects benefit from CoMP in terms of the rate per traffic object. For the evaluation, the Virtual Track mobility model with 570 users, 30 groups and an offered traffic load of 150 Mbit/s is used. The users move with a constant speed of 50 km/h. The traffic objects are grouped into distinct bins and for each bin the rate per traffic object is measured. The upper bound of the bins is determined by 10^i B, where i ranges from 1 to 7 in steps of 0.1. Figure 6.22 shows the simulation results for a reconfiguration interval of $T_R = 1\,\text{ms}$. The x-axis indicates the upper bound of the bins. The y-axis in Figure 6.22a shows the rate per traffic object for the objects of the respective bin and in Figure 6.22b the gains in comparison to the system without CoMP. The 95 % confidence intervals are depicted in form of colored corridors around the mean values. The confidence intervals generally become larger for larger object sizes, because fewer objects are transmitted during the simulation and therefore less samples are available. The gray shaded area indicates the range of bins with less than 500 samples. Although this number is arbitrarily chosen, it helps to identify regions that should not be considered too closely during the interpretation of the results.

The results in Figure 6.22a show that the rates for objects smaller than approximately 1 kB are almost independent of the used CoMP scheme. The reason is that the transmission time for such small objects is mainly dominated by the core network delay. The rate reaches a maximum for objects between 4 kB to 10 kB. Larger objects generally experience a smaller rate, because the scheduler has to assign more resources until they are completely transmitted. E.g., for objects with a size of 100 kB, the transmission time is in the range of seconds, which is far above the coherence time of the channel and also due to the movement of the UE the slow fading changes. Nevertheless, when comparing the relative gains in Figure 6.22b, it becomes visible that DCCSF maintains gains of roughly 100 % for objects between 10 kB and 1 MB. In comparison to FSC this is an improvement of approximately 50 percentage points and in comparison to SC of up to 20 percentage points.



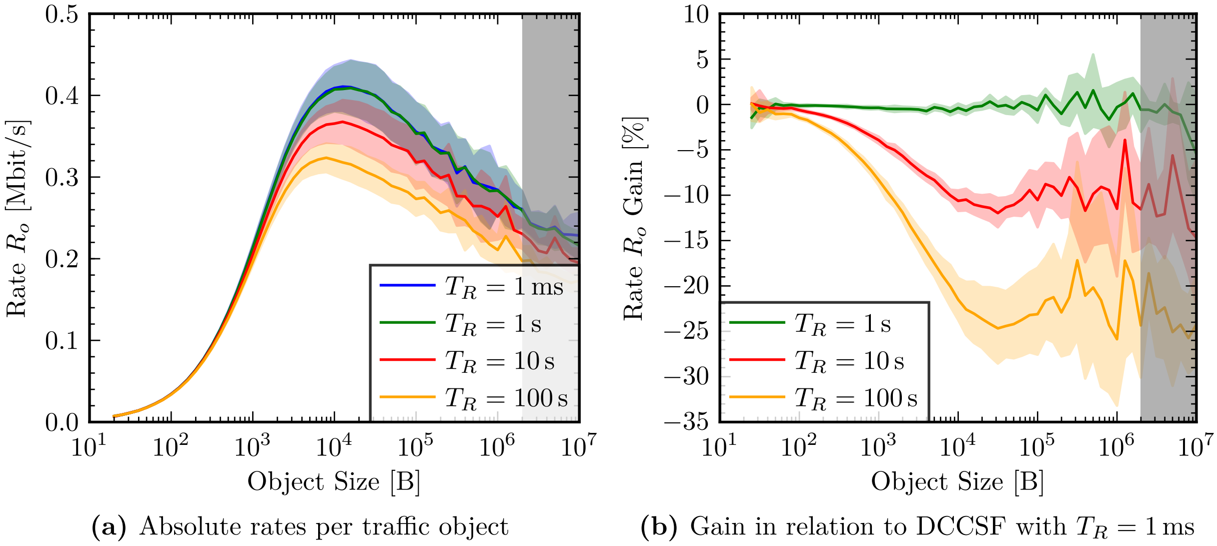
(a) Absolute rates per traffic object

(b) Gain in relation to system without CoMP

Figure 6.22: Relation between traffic object size and rate per traffic object

In Figure 6.23, we study the effect of the cluster reconfiguration interval on the rates per traffic object. As we have already seen in Section 6.2.2, larger reconfiguration intervals reduce the average rate per traffic object. Figure 6.23a shows that mainly objects larger than 1 kB are affected by the reconfiguration interval, while for smaller objects the rate remains constant. To allow an easier comparison, Figure 6.23b shows the relative difference of the rates per traffic object in comparison to DCCSF with $T_R = 1\,\text{ms}$. For a reconfiguration interval of 1 s the rate is almost the same as for a reconfiguration interval of 1 ms. This is in line with the results for the average rates per traffic object in Section 6.1.4, where the rates have not been affected by reconfiguration intervals up to 1 second. For a reconfiguration interval of 10 s the rate per traffic object is for small objects similar as for DCCSF with $T_R = 1\,\text{ms}$. However, the rate decreases by approximately 10 % for increasing object sizes up to 10 kB. For larger objects DCCSF is able to maintain a constant rate reduction despite the relatively large reconfiguration interval. The same also holds for even longer reconfiguration intervals of 100 s, where the rate per traffic object decreases by approximately 25 %. When comparing the results of DCCSF with the results for static clustering and the system without CoMP in Figure 6.22, DCCSF achieves better or similar performance for the considered reconfiguration intervals. E.g., objects with a size of 1 MB are transferred with a rate of approximately 0.15 Mbit/s in the system without CoMP, with 0.19 Mbit/s by FSC and with 0.25 Mbit/s by SC. These objects are transferred by DCCSF with a rate of 0.28 Mbit/s for $T_R = 1\,\text{s}$, 0.25 Mbit/s for $T_R = 10\,\text{s}$ and 0.21 Mbit/s for $T_R = 100\,\text{s}$. This shows that objects with a total transmission time even longer than the reconfiguration interval benefit from DCCSF. E.g., traffic objects with a size of 1 MB have a transmission time of almost 36 s.

Summarizing the findings of these evaluations, we can note that CoMP in general is best suited for traffic objects larger than 10 kB. The rate of smaller traffic objects is mainly determined by the core network delay and therefore does not profit from CoMP. Furthermore, DCCSF provides significant improvements for even larger objects, too.



(a) Absolute rates per traffic object

(b) Gain in relation to DCCSF with $T_R = 1\,\mathrm{ms}$

Figure 6.23: Influence of the cluster reconfiguration interval on the rate per traffic object

6.2.4 Summary of the Influence of Dynamic Effects

In this section, we separated the two major dynamic effects, i.e., mobility and network traffic, and evaluated their influence individually.

To regard the influence of the user distribution on the scenario independently of temporal effects, we started in Section 6.2.1.1 with static user hotspots. From the results we could conclude that the higher the imbalance of the local user density, i.e., the more users are located at hotspots, the better DCCSF performs. Similar results were obtained in Section 6.2.1.2 where we evaluated the influence of group mobility patterns on the performance gain of DCCSF. The relative gains are higher if the users move in groups. In both scenarios, DCCSF shows superior performance, because it is able to create suitable clusters for the users' locations. This also indicates that DCCSF is suited for urban scenarios where relatively static user hotspots and movement in groups occur.

Section 6.2.2 dealt with the influence of different temporal effects, mainly the relation between cluster reconfiguration interval and movement speed. By applying two normalizations, we could show that the performance DCCSF achieves depends on the distance the users move during the cluster reconfiguration interval. DCCSF maintains a constant performance for moved distances up to 50 m per reconfiguration interval. For larger distances, the performance begins to degrade, however higher performances than for the static clustering schemes are maintained for distances up to 200 m and it is always guaranteed that DCCSF achieves at least the same performance as a system without CoMP. Additionally, the performance degradation depends on the mobility pattern. A higher similarity between the users leads to high performance if the moved distance per reconfiguration interval is small, but on the other hand to a larger degradation if the moved distance becomes larger. A lower similarity results in generally lower performance, but also the degradation is lower. In the evaluations with a similarity of $S = 1$ the observed degradation is almost 60 %, while it is only 20 % for $S \approx 0.03$.

In the final Section 6.2.3, we treated several aspects of the relation between DCCSF and the network traffic. First, we had a closer look on which traffic objects profit from CoMP in general and DCCSF in particular. We showed that objects larger than 1 kB profit from CoMP and that DCCSF is able to maintain this gain also for traffic objects larger than 10 kB. By evaluation of the influence of the reconfiguration interval, we could also conclude that the relative performance of DCCSF is not affected by increased durations between cluster reconfigurations, such that the rates of traffic objects with transmission times larger than the reconfiguration interval are similar to those of smaller objects. Even if not directly evaluated, we can conclude that DCCSF is also useful for other traffic and application types that need to transmit larger amounts of data. Examples are streaming media or real-time communication, which both will play an important role in the future. An example is vehicular network traffic, where real-time communication is needed to exchange sensor and driving information between vehicles to support autonomous driving and at the same time streaming media is used for entertainment purposes.

6.3 Evaluation Summary and Implementation Recommendations

From the simulation results several recommendations how DCCSF should be implemented in real systems can be derived. We propose guidelines for the configuration of the two parameters of DCCSF, namely the maximum allowed cluster size and the cluster reconfiguration interval. We also provide recommendations in which scenarios or environments DCCSF is useful.

The results in Section 6.1.3.2 indicate that the maximum cluster size can be restricted to relatively small values, as with increasing cluster sizes the additional performance gains stagnate. A good choice is $S_{C,\max} = 3$, because this offers a reasonable trade-off between performance improvement, scheduling effort and overhead of channel measurement and signaling. This recommendation of the cluster size is independent of the user mobility behavior, as it turned out that significant performance gains are realized for different mobility models.

The best choice of the cluster reconfiguration interval depends on the scenario in which DCCSF should be applied, as the results from Sections 6.1.4 and 6.2.2 indicate. The main influences are the movement speed and the used technology to implement DCCSF. A general recommendation is to configure the cluster reconfiguration interval as short as the used technology allows. Using the results from Section 6.2.2, it is possible to determine the maximum speed for which DCCSF is beneficial for a given reconfiguration interval. If the network is organized in multiple slices, it would be an option to use DCCSF only in those slices in which the users do not move faster than the determined maximum speed. Nevertheless, we have shown that DCCSF achieves performance gains for movement speeds up to 100 km/h, if the clusters can be reconfigured every second. This covers the speeds in typical urban scenarios and could be realized using today's hardware and software technology.

The findings in Section 6.2.3 indicate that DCCSF is best suited for traffic objects larger than 10 kB. In the example of network traffic generated by vehicles, this means that not all thinkable applications profit from DCCSF. While applications resembling the web browsing paradigm benefit from increased data rates, this might not be the case for Ultra-Reliable and Low Latency Communication (URLLC) or massive Machine Type Communication (mMTC) applications. These applications generate relatively small objects and require low latencies. While DCCSF is able to handle small objects, with however limited benefits in comparison to traditional networks, it is not designed to improve the latency. Nevertheless, DCCSF provides further benefits besides increased system capacity and data rates. Because CoMP generally is able to reduce the interference, the network availability could be improved. This also helps URLLC applications, which depend on reliable communication.

7 Conclusions and Outlook

The subject of the thesis is the design and evaluation of a framework for dynamic Transmission Point (TP) clustering and scheduling to facilitate Coordinated Multi-Point (CoMP) in cellular Cloud Radio Access Networks (C-RANs). Dynamic TP clustering for CoMP is not a new problem, but is already discussed in the literature. However, because the research was mainly driven by communication engineering, several system-level aspects were not covered. Examples for effects on longer timescales are scheduling, which has been partly considered, user mobility or the network traffic on the application layer. Practical aspects of implementing dynamic clustering were often neglected in previous works. As a solution, we proposed the Dynamic Cloud Clustering and Scheduling Framework (DCCSF) and showed its system-level performance in various simulation studies. Although DCCSF is designed such that it can be introduced in any cellular network, we used Long Term Evolution (LTE) as a basis, because it is standardized and the most widely deployed cellular network technology at the time of writing. Additionally, the next generation of cellular networks, called fifth generation (5G), is expected to share many design principles with LTE, such that the necessary adaptations to introduce DCCSF in 5G networks are minimal.

In Chapter 2, we provided an overview of the architecture of LTE networks and the concept of C-RANs, which is a new paradigm for the operation and design of cellular networks. In C-RAN architectures the traditionally decentralized network functions, like signal and protocol processing, are centralized and Information Technology (IT) cloud mechanisms are used to operate the network. Additionally, we introduced different CoMP schemes and reviewed multiple publications approaching the dynamic clustering problem.

Chapter 3 treated various dynamic effects in the network, which are caused by the variation of the wireless channel, the movement of users or the behavior of the applications generating network traffic. Then we categorized these dynamic effects according to their temporal behavior ranging from milliseconds to days or even longer timescales. Also, we discussed the relations between these effects because they often cannot be treated separately. E.g., the movement of a user changes its wireless channel. In the second part of this chapter, we discussed how traditional cellular networks cope with dynamic effects. Effects on short timescales are treated by scheduling, mid-range effects are covered by the concept of Self Organizing Networks (SON) and effects on large timescales are handled by manual network planning. We concluded that dynamic clustering fits into the gap between scheduling and SON or can be seen as a part of SON.

Chapter 4 finally introduced DCCSF. After stating the design goals, we derived possibilities to realize DCCSF. Because the aspect of a direct applicability of DCCSF in real systems was an important design goal, DCCSF is realized as a heuristic algorithm. Consequently, the found clustering is not necessarily optimal, but it is determined in short time. Another design goal was that the necessary extensions to the standardized parts of the cellular network are kept small. However, the standards do not comprise the C-RAN cloud environment, such that we were relatively free to change this part of the network. To integrate DCCSF into an existing system, it is necessary to deploy a new component, the Cluster Manager (CM), in the C-RAN environment. The CM is responsible to execute the algorithm determining the clustering. Besides the CM, DCCSF relies on Cluster Entities (CEs), which are the representation of a cluster, i.e., User Equipments (UEs) and TPs, in the C-RAN. Each CE includes a scheduler, the computing hardware performing signal and protocol processing and information about the UEs within the cluster in terms of UE states. The UEs transmit Cluster Measurement Report (CMR) messages to the CM, to provide it with the current system state, which is used to generate the clustering.

In direct comparison with other dynamic clustering algorithms in Section 4.6.3, we have seen that their principles for defining the clustering share common properties with DCCSF. However, the concept of partitionings is a unique feature of DCCSF. The partitionings are used by the schedulers to evaluate reasonable scheduling options only. In contrast to exhaustive search, the scheduling effort is significantly reduced as we have shown theoretically in Section 4.6.2 and in simulations in Section 6.1.5. DCCSF provides an interface to the schedulers but is not restricted to a specific scheduling algorithm. In Chapter 5, we have exemplary shown the integration of a scheduler following the Proportional Fair (PF) principle. Additionally, DCCSF is not limited to a specific precoding method, which is another distinction from other dynamic clustering schemes.

In Chapter 5, we introduced the simulation model used for the performance evaluation in Chapter 6. The model includes the LTE cellular network, a network traffic model, user mobility models and a method to abstract the performance of CoMP from real transmission techniques. The performance abstraction generally overestimates the inter-cluster interference and underestimates the signal power, such that the overall performance is smaller in comparison to the achievable performance using real transmission techniques.

From the evaluation results, we could conclude that introducing DCCSF significantly improves the network performance. Especially in scenarios resembling urban environments, e.g., in scenarios with inhomogeneous user densities or group mobility patterns, DCCSF proved to be beneficial. Applying DCCSF increases the total system capacity as well as the perceived performance from the user's perspective. We have shown that the necessary cluster sizes are relatively small and cluster reconfigurations can be performed relatively seldom in comparison to the scheduling interval of 1 ms. In the evaluated scenarios cluster sizes larger than three TPs provide only marginal performance improvements and for movement speeds of up to 100 km/h it is sufficient to reconfigure the clusters every second to achieve full performance. Thus, the overhead and additional complexity of CoMP can be kept small. DCCSF does not require any changes on the physical layer hardware, i.e., antennas, sites or power amplifiers. The changes are purely logical and have to be performed only within the central office of the C-RAN. Thereby, the Capital Expenses (Capex) and effort to introduce DCCSF in existing networks are small. The Operating Expenses (Opex) are similar as for a C-RAN system without DCCSF.

During the design of DCCSF, we simplified or neglected multiple aspects and challenges, which could be subject of further studies. In the following, we briefly mention them and provide directions, how they could be solved.

The design and evaluation of DCCSF focused on the downlink direction only. However, the uplink is important, too, and becomes more and more relevant as future applications, like real-time video communication or sensor data fusion for autonomous vehicles, need to transmit large amounts of data in the uplink direction. Because uplink CoMP schemes generally need less assistance from the UEs and often could be implemented completely on the network side, adapting DCCSF for uplink transmissions should be possible without larger changes of the overall concept.

One of the configuration parameters of DCCSF is the constant reconfiguration interval. Thus, clusters are reconfigured after certain intervals, even if this is not required because the network conditions have not changed. One possibility to improve DCCSF would be to trigger cluster reconfigurations by changing network conditions. This could be e.g., achieved if the UEs only send a CMR message if the measured channel qualities change by certain thresholds. Then the dynamic clustering algorithm could be executed when a certain number of CMR messages have been received by the CM. Another option would be to proactively adapt the clustering using predictions of the future network conditions. Additionally, a static clustering scheme could be used as a fallback option, in case that certain reconfiguration interval thresholds are exceeded. This could be the case if the network is overloaded and the calculation of a dynamic clustering takes too much time.

As we have stated in Section 4.2.3, during the reconfiguration of the clusters it is necessary to assign a sufficient amount of processing capacity to the CEs. The amount of processing capacity is influenced by factors like the cluster size, the used Modulation and Coding Scheme (MCS) to serve the UEs, the number of transmit antennas and the usable radio bandwidth [Des+12]. Some aspects are directly known during the cluster reconfiguration, like the cluster size and the number of transmit antennas. Others like the used MCS have to be predicted, e.g., based on previous measurements. Even if the dimensioning of computing resources is not sufficient, it is still possible to operate the network as we have shown in previous publications [Wer+15; SG16]. However, this comes at the cost of reduced system performance. It would be interesting to extend DCCSF by a prediction and allocation of processing capacity.

This brings up the next aspect, because we only outlined the system architecture necessary for DCCSF, but did not discuss any real implementation options. As we have seen in the evaluation in Section 6.2.2, DCCSF depends on cluster reconfiguration intervals in the range of seconds to achieve performance gains. However, the actual cluster reconfiguration may only use a smaller fraction of the total interval duration, because during these intervals also other tasks like channel measurements or the transmission of CMR messages have to be performed. To realize DCCSF it would be necessary to evaluate which cloud technologies support the requirements of DCCSF.

For the evaluation, we used a simplified abstraction from CoMP transmissions, which is only based on transmit power and signal attenuation. Here it would be interesting to see how the results change if more realistic transmission techniques are included. Our abstraction generally overestimates inter-cluster interference, such that it could be expected that using a real precoding scheme would result in higher performance gains for DCCSF. However, this would require a detailed channel model, such that the overall simulation effort would increase. In the evaluation, we combined a regular cell layout with a randomly generated street topology. If a more detailed channel model is implemented, this could also be used to model irregular cell layouts or to include smaller cells in the simulation. We also assumed ideal channel knowledge, which is not available in real systems. One extension of the model could be to introduce imperfect Channel State Information (CSI) and to evaluate how DCCSF can cope with it. Another extension would be to introduce more receive antennas at the UEs to allow Single-User MIMO (SU-MIMO) and receiver-side signal processing. This introduces the trade-off between improving the data rate of a single UE by using SU-MIMO or to transmit data to multiple UEs using Multi-User MIMO (MU-MIMO).

Also, the data traffic model only covers a small fraction of the traffic share occurring in future cellular networks. Here it would be interesting to improve the model and include other applications. Additionally, the evaluation of dynamics arising from effects of the transport layer, especially from the Transmission Control Protocol (TCP), would be a worthwhile study.

Finally, it would be interesting to see how DCCSF performs in a real-world network. Therefore, a reasonable large test deployment would be required to capture the effects of user mobility and provide enough degrees of freedom to define the TPs clusters.

A Mathematical Derivations

A.1 Number of Clusters

A.1.1 Limited Cluster Size

For a limited cluster size of $S_{C,\max}$, the total number of clusters is derived by summing up the number of clusters for a given cluster size. The number of clusters for a given size is determined by the binomial coefficient. Therefore, the total number of clusters N_C is:

$$N_C = \sum_{s=1}^{S_{C,\max}} \binom{N_{\mathrm{TP}}}{s} \tag{A.1}$$

From this equation, the special case of unlimited cluster size can be derived by setting $S_{C,\max} = N_{\mathrm{TP}}$.

A.1.2 Unlimited Cluster Size

If the cluster size is only limited by the number of available TPs, the number of possible clusters is given by Equation (A.1) with $S_{C,\max} = N_{\mathrm{TP}}$:

$$N_C = \sum_{s=1}^{N_{\mathrm{TP}}} \binom{N_{\mathrm{TP}}}{s} = 2^{N_{\mathrm{TP}}} - 1 \tag{A.2}$$

Table A.1 shows examples for small values of N_{TP}. The number of clusters N_C is calculated according to eq. (A.2). Here the TPs are enumerated starting from 1.

N_{TP}	N_C	Clusters
1	1	[1]
2	3	[1], [2], [1, 2]
3	7	[1], [2], [3], [1, 2], [1, 3], [2, 3], [1, 2, 3]
4	15	[1], [2], [3], [4], [1, 2], [1, 3], [1, 4], [2, 3], [2, 4], [3, 4], [1, 2, 3], [1, 2, 4], [1, 3, 4], [2, 3, 4], [1, 2, 3, 4]
...		

Table A.1: Examples of clusters

A.2 Number of Scheduling Options

A.2.1 Number of Partitionings

The number of scheduling options per cluster depends on the cluster size, the number of UEs served by the cluster and the number of transmit antennas. To calculate the number of scheduling options, it is first necessary to determine how many partitionings are available for the cluster. The total number of partitionings is given by the series of Bell numbers:

$$B_{S_C} = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^{S_C}}{k!} \tag{A.3}$$

The Bell numbers for $S_C = 0, 1, \ldots$ are:

$$1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, \ldots \tag{A.4}$$

So for a cluster with $S_C = 2$ two partitionings exist, for a cluster with $S_C = 3$ five partitionings exist, and so on. For comparatively small cluster sizes it is possible to enumerate the partitionings completely.

A.2.2 Number of Scheduling Options

For a given partitioning and number of UEs, the number of scheduling options is obtained by the multinomial coefficient:

$$\binom{n}{k_1, k_2, \ldots, k_m} \tag{A.5}$$

With $n = \sum_{i=1}^{m} k_i$. To obtain the scheduling options, m equals the number of partitions in the partitioning and k_i is set to the size of partitioning i multiplied by the number of transmit antennas per TP. As the number of UEs is in general not equal to n as defined before, an additional step to select exactly n UEs out of the N_{UE} available UEs. n is thereby also equal to $N_{\text{TP}} \cdot N_{\text{TX}}$. Therefore, the number of scheduling options for partitioning $\mathcal{P}$ $N^{\text{exhaustive}}_{\text{options},\mathcal{P}}$ becomes:

$$N^{\text{exhaustive}}_{\text{options},\mathcal{P}} = \binom{N_{\text{UE}}}{N_{\text{TP}} \cdot N_{\text{TX}}} \binom{N_{\text{TP}} \cdot N_{\text{TX}}}{k_1, k_2, \ldots, k_m} \tag{A.6}$$

The total number of scheduling options is obtained by summing up over all partitionings of the cluster:

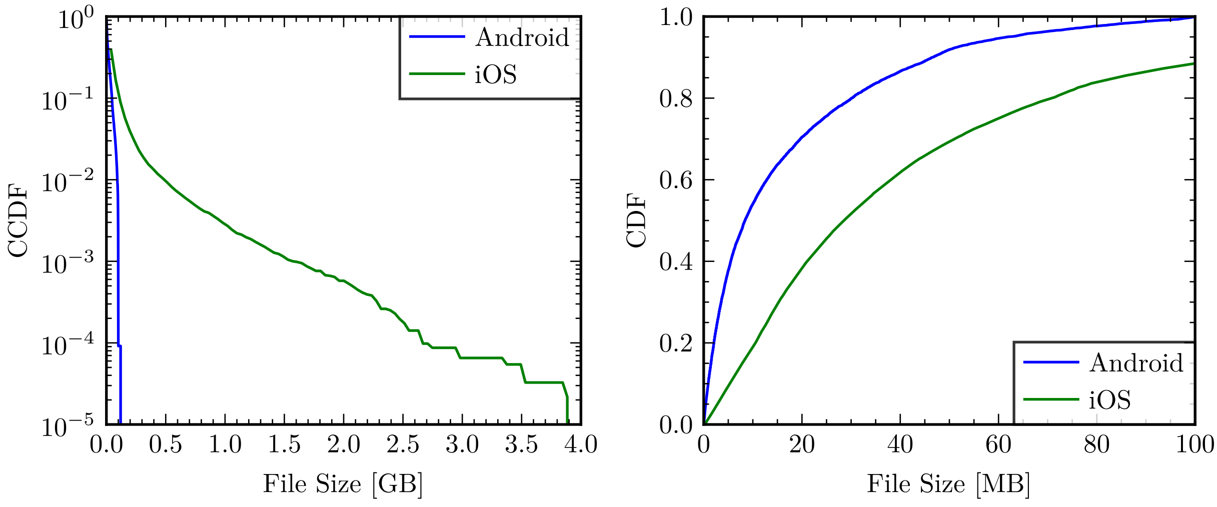
$$N^{\text{exhaustive}}_{\text{options}} = \sum_{\mathcal{P} \in \mathcal{S}_{\mathcal{P}}} N^{\text{exhaustive}}_{\text{options},\mathcal{P}} \tag{A.7}$$

B Mobile App Sizes

The iOS AppStore allows accessing app metadata by using a publicly available Application Programming Interface (API)[1]. We used this interface to extract metadata from 91 792 randomly selected apps. Although this number is relatively small in comparison to the stated 2 200 000 available apps, the sample size is sufficient to evaluate general app properties.

The Google Play Store[2] does not provide a convenient API to access metadata of Android apps. However, the size information can directly be obtained on the web version of the app store. We extracted size information of 10 919 randomly selected apps. From these apps, we derived the following statistics of the application file sizes.

We derived an average size of 55.26 MB and a median size of 28.54 MB for Apple iOS apps and an average size of 17.08 MB and a median of 8.4 MB for Android apps. The measured sizes range from 0.02 MB to 3924.44 MB for iOS and from 0.01 MB to 120.9 MB for Android. The smaller apps are mainly utilities or helper apps, while games account for the larger sizes. Figure B.1 presents the distributions for both app stores. Figure B.1a shows the Complementary Cumulative Distribution Function (CCDF) of all measured sizes. As visible, the probability for large app sizes is relatively small. Figure B.1b provides a more detailed insight in the Cumulative Distribution Function (CDF) of the app sizes up to 100 MB, which covers almost all Android apps and almost 90 % of the iOS apps.



(a) CCDF

(b) CDF

Figure B.1: App file size distribution

[1] https://affiliate.itunes.apple.com/resources/documentation/itunes-store-web-service-search-api visited on December 6, 2017

[2] https://play.google.com/store visited on December 20, 2017

C Speed of Vehicles

We use the Google directions API[1] to gather information about average speeds in several of the largest German cities. In total 13 985 randomly selected routes in 35 cities are evaluated. The Google directions API allows to obtain the travel time on a route by two ways. The first only considers speed limits and the second additionally considers traffic conditions caused by congestion. To cover the impact of rush hour traffic, the travel time was obtained on a weekday morning at 8am.

The results reveal an average speed of 24.96 km/h without considering traffic and an average speed of 23.98 km/h if considering traffic conditions. The slightly lower speeds if considering traffic influences are caused by heavy traffic during the morning rush hour. Figure C.1 shows the CDF of the speed on all gathered routes. As visible, the impact of the traffic conditions is highest for speeds around 40 km/h.

Figure C.2a presents the CDF of the average speeds of all routes per city with and without the influence of the traffic conditions. Due to the limited number of considered cities, the distribution shows discrete steps. Figure C.2b shows the detailed distribution of the speed on the individual roads of all routes. Here the influence of the traffic conditions cannot be shown, because it is not provided by the API. In total 129 145 roads are contained in the used sample. The figure reveals that all types of roads are contained in the randomly chosen routes, because speeds from close to zero up to almost 120 km/h are observed. These speeds could be achieved on living streets (traffic-calming areas) and inner-city highways. However, the major fraction of the streets of approximately 95 % has speeds below 50 km/h.

[1] `https://developers.google.com/maps/documentation/directions/` visited on December 12, 2017

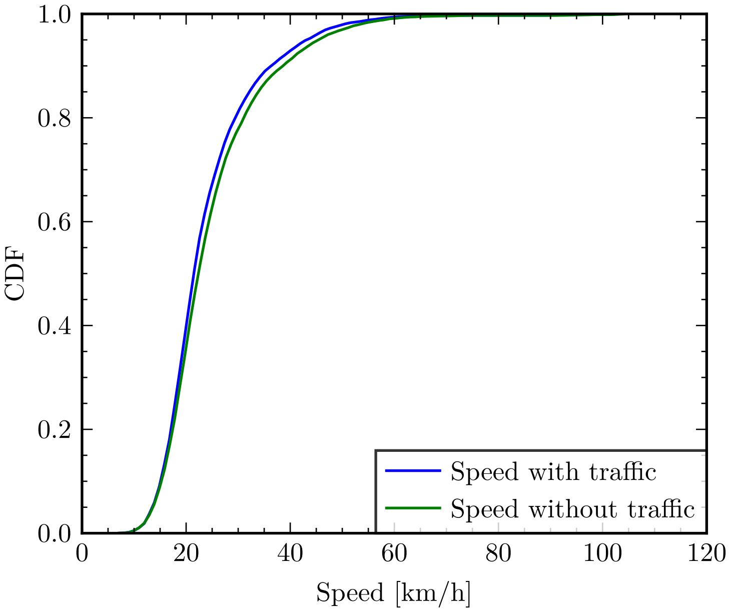
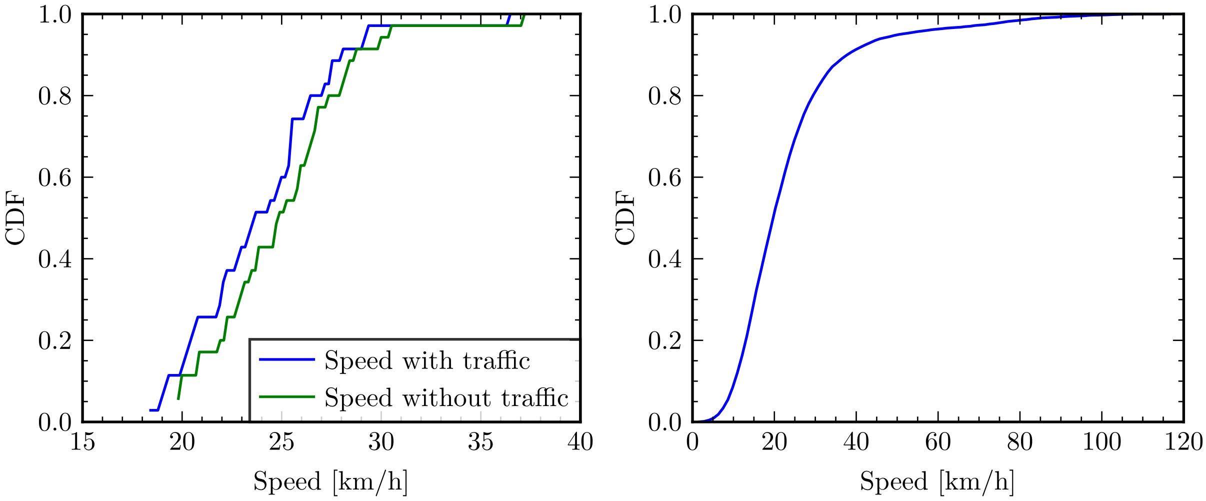


Figure C.1: Distribution of the speed per route



(a) Distribution of the average speed per city **(b)** Distribution of the speed on individual roads

Figure C.2: Speed distributions

Bibliography

[3GPP 23.203] 3GPP WG S2. *Policy and charging control architecture*. TS 23.203 ver. 13.10.0. 3GPP.

[3GPP 25.814] 3GPP WG R1. *Physical layer aspect for evolved Universal Terrestrial Radio Access (UTRA)*. TR 25.814 ver. 7.1.0. 3GPP.

[3GPP 36.104] 3GPP WG R4. *Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception*. TS 36.104 ver. 13.8.0. 3GPP.

[3GPP 36.211] 3GPP WG R1. *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation*. TS 36.211 ver. 13.6.0. 3GPP.

[3GPP 36.213] 3GPP WG R1. *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures*. TS 36.213 ver. 13.6.0. 3GPP.

[3GPP 36.300a] 3GPP WG R2. *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2*. TS 36.300 ver. 13.8.0. 3GPP.

[3GPP 36.300b] 3GPP WG R2. *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2*. TS 36.300 ver. 10.12.0. 3GPP.

[3GPP 36.304] 3GPP WG R2. *Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode*. TS 36.304 ver. 13.6.0. 3GPP.

[3GPP 36.331] 3GPP WG R2. *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification*. TS 36.331 ver. 13.6.0. 3GPP.

[3GPP 36.355] 3GPP WG R2. *Evolved Universal Terrestrial Radio Access (E-UTRA); LTE Positioning Protocol (LPP)*. TS 36.355 ver. 13.3.0. 3GPP.

[3GPP 36.423a] 3GPP WG R3. *Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 Application Protocol (X2AP)*. TS 36.423 ver. 13.7.0. 3GPP.

[3GPP 36.423b] 3GPP WG R3. *Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 Application Protocol (X2AP)*. TS 36.423 ver. 12.9.0. 3GPP.

[3GPP 36.741] 3GPP WG R1. *Study on further enhancements to Coordinated Multi-Point (CoMP) operation for LTE*. TR 36.741 ver. 14.0.0. 3GPP.

[3GPP 36.742] 3GPP WG R3. *Study on Self-Organizing Networks (SON) for enhanced Coordinated Multi-Point (eCoMP)*. TR 36.742. 3GPP.

[3GPP 36.814] 3GPP WG R1. *Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects*. TR 36.814 ver. 9.2.0. 3GPP.

[3GPP 36.819] 3GPP WG R1. *Coordinated multi-point operation for LTE physical layer aspects*. TR 36.819 ver. 11.2.0. 3GPP.

[3GPP 36.913] 3GPP WG RP. *Requirements for further advancements for Evolved Universal Terrestrial Radio Access (E-UTRA) (LTE-Advanced)*. TR 36.913 ver. 14.0.0. 3GPP.

[3GPP2] *cdma2000 Evaluation Methodology*. Tech. rep. A. 3rd Generation Partnership Projekt 2, May 2009.

[AC14] M. Artuso and H. Christiansen. “Discrete-event simulation of coordinated multi-point joint transmission in LTE-Advanced with constrained backhaul.” In: *Wireless Communications Systems (ISWCS), 2014 11th International Symposium on*. Aug. 2014. DOI: 10.1109/ISWCS.2014.6933329.

[Afa+10] A. Afanasyev et al. “Host-to-Host Congestion Control for TCP.” In: *IEEE Communications Surveys Tutorials* 12.3 (2010). DOI: 10.1109/SURV.2010.042710.00114.

[Ahm+16] A. M. Ahmadian et al. “Low complexity Moore-Penrose inverse for large CoMP areas with sparse massive MIMO channel matrices.” In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Sept. 2016. DOI: 10.1109/PIMRC.2016.7794773.

[Ali+13] O. G. Aliu et al. “A Survey of Self Organisation in Future Cellular Networks.” In: *IEEE Communications Surveys Tutorials* 15.1 (2013). DOI: 10.1109/SURV.2012.021312.00116.

[Aly+15] Islam Alyafawi et al. “Critical issues of centralized and cloudified LTE-FDD radio access networks.” In: *ICC 2015, IEEE International Conference on Communications, 8-12 June 2015, London, United Kingdom*. London, June 2015. DOI: 10.1109/ICC.2015.7249202. URL: http://www.eurecom.fr/publication/4528.

[And+01] M. Andrews et al. “Providing quality of service over a shared wireless link.” In: *IEEE Communications Magazine* 39.2 (Feb. 2001). DOI: 10.1109/35.900644.

[AS12] S. S. Ali and N. Saxena. “A novel static clustering approach for CoMP.” In: *2012 7th International Conference on Computing and Convergence Technology (ICCCT)*. Dec. 2012.

[Bai+00] P. W. Baier et al. “Joint transmission (JT), an alternative rationale for the downlink of time division CDMA using multi-element transmit antennas.” In: *2000 IEEE Sixth International Symposium on Spread Spectrum Techniques and Applications. ISSTA 2000. Proceedings (Cat. No.00TH8536)*. Vol. 1. Sept. 2000. DOI: 10.1109/ISSSTA.2000.878069.

[Bar+13] J. Barrachina et al. “V2X-d: A vehicular density estimation system that combines V2V and V2I communications.” In: *2013 IFIP Wireless Days (WD)*. Nov. 2013. DOI: 10.1109/WD.2013.6686518.

[Bas+16] S. Bassoy et al. "Load Aware Self-Organising User-Centric Dynamic CoMP Clustering for 5G Networks." In: *IEEE Access* 4 (2016). DOI: 10.1109/ACCESS.2016.2569824.

[Bas+17] S. Bassoy et al. "Coordinated Multi-Point Clustering Schemes: A Survey." In: *IEEE Communications Surveys Tutorials* 19.2 (2017). DOI: 10.1109/COMST.2017.2662212.

[Bat+10] R.L. Batista et al. "Performance Evaluation for Resource Allocation Algorithms in CoMP Systems." In: *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd.* Sept. 2010. DOI: 10.1109/VETECF.2010.5594241.

[BBB12] P. Baracca, F. Boccardi, and V. Braun. "A dynamic joint clustering scheduling algorithm for downlink CoMP systems with limited CSI." In: *Wireless Communication Systems (ISWCS), 2012 International Symposium on.* Aug. 2012. DOI: 10.1109/ISWCS.2012.6328484.

[BBB14] Paolo Baracca, Federico Boccardi, and Nevio Benvenuto. "A dynamic clustering algorithm for downlink CoMP systems with multiple antenna UEs." In: *EURASIP Journal on Wireless Communications and Networking* 2014.1 (2014). DOI: 10.1186/1687-1499-2014-125.

[Bec+11] R. A. Becker et al. "A Tale of One City: Using Cellular Network Data for Urban Planning." In: *IEEE Pervasive Computing* 10.4 (Apr. 2011). DOI: 10.1109/MPRV.2011.44.

[BH06] F. Boccardi and H. Huang. "Zero-Forcing Precoding for the MIMO Broadcast Channel under Per-Antenna Power Constraints." In: *2006 IEEE 7th Workshop on Signal Processing Advances in Wireless Communications.* July 2006. DOI: 10.1109/SPAWC.2006.346354.

[BH07] F. Boccardi and H. Huang. "Limited Downlink Network Coordination in Cellular Networks." In: *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications.* Sept. 2007. DOI: 10.1109/PIMRC.2007.4394807.

[Bjö+13] E. Björnson et al. "Receive Combining vs. Multi-Stream Multiplexing in Downlink Systems With Multi-Antenna Users." In: *IEEE Transactions on Signal Processing* 61.13 (July 2013). DOI: 10.1109/TSP.2013.2260331.

[BMB16] R. Brandt, R. Mochaourab, and M. Bengtsson. "Distributed Long-Term Base Station Clustering in Cellular Networks Using Coalition Formation." In: *IEEE Transactions on Signal and Information Processing over Networks* 2.3 (Sept. 2016). DOI: 10.1109/TSIPN.2016.2548781.

[BNetzA16] *Übersicht Mobilfunkspektrum nach der Auktion - Zuordnung ab 01.01.2017 gültig.* Bundesnetzagentur, Jan. 14, 2016. URL: https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/OffentlicheNetze/Mobilfunk/DrahtloserNetzzugang/Projekt2016/Frequenzen700bis1800_pdf.pdf (visited on Apr. 19, 2017).

[BO16] Anthony Beylerian and Tomoaki Ohtsuki. “Multi-point fairness in resource allocation for C-RAN downlink CoMP transmission.” In: *EURASIP Journal on Wireless Communications and Networking* 2016.1 (2016). DOI: 10.1186/s13638-015-0501-4.

[BRF14] N. Becker, A. Rizk, and M. Fidler. “A measurement study on the application-level performance of LTE.” In: *2014 IFIP Networking Conference*. June 2014. DOI: 10.1109/IFIPNetworking.2014.6857113.

[BSH03] F. Bai, Narayanan Sadagopan, and A. Helmy. “IMPORTANT: a framework to systematically analyze the Impact of Mobility on Performance of Routing Protocols for Adhoc Networks.” In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*. Vol. 2. Mar. 2003. DOI: 10.1109/INFCOM.2003.1208920.

[Cas+14] P. Casas et al. “YouTube in the move: Understanding the performance of YouTube in cellular networks.” In: *2014 IFIP Wireless Days (WD)*. Nov. 2014. DOI: 10.1109/WD.2014.7020798.

[Che+15] A. Checko et al. “Cloud RAN for Mobile Networks - A Technology Overview.” In: *Communications Surveys Tutorials, IEEE* 17.1 (2015). DOI: 10.1109/COMST.2014.2355255.

[CHY16] Z. Chen, X. Hou, and C. Yang. “Training Resource Allocation for User-Centric Base Station Cooperation Networks.” In: *IEEE Transactions on Vehicular Technology* 65.4 (Apr. 2016). DOI: 10.1109/TVT.2015.2420114.

[ÇLG16] Serdar Çolak, Antonio Lima, and Marta C. González. “Understanding congested travel in urban areas.” In: 7 (Mar. 2016). DOI: 10.1038/ncomms10793.

[CMRI11] China Mobile Research Institute. *C-RAN The Road Towards Green RAN*. Tech. rep. v2.5. 2011. URL: %5Curl%7Bhttp://labs.chinamobile.com/report/view%5C_59826%7D.

[Cos83] M. Costa. “Writing on dirty paper (Corresp.)” In: *IEEE Transactions on Information Theory* 29.3 (May 1983). DOI: 10.1109/TIT.1983.1056659.

[COST231] *COST 231 Final Report: Digital Mobile Radio Towards Future Generation Systems*. Tech. rep. Office for Official Publications of the European Communities, 1999.

[Cox14] Christopher Cox. *An introduction to LTE: LTE, LTE-advanced, SAE, VoLTE and 4G mobile communications*. 2nd Ed. Chichester: Wiley-VCH, 2014. ISBN: 978-1-11-881803-9. URL: http://swbplus.bsz-bw.de/bsz406539146cov.htm.

[CPR15] CPRI. *Common Public Radio Interface (CPRI); Interface Specification*. Tech. rep. Oct. 9, 2015. URL: http://www.cpri.info/downloads/CPRI_v_7_0_2015-10-09.pdf.

[CSL16] Junsu Choi, Illsoo Sohn, and Kwang Bok Lee. “Adaptive remote radio head control for cloud radio access networks.” In: *EURASIP Journal on Wireless Communications and Networking* 2016.1 (2016). DOI: `10.1186/s13638-016-0654-9`.

[CWB08] Emmanuel J. Candès, Michael B. Wakin, and Stephen P. Boyd. “Enhancing Sparsity by Reweighted 1 Minimization.” In: *Journal of Fourier Analysis and Applications* 14.5 (2008). DOI: `10.1007/s00041-008-9045-x`. URL: `http://dx.doi.org/10.1007/s00041-008-9045-x`.

[DA04] Jianhe Du and Lisa Aultman-Hall. “An investigation of the distribution of driving speeds using in-vehicle GPS data.” In: *Vermont Institute of Transportation Engineers Annual Meeting*. 2004.

[Dav+13] A. Davydov et al. “Evaluation of Joint Transmission CoMP in C-RAN based LTE-A HetNets with large coordination areas.” In: *Globecom Workshops (GC Wkshps), 2013 IEEE*. Dec. 2013. DOI: `10.1109/GLOCOMW.2013.6825087`.

[DBC93] P. Dent, G.E. Bottomley, and T. Croft. “Jakes fading model revisited.” In: *Electronics Letters* 29.13 (June 1993). DOI: `10.1049/el:19930777`.

[Des+12] C. Desset et al. “Flexible power modeling of LTE base stations.” In: *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*. Apr. 2012. DOI: `10.1109/WCNC.2012.6214289`.

[DPS16] Erik Dahlman, Stefan Parkvall, and Johan Skld. *4G: LTE-Advanced Pro and The Road to 5G*. Third. 2016.

[Dup+11] Jonathan Duplicy et al. “MU-MIMO in LTE Systems.” In: *EURASIP Journal on Wireless Communications and Networking* 2011.1 (2011). DOI: `10.1155/2011/496763`.

[DVR03] Suman Das, Harish Viswanathan, and G. Rittenhouse. “Dynamic load balancing through coordinated scheduling in packet data systems.” In: *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*. Vol. 1. Mar. 2003. DOI: `10.1109/INFCOM.2003.1208728`.

[DY14] Binbin Dai and Wei Yu. “Sparse Beamforming and User-Centric Clustering for Downlink Cloud Radio Access Network.” In: *Access, IEEE* 2 (2014). DOI: `10.1109/ACCESS.2014.2362860`.

[Eri17] Ericsson. *Ericsson Mobility Report*. Tech. rep. Nov. 2017. URL: `https://www.ericsson.com/en/mobility-report/reports/november-2017`.

[Fal+10] Hossein Falaki et al. “A First Look at Traffic on Smartphones.” In: *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*. IMC ’10. Melbourne, Australia: ACM, 2010. ISBN: 978-1-4503-0483-2. DOI: `10.1145/1879141.1879176`.

[FHN12] Viktor Farkas, Balázs Héder, and Szabolcs Nováczki. "A Split Connection TCP Proxy in LTE Networks." In: *Information and Communication Technologies: 18th EUNICE/ IFIP WG 6.2, 6.6 International Conference, EUNICE 2012, Budapest, Hungary, August 29-31, 2012. Proceedings*. Ed. by Róbert Szabó and Attila Vidács. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. ISBN: 978-3-642-32808-4. DOI: `10.1007/978-3-642-32808-4_24`.

[Fie54] E. C. Fieller. "Some Problems in Interval Estimation." In: *Journal of the Royal Statistical Society. Series B (Methodological)* 16.2 (1954). URL: `http://www.jstor.org/stable/2984043`.

[GBZ14] F. Guidolin, L. Badia, and M. Zorzi. "A Distributed Clustering Algorithm for Coordinated Multipoint in LTE Networks." In: *IEEE Wireless Communications Letters* 3.5 (Oct. 2014). DOI: `10.1109/LWC.2014.2340405`.

[GKB12] A. Giovanidis, J. Krolikowski, and S. Brueck. "A 0-1 program to form minimum cost clusters in the downlink of cooperating base stations." In: *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. Apr. 2012. DOI: `10.1109/WCNC.2012.6214510`.

[GMR01] Nathan Gartner, Carroll Messer, and Ajay Rathi, eds. *Revised Monograph on Traffic Flow Theory*. Federal Highway Administration Research and Technology, 2001. URL: `https://www.fhwa.dot.gov/publications/research/operations/tft/` (visited on June 13, 2017).

[Gol+03] A. Goldsmith et al. "Capacity limits of MIMO channels." In: *IEEE Journal on Selected Areas in Communications* 21.5 (June 2003). DOI: `10.1109/JSAC.2003.810294`.

[Gon+11] Jie Gong et al. "Joint Scheduling and Dynamic Clustering in Downlink Cellular Networks." In: *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. Dec. 2011. DOI: `10.1109/GLOCOM.2011.6133679`.

[Gre59] Harold Greenberg. "An Analysis of Traffic Flow." In: *Operations Research* 7.1 (1959). URL: `http://www.jstor.org/stable/167595`.

[Grø+17] O. Grøndalen et al. "Scheduling Policies in Time and Frequency Domains for LTE Downlink Channel: A Performance Comparison." In: *IEEE Transactions on Vehicular Technology* 66.4 (Apr. 2017). DOI: `10.1109/TVT.2016.2589462`.

[Gud91] M. Gudmundson. "Correlation model for shadow fading in mobile radio systems." In: *Electronics Letters* 27.23 (Nov. 1991). DOI: `10.1049/el:19911328`.

[Hab+13] Bernd Haberland et al. "Radio Base Stations in the Cloud." In: *Bell Labs Technical Journal* 18.1 (2013). DOI: `10.1002/bltj.21596`.

[Hat80] M. Hata. "Empirical formula for propagation loss in land mobile radio services." In: *IEEE Transactions on Vehicular Technology* 29.3 (Aug. 1980). DOI: `10.1109/T-VT.1980.23859`.

[Her+04] F. Hernández-Campos et al. “Variable heavy tails in internet traffic.” In: *Perform. Eval.* 58.2+3 (Nov. 2004). DOI: 10.1016/j.peva.2004.07.008.

[HG17] S. Haddadi and A. Ghasemi. “Coordinated multi-point joint transmission evaluation in heterogenous cloud radio access networks.” In: *2017 Iranian Conference on Electrical Engineering (ICEE).* May 2017. DOI: 10.1109/IranianCEE.2017.7985372.

[HM72] H. Harashima and H. Miyakawa. “Matched-Transmission Technique for Channels With Intersymbol Interference.” In: *IEEE Transactions on Communications* 20.4 (Aug. 1972). DOI: 10.1109/TCOM.1972.1091221.

[Hon+13] Mingyi Hong et al. “Joint Base Station Clustering and Beamformer Design for Partial Coordinated Transmission in Heterogeneous Networks.” In: *Selected Areas in Communications, IEEE Journal on* 31.2 (Feb. 2013). DOI: 10.1109/JSAC.2013.130211.

[HPS05] B. M. Hochwald, C. B. Peel, and A. L. Swindlehurst. “A vector-perturbation technique for near-capacity multiantenna multiuser communication-part II: perturbation.” In: *IEEE Transactions on Communications* 53.3 (Mar. 2005). DOI: 10.1109/TCOMM.2004.841997.

[HTTP Archive] HTTP Archive. June 12, 2017. URL: http://httparchive.org (visited on June 12, 2017).

[Hua+13] Junxian Huang et al. “An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance.” In: *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM.* SIGCOMM ’13. Hong Kong, China: ACM, 2013. ISBN: 978-1-4503-2056-6. DOI: 10.1145/2486001.2486006.

[Hub57] Matthew J Huber. “Effect of temporary bridge on parkway performance.” In: *Highway Research Board Bulletin* 167 (1957).

[IEEE P1914.3] P1914.3. *Standard for Radio Over Ethernet Encapsulations and Mappings.* Tech. rep. IEEE, 2017. URL: https://standards.ieee.org/develop/project/1914.3.html.

[IP11] Sunghwan Ihm and Vivek S. Pai. “Towards Understanding Modern Web Traffic.” In: *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference.* IMC ’11. Berlin, Germany: ACM, 2011. ISBN: 978-1-4503-1013-0. DOI: 10.1145/2068816.2068845.

[Irm+11] R. Irmer et al. “Coordinated multipoint: Concepts, performance, and field trial results.” In: *Communications Magazine, IEEE* 49.2 (Feb. 2011). DOI: 10.1109/MCOM.2011.5706317.

[ITU08] ITU-R. *Requirements related to technical performance for IMT-advanced radio interface(s).* Tech. rep. M.2134. Nov. 2008.

[ITU15] ITU-R. *IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond.* Tech. rep. M.2083-0. Sept. 2015.

[IZA14] A. Imran, A. Zoha, and A. Abu-Dayya. "Challenges in 5G: how to empower SON with big data for enabling 5G." In: *IEEE Network* 28.6 (Nov. 2014). DOI: 10.1109/MNET.2014.6963801.

[JPP00] A. Jalali, R. Padovani, and R. Pankaj. "Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system." In: *VTC2000-Spring. 2000 IEEE 51st Vehicular Technology Conference Proceedings (Cat. No.00CH37026)*. Vol. 3. 2000. DOI: 10.1109/VETECS.2000.851593.

[Kar+17] M. Karavolos et al. "A Dynamic Hybrid Clustering Scheme for LTE-A networks employing CoMP-DPS." In: *2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. June 2017. DOI: 10.1109/CAMAD.2017.8031614.

[Kel97] Frank Kelly. "Charging and rate control for elastic traffic." In: *European transactions on Telecommunications* 8.1 (1997).

[Kha+11] Z. Khan et al. "Modeling the Dynamics of Coalition Formation Games for Cooperative Spectrum Sharing in an Interference Channel." In: *IEEE Transactions on Computational Intelligence and AI in Games* 3.1 (Mar. 2011). DOI: 10.1109/TCIAIG.2010.2080358.

[Kin+17] K. Kinoshita et al. "A CoMP clustering method in consideration of spectrum sharing for fairness improvement." In: *2017 9th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. Nov. 2017. DOI: 10.1109/ICUMT.2017.8255192.

[KK16] H. S. Kang and D. K. Kim. "User-Centric Overlapped Clustering Based on Anchor-Based Precoding in Cellular Networks." In: *IEEE Communications Letters* 20.3 (Mar. 2016). DOI: 10.1109/LCOMM.2016.2515085.

[Knu98] D.E. Knuth. *The Art of Computer Programming: Sorting and searching*. The Art of Computer Programming. Addison-Wesley, 1998. ISBN: 9780201896855.

[Köh14] Uwe Köhler. *Einführung in die Verkehrsplanung: Grundlagen, Modellbildung, Verkehrsprognose, Verkehrsnetze*. Stuttgart: Fraunhofer-IRB-Verl., 2014. ISBN: 978-3-8167-9041-9. URL: http://deposit.d-nb.de/cgi-bin/dokserv?id=4572587&prov=M&dok_var=1&dok_ext=htm.

[KS10] Farhan Khalid and Joachim Speidel. "Advances in MIMO Techniques for Mobile Communications - A Survey." In: *International Journal of Communications, Network and System Sciences* 3.03 (2010). DOI: 10.4236/ijcns.2010.33031.

[KSB09] Jonghyun Kim, Vinay Sridhara, and Stephan Bohacek. "Realistic mobility simulation of urban mesh networks." In: *Ad Hoc Networks* 7.2 (2009). DOI: 10.1016/j.adhoc.2008.04.008. URL: http://www.sciencedirect.com/science/article/pii/S1570870508000590.

[LC15] Y. Li and M. Chen. "Software-Defined Network Function Virtualization: A Survey." In: *IEEE Access* 3 (2015). DOI: 10.1109/ACCESS.2015.2499271.

[LCJ17] Feng Li, Jae Won Chung, and Xiaoxiao Jiang. "Driving TCP Congestion Control Algorithms on Highway." In: *Netdev 2.1, The Technical Conference on Linux Networking* (Apr. 17, 2017).

[Lee+12] Daewon Lee et al. "Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges." In: *Communications Magazine, IEEE* 50.2 (Feb. 2012). DOI: 10.1109/MCOM.2012.6146494.

[Lee+13] N. Lee et al. "Base station cooperation with dynamic clustering in super-dense cloud-RAN." In: *2013 IEEE Globecom Workshops (GC Wkshps)*. Dec. 2013. DOI: 10.1109/GLOCOMW.2013.6825084.

[Leu88] Wilhelm Leutzbach. *Introduction to the theory of traffic flow*. Berlin; Heidelberg [u.a.]: Springer, 1988. ISBN: 3-540-17113-4. URL: http://swbplus.bsz-bw.de/bsz013704311cov.htm.

[Li+14] G. Y. Li et al. "Multi-Cell Coordinated Scheduling and MIMO in LTE." In: *IEEE Communications Surveys Tutorials* 16.2 (2014). DOI: 10.1109/SURV.2014.022614.00186.

[Lin+10] Y. Lin et al. "Wireless network cloud: Architecture and system requirements." In: *IBM Journal of Research and Development* 54.1 (Jan. 2010). DOI: 10.1147/JRD.2009.2037680.

[Liu+15] D. Liu et al. "Semi-Dynamic User-Specific Clustering for Downlink Cloud Radio Access Network." In: *Vehicular Technology, IEEE Transactions on* PP.99 (2015). DOI: 10.1109/TVT.2015.2431917.

[Lou+14] Thomas Louail et al. "From mobile phone data to the spatial structure of cities." In: *Scientific Reports* 4 (June 2014). DOI: 10.1038/srep05276.

[Lov+08] D. J. Love et al. "An overview of limited feedback in wireless communication systems." In: *IEEE Journal on Selected Areas in Communications* 26.8 (Oct. 2008). DOI: 10.1109/JSAC.2008.081002.

[LW09] Jingxin Liu and Dongming Wang. "An improved dynamic clustering algorithm for multi-user distributed antenna system." In: *Wireless Communications Signal Processing, 2009. WCSP 2009. International Conference on*. Nov. 2009. DOI: 10.1109/WCSP.2009.5371570.

[LZ08] Z. Q. Luo and S. Zhang. "Dynamic Spectrum Management: Complexity and Duality." In: *IEEE Journal of Selected Topics in Signal Processing* 2.1 (Feb. 2008). DOI: 10.1109/jstsp.2007.914876.

[LZZ14] B. Li, T. Zhang, and Z. Zeng. "Bipartite network based multi-cell clustering scheme in randomly located CoMP systems." In: *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*. Sept. 2014. DOI: 10.1109/PIMRC.2014.7136193.

[Maa+12] Helka-Liina Maattanen et al. "System-level performance of LTE-Advanced with joint transmission and dynamic point selection schemes." In: *EURASIP Journal on Advances in Signal Processing* 2012.1 (2012). DOI: 10.1186/1687-6180-2012-247. URL: http://asp.eurasipjournals.com/content/2012/1/247.

[Mah13] Shyam Babu Mahato. “Radio Resource Scheduling in Homogeneous Coordinated Multi-Point Joint Transmission of Future Mobile Networks.” PhD thesis. University of Bedfordshire, Department of Computer Science & Technology, June 2013.

[Man+15] K. Manolakis et al. “Cooperative Cellular Networks: Overcoming the Effects of Real-World Impairments.” In: *IEEE Vehicular Technology Magazine* 10.3 (Sept. 2015). DOI: 10.1109/MVT.2015.2446420.

[Mar+16] R. Margolies et al. “Exploiting Mobility in Proportional Fair Cellular Scheduling: Measurements and Algorithms.” In: *IEEE/ACM Transactions on Networking* 24.1 (Feb. 2016). DOI: 10.1109/TNET.2014.2362928.

[Mar10] Patrick Marsch. “Coordinated Multi-Point under a Constrained Backhaul and Imperfect Channel Knowledge.” PhD thesis. Technische Universität Dresden, Mar. 2010. URL: http://www.pmarsch.de/diss.pdf.

[MC13] Jung-Min Moon and Dong-Ho Cho. “Formation of cooperative cluster for coordinated transmission in multi-cell wireless networks.” In: *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC).* Jan. 2013. DOI: 10.1109/CCNC.2013.6488494.

[MET16] METIS-II. *Deliverable D1.1Refined scenarios and requirements, consolidated use cases, and qualitative techno-economic feasibility assessment.* Tech. rep. Mobile and wireless communications Enablers for the Twenty-twenty Information Society-II, Jan. 31, 2016.

[MF11] Patrick Marsch and Gerhard Fettweis, eds. *Coordinated Multi-Point in Mobile Communications - from theory to practice.* Cambridge University Press, 2011.

[MGF15] Nahid Mohajeri, Agust Gudmundsson, and Jon R. French. “CO2 emissions in relation to street-network configuration and city size.” In: *Transportation Research Part D: Transport and Environment* 35 (2015). DOI: http://dx.doi.org/10.1016/j.trd.2014.11.025. URL: http://www.sciencedirect.com/science/article/pii/S1361920914001886.

[Mis07] Ajay Ranjan Mishra, ed. *Advanced cellular network planning and optimisation: 2G/2.5G/3G ... evolution to 4G.* Chichester: Wiley, 2007. ISBN: 0-470-01471-7. URL: http://swbplus.bsz-bw.de/bsz257557903cov.htm.

[Mog+07] P. Mogensen et al. “LTE Capacity Compared to the Shannon Bound.” In: *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring.* Apr. 2007. DOI: 10.1109/VETECS.2007.260.

[MSF10] Gregor Maier, Fabian Schneider, and Anja Feldmann. “A first look at mobile hand-held device traffic.” In: PAM’10 (2010). URL: http://dl.acm.org/citation.cfm?id=1889324.1889341.

[MTP90] S. Mockford, A. M. D. Turkmani, and J. D. Parsons. “Local mean signal variability in rural areas at 900 MHz.” In: *40th IEEE Conference on Vehicular Technology.* May 1990. DOI: 10.1109/VETEC.1990.110392.

[Mül11] C. M. Müller. "Analysis of interactions between Internet data traffic characteristics and Coordinated Multipoint transmission schemes." In: *2011 IEEE Wireless Communications and Networking Conference*. Mar. 2011. DOI: 10.1109/WCNC.2011.5779138.

[Nec06] Marc C. Necker. "A comparison of scheduling mechanisms for service class differentiation in HSDPA networks." In: *AEU - International Journal of Electronics and Communications* 60.2 (2006). DOI: 10.1016/j.aeue.2005.11.014. URL: http://www.sciencedirect.com/science/article/pii/S143484110500186X.

[Nec09] M.C. Necker. "A Novel Algorithm for Distributed Dynamic Interference Coordination in Cellular OFDMA Networks - Communication Networks and Computer Engineering Report No. 101." PhD thesis. Universität Stuttgart, 2009.

[Net13] MobileCloud Networking. *D3.1 Infrastructure Management Foundations – Specifications & Design for Mobile Cloud framework*. Deliverable 3.1. Future Communication Architecture for Mobile Cloud Services, Nov. 8, 2013.

[Ngu+17] V. G. Nguyen et al. "SDN/NFV-Based Mobile Packet Core Network Architectures: A Survey." In: *IEEE Communications Surveys Tutorials* 19.3 (2017). DOI: 10.1109/COMST.2017.2690823.

[OBS06] OBSAI. *BTS SYSTEM REFERENCE DOCUMENT*. Tech. rep. Open Base Station Architecture Initiative, 2006. URL: http://www.obsai.com/specs/OBSAI_System_Spec_V2.0.pdf.

[OFl+96] CA O'Flaherty et al., eds. *Transport planning and traffic engineering*. Oxford: Butterworth-Heinemann, 1996. ISBN: 0-34066-279-4. URL: http://www.sciencedirect.com/science/book/9780340662793.

[Ols+09] Magnus Olsson et al. *SAE and the Evolved Packet Core: Driving the Mobile Broadband Revolution*. Academic Press, 2009. ISBN: 0123748267, 9780123748263.

[PF05] D. P. Palomar and J. R. Fonollosa. "Practical algorithms for a family of waterfilling solutions." In: *IEEE Transactions on Signal Processing* 53.2 (Feb. 2005). DOI: 10.1109/TSP.2004.840816.

[PGH08] A. Papadogiannis, D. Gesbert, and E. Hardouin. "A Dynamic Clustering Approach in Wireless Networks with Multi-Cell Cooperative Processing." In: *Communications, 2008. ICC '08. IEEE International Conference on*. May 2008. DOI: 10.1109/ICC.2008.757.

[PHS05] C. B. Peel, B. M. Hochwald, and A. L. Swindlehurst. "A vector-perturbation technique for near-capacity multiantenna multiuser communication-part I: channel inversion and regularization." In: *IEEE Transactions on Communications* 53.1 (Jan. 2005). DOI: 10.1109/TCOMM.2004.840638.

[PLH16] J. Park, N. Lee, and R. W. Heath. "Cooperative Base Station Coloring for Pair-Wise Multi-Cell Coordination." In: *IEEE Transactions on Communications* 64.1 (Jan. 2016). DOI: 10.1109/TCOMM.2015.2495355.

[Pro+12] Magnus Proebster et al. "Context-aware resource allocation for cellular wireless networks." In: *EURASIP Journal on Wireless Communications and Networking* 2012.1 (2012). DOI: 10.1186/1687-1499-2012-216.

[QT14] C. Qin and H. Tian. "A greedy dynamic clustering algorithm of joint transmission in dense small cell deployment." In: *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*. Jan. 2014. DOI: 10.1109/CCNC.2014.6866638.

[Rah+15] M. M. U. Rahman et al. "RRH Clustering and Transmit Precoding for Interference-Limited 5G CRAN Downlink." In: *2015 IEEE Globecom Workshops (GC Wkshps)*. Dec. 2015. DOI: 10.1109/GLOCOMW.2015.7414198.

[Ram+14] J.J. Ramos-munoz et al. "Characteristics of mobile YouTube traffic." In: *Wireless Communications, IEEE* 21.1 (Feb. 2014). DOI: 10.1109/MWC.2014.6757893.

[Rat+06] Carlo Ratti et al. "Mobile Landscapes: Using Location Data from Cell Phones for Urban Analysis." In: *Environment and Planning B: Planning and Design* 33.5 (2006). DOI: 10.1068/b32047. eprint: http://dx.doi.org/10.1068/b32047.

[RC09] S. A. Ramprashad and G. Caire. "Cellular vs. Network MIMO: A comparison including the channel state information overhead." In: *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*. Sept. 2009. DOI: 10.1109/PIMRC.2009.5450085.

[RFC 2581] M. Allman, V. Paxson, and W. Stevens. *TCP Congestion Control*. RFC 2581. IETF, Apr. 1999.

[RFC 3135] J. Border et al. *Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations*. RFC 3135. IETF, June 2001.

[RFC 768] J. Postel. *User Datagram Protocol*. RFC 768. IETF, Aug. 1980.

[RFC 793] J. Postel. *Transmission Control Protocol*. RFC 793. IETF, Sept. 1981.

[Ros+13] Dennis M. Rose et al. "The IC 1004 Urban Hannover Scenario – 3D Pathloss Predictions and Realistic Traffic and Mobility Patterns." In: *COST IC1004 TD(13)08054* (2013).

[RTL98] F. Rashid-Farrokhi, L. Tassiulas, and K. J. R. Liu. "Joint optimal power control and beamforming in wireless networks using antenna arrays." In: *IEEE Transactions on Communications* 46.10 (Oct. 1998). DOI: 10.1109/26.725309.

[Saa+09] W. Saad et al. "A distributed coalition formation framework for fair user cooperation in wireless networks." In: *IEEE Transactions on Wireless Communications* 8.9 (Sept. 2009). DOI: 10.1109/TWC.2009.080522.

[San16] Sandvine Incorporated. *Global Internet Phenomena Report*. Tech. rep. 2016. URL: http://www.sandvine.com (visited on Feb. 16, 2017).

[Saw+10] M. Sawahashi et al. “Coordinated multipoint transmission/reception techniques for LTE-advanced [Coordinated and Distributed MIMO].” In: *Wireless Communications, IEEE* 17.3 (June 2010). DOI: 10.1109/MWC.2010.5490976.

[Sch17] Sebastian Scholz. “Combining Dynamic Clustering and Scheduling for Coordinated Multi-Point Transmission in LTE.” In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Oct. 2017.

[Sen+16] Ryuma Seno et al. “A low-complexity cell clustering algorithm in dense small cell networks.” In: *EURASIP Journal on Wireless Communications and Networking* 2016.1 (2016). DOI: 10.1186/s13638-016-0765-3. URL: http://dx.doi.org/10.1186/s13638-016-0765-3.

[SG16] S. Scholz and H. Grob-Lipski. “Reallocation strategies for user processing tasks in future cloud-RAN architectures.” In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Sept. 2016. DOI: 10.1109/PIMRC.2016.7794958.

[SGS14] N. Shojaedin, M. Ghaderi, and A. Sridharan. “TCP-aware scheduling in LTE networks.” In: *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*. June 2014. DOI: 10.1109/WoWMoM.2014.6918946.

[Sha49] C. E. Shannon. “Communication in the Presence of Noise.” In: *Proceedings of the IRE* 37.1 (Jan. 1949). DOI: 10.1109/JRPROC.1949.232969.

[She+06] Zukang Shen et al. “Low complexity user selection algorithms for multiuser MIMO systems with block diagonalization.” In: *Signal Processing, IEEE Transactions on* 54.9 (Sept. 2006). DOI: 10.1109/TSP.2006.879269.

[SimLib] *IKR Simulation Library (SimLib)*. May 10, 2017. URL: http://www.ikr.uni-stuttgart.de/Content/IKRSimLib/ (visited on May 10, 2017).

[SLA11] I. Sohn, S. H. Lee, and J. G. Andrews. “Belief Propagation for Distributed Downlink Beamforming in Cooperative MIMO Cellular Networks.” In: *IEEE Transactions on Wireless Communications* 10.12 (Dec. 2011). DOI: 10.1109/TWC.2011.101210.101698.

[Spe+04] Q. H. Spencer et al. “An introduction to the multi-user MIMO downlink.” In: *IEEE Communications Magazine* 42.10 (Oct. 2004). DOI: 10.1109/MCOM.2004.1341262.

[SSH04] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt. “Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels.” In: *IEEE Transactions on Signal Processing* 52.2 (Feb. 2004). DOI: 10.1109/TSP.2003.821107.

[STB11] Stefania Sesia, Issam Toufik, and Matthew Baker, eds. *LTE - the UMTS long term evolution: from theory to practice*. Second edition. UB Vaihingen. Chichester: Wiley, 2011. ISBN: 978-0-470-66025-6.

[Sti+10] Clemens Stierstorfer et al. “Network MIMO Downlink Transmission.” In: *Proceedings of 15th International OFDM Workshop (InOWo)*. Hamburg, Germany, Sept. 2010. URL: http://www.lit.lnt.de/papers/ofdm_2010_stierstorfer.pdf.

[Stü01] Gordon L. Stüber. *Principles of mobile communication*. 2. ed. Boston, Mass.: Kluwer Academic, 2001. ISBN: 0-7923-7998-5.

[SZ01] S. Shamai and B. M. Zaidel. “Enhancing the cellular downlink capacity via co-processing at the transmitting end.” In: *IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No.01CH37202)*. Vol. 3. 2001. DOI: 10.1109/VETECS.2001.944993.

[Tan+17] J. Tang et al. “Fully Exploiting Cloud Computing to Achieve a Green and Flexible C-RAN.” In: *IEEE Communications Magazine* 55.11 (Nov. 2017). DOI: 10.1109/MCOM.2017.1600922.

[Thi+12] L. Thiele et al. “User-aided sub-clustering for CoMP transmission: Feedback overhead vs. data rate trade-off.” In: *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*. Nov. 2012. DOI: 10.1109/ACSSC.2012.6489199.

[Tom71] M. Tomlinson. “New automatic equaliser employing modulo arithmetic.” In: *Electronics Letters* 7.5 (Mar. 1971). DOI: 10.1049/el:19710089.

[TP15] Tuyen X Tran and Dario Pompili. “Dynamic Radio Cooperation for Downlink Cloud-RANs with Computing Resource Sharing.” In: *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on*. IEEE. 2015.

[Tra+12] L. Tran et al. “Fast Converging Algorithm for Weighted Sum Rate Maximization in Multicell MISO Downlink.” In: *Signal Processing Letters, IEEE* 19.12 (Dec. 2012). DOI: 10.1109/LSP.2012.2223211.

[Tra+15] Roberto Trasarti et al. “Discovering Urban and Country Dynamics from Mobile Phone Data with Spatial Correlation Patterns.” In: *Telecommun. Policy* 39.3 (May 2015). DOI: 10.1016/j.telpol.2013.12.002.

[Tru17] Volker Trucksees. “Modellierung und Bewertung von Nutzerbewegung in Mobilfunknetzen.” Studienarbeit. Universität Stuttgart, 2017.

[TTV07] F. Theoleyre, R. Tout, and F. Valois. “New metrics to evaluate mobility models properties.” In: *2007 2nd International Symposium on Wireless Pervasive Computing*. Feb. 2007. DOI: 10.1109/ISWPC.2007.342624.

[TV05] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[VVH03] H. Viswanathan, S. Venkatesan, and H. Huang. “Downlink capacity evaluation of cellular networks with known-interference cancellation.” In: *IEEE Journal on Selected Areas in Communications* 21.5 (June 2003). DOI: 10.1109/JSAC.2003.810346.

[Wan+09] Q. Wang et al. “Application of BBU+RRU Based Comp System to LTE-Advanced.” In: *2009 IEEE International Conference on Communications Workshops*. June 2009. DOI: 10.1109/ICCW.2009.5208027.

[Wan+15] Huandong Wang et al. "Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment." In: *Proceedings of the 2015 Internet Measurement Conference*. IMC '15. Tokyo, Japan: ACM, 2015. ISBN: 978-1-4503-3848-6. DOI: 10.1145/2815675.2815680.

[Web+11] R. Weber et al. "Self-Organizing Adaptive Clustering for Cooperative Multipoint Transmission." In: *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*. May 2011. DOI: 10.1109/VETECS.2011.5956490.

[Wer+15] T. Werthmann et al. "Task assignment strategies for pools of baseband computation units in 4G cellular networks." In: *2015 IEEE International Conference on Communication Workshop (ICCW)*. June 2015. DOI: 10.1109/ICCW.2015.7247589.

[Wil15] Thorsten Wild. "Channel Estimation and Precoding in Closed-Loop Distributed Multi-Antenna Systems." PhD thesis. University of Stuttgart, 2015.

[WP16] D. Wübben and H. Paul. "Analysis of virtualized Turbo-decoder implementation for Cloud-RAN systems." In: *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*. Sept. 2016. DOI: 10.1109/ISTC.2016.7593142.

[Xu+11] Qiang Xu et al. "Identifying Diverse Usage Behaviors of Smartphone Apps." In: *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*. IMC '11. Berlin, Germany: ACM, 2011. ISBN: 978-1-4503-1013-0. DOI: 10.1145/2068816.2068847.

[XY13] Y. Xia and C. K. Yeo. "Measuring Group Mobility: A Topology Based Approach." In: *IEEE Wireless Communications Letters* 2.1 (Feb. 2013). DOI: 10.1109/WCL.2012.101812.120699.

[YL07] W. Yu and T. Lan. "Transmitter Optimization for the Multi-Antenna Downlink With Per-Antenna Power Constraints." In: *IEEE Transactions on Signal Processing* 55.6 (June 2007). DOI: 10.1109/TSP.2006.890905.

[YT2016] youtube.com. *snapshot available at https://web.archive.org/web/20161130040108/https://youtube.com/yt/press/statistics.html*. Nov. 30, 2016. URL: https://youtube.com/yt/press/statistics.html (visited on Nov. 30, 2016).

[Yu+01] Wei Yu et al. "Iterative water-filling for Gaussian vector multiple access channels." In: *Proceedings. 2001 IEEE International Symposium on Information Theory (IEEE Cat. No.01CH37252)*. 2001. DOI: 10.1109/ISIT.2001.936185.

[ZÅ12] Ying Zhang and Ake Årvidsson. "Understanding the Characteristics of Cellular Data Traffic." In: *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*. CellNet '12. Helsinki, Finland: ACM, 2012. ISBN: 978-1-4503-1475-6. DOI: 10.1145/2342468.2342472.

[Zha+12] Jingjing Zhao et al. “An overlapped clustering scheme of coordinated multi-point transmission for LTE-A systems.” In: *2012 IEEE 14th International Conference on Communication Technology*. Nov. 2012. DOI: `10.1109/ICCT.2012.6511266`.

[Zha+15] H. Zhang et al. “A Practical Semidynamic Clustering Scheme Using Affinity Propagation in Cooperative Picocells.” In: *IEEE Transactions on Vehicular Technology* 64.9 (Sept. 2015). DOI: `10.1109/TVT.2014.2361931`.

[Zha+16] L. Zhang et al. “Performance Analysis and Optimal Cooperative Cluster Size for Randomly Distributed Small Cells Under Cloud RAN.” In: *IEEE Access* 4 (2016). DOI: `10.1109/ACCESS.2016.2550758`.

[Zho+09] S. Zhou et al. “A Decentralized Framework for Dynamic Downlink Base Station Cooperation.” In: *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*. Nov. 2009. DOI: `10.1109/GLOCOM.2009.5425212`.

[Zho+13] D. Zhou et al. “Evaluation of TCP performance with LTE downlink schedulers in a vehicular environment.” In: *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. July 2013. DOI: `10.1109/IWCMC.2013.6583704`.

[Zhu+11] ZhenBo Zhu et al. “Virtual Base Station Pool: Towards a Wireless Network Cloud for Radio Access Networks.” In: *Proceedings of the 8th ACM International Conference on Computing Frontiers*. CF ’11. Ischia, Italy: ACM, 2011. ISBN: 978-1-4503-0698-0. DOI: `10.1145/2016604.2016646`.

[ZXG04] Biao Zhou, Kaixin Xu, and M. Gerla. “Group and swarm mobility models for ad hoc network scenarios using virtual tracks.” In: *IEEE MILCOM 2004. Military Communications Conference, 2004*. Vol. 1. Oct. 2004. DOI: `10.1109/MILCOM.2004.1493283`.