Analysis of interactions between Internet data traffic characteristics and Coordinated Multipoint transmission schemes

Christian M. Mueller
University of Stuttgart, Institute of Communication Networks & Computer Engineering
Pfaffenwaldring 47, 70569 Stuttgart, Germany
Email: christian.mueller@ikr.uni-stuttgart.de

***Abstract*—Several studies have shown the potential of Coordinated Multipoint (CoMP) transmission schemes to increase spectral efficiency. However, most CoMP algorithms are susceptible to outdated channel state information, which is a consequence of inter-eNodeB signaling over the backhaul network. User traffic variation constitutes another time-varying process which interacts with CoMP algorithms. These interactions usually are undesired and can reduce achievable CoMP gains in practical settings. We analyze these interactions using an abstract CoMP model and realistic traffic patterns. We propose two different ways to initiate coordinated transmission and present an analysis of the resulting coordination gains and transmission delays at the application layer. Our analysis reveals that traffic characteristics have significant impact on coordination gains as seen by a user.**

I. INTRODUCTION

Coordinated multipoint (CoMP) transmission schemes are a promising technique to increase the spectral efficiency of cellular networks. Following the 3GPP terminology in the development of the LTE-Advanced specifications [1], we distinguish between the CoMP schemes Joint Transmission, Joint Detection, Coordinated Beamforming and Coordinated Scheduling. The common aspect of all these schemes is to improve the signal-to-interference ratio at the user terminals by coordinated resource allocation and scheduling among neighboring base stations. This improvement is achieved either by choosing orthogonal resources for interfering transmissions in the time, frequency or space domain, or by simultaneous transmission of the same data from several base stations. An overview and brief description of the various CoMP schemes is provided in [2]. In the following, we focus on Coordinated Scheduling and Coordinated Beamforming (CS/CB) in the downlink of an LTE-like system. We further assume all eNodeBs to cooperate using the X2 interface and do not consider deployments with remote radio heads.

CoMP schemes have been analyzed by a number of researches. Large theoretical gains have been reported and its potential has been confirmed by field trials. For details, consider [3] and the references therein. While early studies were conducted in an ideal setup with perfect channel state information and zero delay for the signaling between base stations, more recent studies show that CoMP gains are sensitive to imperfect channel state information [4], [5]. In most of these studies, a full buffer traffic model was assumed, meaning that users always have data to transmit. This is a perfectly valid assumption to determine the theoretically achievable gains of the various CoMP schemes, but will lead to overly optimistic results if Internet data traffic is considered. Internet data traffic consists of burst arrivals of packets and idle times between these bursts. It thus constitutes a time-varying process whose characteristics have to be taken into account to determine the benefits of CoMP from a user perspective. Interactions between window-based transport protocols (e.g. TCP) and CoMP algorithms might lead to unwanted effects such as bad link utilization or large transmission delay, as it was experienced for TCP flows transmitted over GPRS networks [6]. Understanding these interactions is crucial for the design of efficient PHY and MAC layer algorithms to let applications benefit from the higher CoMP data rates.

Our main contribution is the analysis of possible interactions between user traffic characteristics, especially web traffic, and the CoMP schemes Coordinated Scheduling and Coordinated Beamforming. To the best of our knowledge, no previous work on this issue exists. Our analysis makes use of an abstract CoMP model and discusses two different ways of how CoMP transmissions are initiated. We analyze the resulting goodput and transmission delays at the application layer and provide a measure for the overhead involved in coordinated transmissions. To adequately model user traffic characteristics, we use measurement data of Internet data traffic and a well-known web traffic model.

This paper is organized as follows: Section II describes our coordination model and two variants of how coordinated user data transmission is initiated. Section III presents the traffic models used in our analysis. Section IV contains a semi-analytical evaluation of coordination gains, whereas section V contains further results achieved by simulation. Finally, section VI draws conclusions.

II. COORDINATION MODEL

The setup of a CS/CB transmission involving multiple base stations generally includes the following steps:

1) In addition to CQI (Channel Quality Indication) and PMI (Precoding Matric Indication) measurements for its serving cell, the user terminal is requested to perform and report additional measurements to identify interferers.

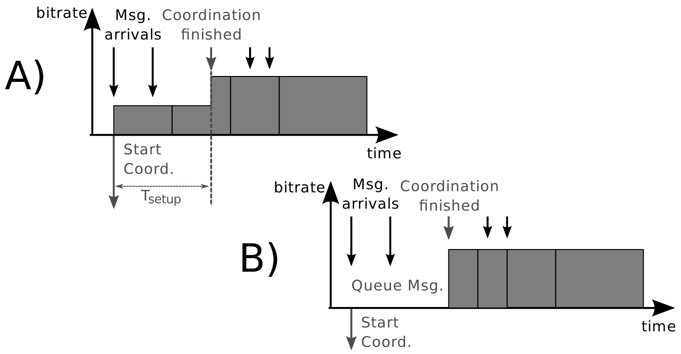


Fig. 1: Variants on the initiation of coordinated transmission

2) Based on this information, the base station decides which neighboring cells are part of the CoMP set and sends signaling messages over the X2 interface to start a negotiation on resource blocks and PMIs to be used for this transmission. The coordination algorithms usually require one or more round-trip-times over the X2 interface to complete.
3) After this preparation, the base station starts transmitting data to the user terminal on the coordinated resources.

To abstract from a particular coordination algorithm, we define the term *coordination setup time* T_{setup} as the sum of steps 1 and 2. We measure the coordination setup delay starting from the time the user data arrives at the base station. Our setup delay T_{setup} thus also includes the time required for scheduling and is slightly larger than just the signaling message exchange over the X2 interface.

In our analysis, we take the point of view of one particular base station which initiates or stops coordination processes with its neighbors. To gain a general insight in possible interactions between CoMP algorithms and traffic characteristics, we do not explicitly model channel measurements nor signalling message exchanges because they are algorithm-specific and differ from one CoMP algorithm to another. We also do not account for the fact that neighboring base stations might also initiate coordinated transmissions. This is valid as long as we only analyze the transmission towards a single user and if we assume all transmissions in a cell to be independent.

A. Variants and Parameters of Coordinated Transmissions

We consider two basic alternatives about when to start a coordinated transmission. For *variant A* in Fig. 1, as soon as a message is received in the downlink, the message is being transmitted on uncoordinated resources at a certain constant rate R_U. At the same time, the base station initiates the coordination process with its neighbor base stations. After time T_{setup}, the coordination process is finished and the base stations have agreed on a set of resources, e.g. resource blocks and corresponding precoding vectors. In our model, the remaining data of the user queue is then transmitted at a higher rate $R_{\text{C}} = g \cdot R_U$ with $g > 1$.

For *variant B*, an incoming message also initiates the coordination process, but the data is buffered until the coordination process is complete. The message is thus sent completely at the higher rate R_{C}, but suffers from a small additional delay.

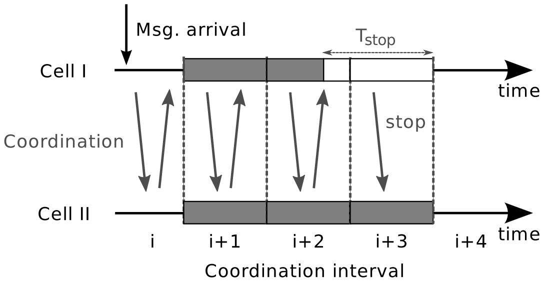


Fig. 2: Coordination process among two neighbor cells

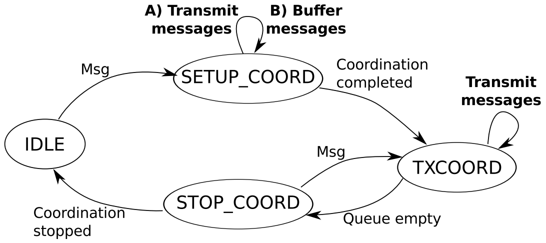


Fig. 3: Coordination model as finite state machine

During user data transmission, the coordination process among the neighboring base stations has to be continuously ongoing to account for user mobility and varying radio channel quality. Many CoMP algorithms propose to coordinate resource usage on time intervals larger than the TTI of the system. We denote this coordination interval length as T_{int}.

Once the user queue runs empty, all user data have been transmitted and coordination can be stopped. To relax the constraints on the time/frequency resources imposed by coordination, again signaling messages have to be exchanged with neighbor base stations. This is illustrated in Fig. 2. The exchange of signaling messages in coordination interval i is used to set up coordinated transmission in interval $i+1$. If the user queue runs empty in interval $i+2$, coordination is stopped after interval $i+3$. However, resource usage in interval $i+3$ is still constrained to what has been agreed on in interval $i+2$.

B. Single User Model

To keep our evaluation model simple, we only consider a single user in the downlink of an LTE-like system. A user is served with constant (but different) rates while he is transmitting data on coordinated or uncoordinated resources. We thus abstract from any details of the radio channel. This model is adequate, as we are interested in average values over rather long time intervals. Evaluations with web traffic have to be conducted over multiple hours, due to the heavy-tailed distribution of HTTP object sizes. A per TTI model including fast fading, inter-cell interference and other effects of the radio channel would neither be feasible nor is it necessary here.

Figure 3 depicts the single-user coordination model described in the previous paragraphs as a state machine with four states. The only difference between variants A and B

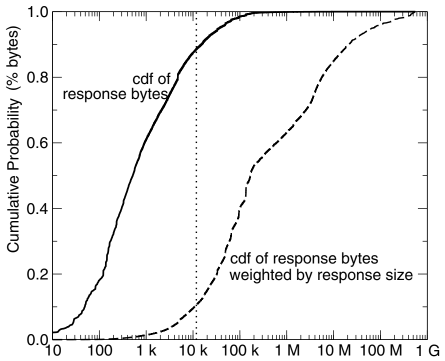


Fig. 4: Distribution of response object sizes in UNC traces

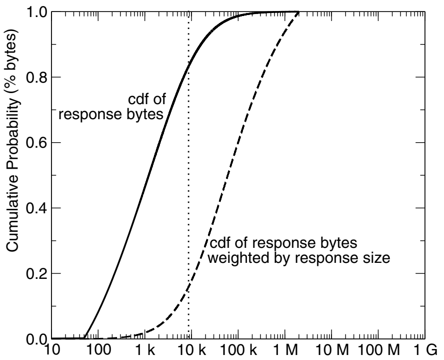


Fig. 5: Distribution of response object size in web traffic model [7]

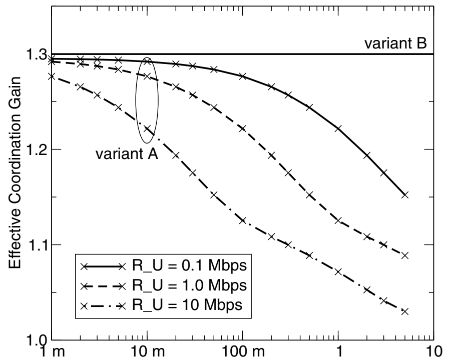


Fig. 6: Effective coordination gains G_A and G_B as a function of T_{setup}

is in the handling of messages in the *SETUP_COORD* state, i.e. when we wait for the coordination process among the base stations to complete. The actual state machine is slightly more complicated due to the handling of coordination intervals, start and stop durations, but its implementation is straightforward.

III. INTERNET DATA TRAFFIC

Data traffic in packet-switched networks, in particular the traffic generated by applications in the Internet, exhibits a very bursty behavior. A data transmission usually consists of several sequences of packets of different size with periods of inactivity in between. The length of these sequences, packet sizes and inactivity periods vary depending on the application and the protocols being used. Furthermore, the available network resources have an influence on the application and protocol behavior, which makes it impossible to develop generally applicable packet-level models of Internet traffic. Consequently, a number of network independent so-called behavioral traffic models have been developed, which mimic the behavior of users and applications above the transport layer.

A. Traffic Models

A well-known behavioral model of web traffic is the model described in [7]. This model was used for evaluations in the context of LTE and is similar to the model presented by [8]. Web traffic here is regarded as a sequence of thinking times, transmission periods and parsing times. A web site consists of a main object and a number of embedded objects, which are downloaded sequentially with parsing times in between. Main object and embedded object sizes follow a truncated lognormal distribution with different parameters, whereas the number of embedded objects follows a truncated heavy-tailed Pareto distribution. The parameters have been empirically derived from web traffic measurements in the 1990s. The model does not support HTTP pipelining nor keepalives. Fig. 5 shows the cumulative distribution function of the object sizes of this model. The solid line gives the CDF of the response object size, which has a mean value of about 10 KBytes. The dashed line gives the response object size, weighted by its contribution to the overall traffic volume for a synthetic trace of 96 GBytes length. As it can be seen, over 80% of the objects are smaller than average, but account for only 10% of the traffic volume. For a detailed description of all model parameters it is referred to [7].

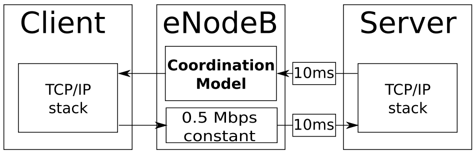
Given that the web traffic model described above is based on relatively old measurement data, we further consider the more recent traffic measurements provided by the University of North Carolina (UNC) [9]. For these traces, the TCP/IP headers of the traffic from and to the UNC campus were captured over a period of several hours in 2008. The traces were processed to a sequence of request-response patterns that are typical for web usage and other applications, following the methodology described in [10]. The trace data is suited for the so-called a-b-t model, which is a non-parametric traffic model for request-response types of applications [11][1]: A request-response communication in this model is described as a sequence of *connection vectors*, which are triplets of request size a, response size b and idle time t. A typical web session can thus be described as a sequence of connection vectors. The traces files include timestamps and sizes of the application layer request and response objects of over 4 million connections. Figure 4 depicts the cumulative distribution function of the response size for the UNC traces, which registered over 11 million response objects. Similar to the web traffic model, small response object sizes are very frequent, but account for only a small fraction of the overall traffic volume. The web traffic model shows a relatively good compliance with the trace data for small to moderate object sizes, although the UNC traces represent a traffic mix and not web traffic only.

B. Transport Protocols

The traffic models presented in the previous section specify the object sizes (web pages, images etc.) at the application layer. To model the packet arrival process at the base station, effects of the transport protocols and the underlying link layer technology have to be taken into account. The main effect of the link layer technology is the definition of minimum

---

[1] The a-b-t model is also applicable to other communication patterns than request-response sequences, but this is not considered here



T_{setup}	10 ms (if not stated otherwise)
T_{int}	1 ms
T_{stop}	equals T_{setup}
g	1.3
Traffic	NGMN web traffic model

Fig. 7: Illustration of simulation setup and parameters

and maximum packet sizes. In the Internet, the packet size distribution has found to be bi-modal, with frequent packet sizes being around 40 Bytes and 1500 Bytes [12].

The predominant transport protocol in the Internet is TCP, which is a window-based protocol. Many different versions of TCP exist, which differ in aspects such as RTT estimation, ACK behavior and congestion control. After connection setup, TCP gradually increases the data volume which is sent out before the sender needs to wait for an acknowledgment of the receiver side. With every received acknowledgment this window size is increased, which results in an exponential increase of the data rate. This is known as the TCP slow start algorithm. A consequence of the TCP slow start is that it takes a couple of round-trip-times to fully utilize the available transmission resources. Furthermore, during slow start, data transmission is not continuous, but is a sequence of IP packet bursts with idle times in between. Both effects might lead to interactions with CoMP algorithms, as will be shown in the results section. For more details on TCP, it is referred to [13].

IV. SEMI-ANALYTICAL EVALUATION

To get a basic understanding of our problem, we first consider the transmission of a single object of size s to a user and neglect the influence coordination intervals and TCP. The transmission time in coordination variant A in Fig. 1 is

$$\tau_A = \begin{cases} s/R_U & \text{for } s \leq R_u \cdot T_{\text{setup}} \\ T_{\text{setup}} + \frac{s - R_U T_{\text{setup}}}{R_C} & \text{otherwise} \end{cases} \tag{1}$$

For variant B, the transmission time simply is

$$\tau_B = T_{\text{setup}} + s/R_C \tag{2}$$

The transmission time τ_A obviously is strictly less than τ_B. For variant A, it might happen that $\tau_A < T_{\text{coord}}$, in which case the higher throughput of the coordinated transmission cannot be utilized because all user data have already been transmitted and the queue has run empty before coordination among the base stations is complete. This behavior does not occur in variant B. The downside of variant B is to trade larger transmission delays against higher throughput. If T_{setup} is larger than just a few milliseconds, the additional delay might have negative effects on the quality of service. For $s \gg R_u T_{\text{setup}}$, both variants show similar transmission times at the same cost. For T_{setup} in the order of a few milliseconds, variant B is preferable. If T_{setup} is larger, the choice depends on the object size distribution and on how transmission delay is weighted against capacity.

The capacity gain of CoMP in variant B, G_B, always equals g. For variant A, the resulting capacity gain G_A is affected by the number and probability of objects with $s \leq R_u T_{\text{coord}}$. It is

$$G_A = \frac{\mathrm{E}[s]/R_U}{\mathrm{E}[\tau_A(s)]} \tag{3}$$

This effective gain is always less than G_B and approaches 1 as the coordination setup time gets larger. Figure 6 depicts the decay of the effective coordination gains G_A and G_B for the weighted response size distribution of the UNC measurements as a function of T_{setup}, if an uncoordinated transmission rate R_U of 1 Mbps and a maximum gain of $g = 1.3$ is assumed[2]. This simple analysis reveals that for high transmission rates and coordination setup times of several 10 ms, only a fraction of the higher throughput from coordinated transmission can actually be utilized in the system if variant A is used. Regarding variant B, gains are not affected by the object size distribution, given that variant B trades higher throughput against transmission delay.

V. SIMULATIVE EVALUATION

In addition to the analysis in the previous section, we consider the implementation of both coordination variants in more detail and analyze which implications they might have on data being sent over TCP.

A. Simulation Methodology

We use a setup consisting of an abstract coordination model as described in section II and a traffic generator. Again, we only consider the transmission to a single user and assume constant, but different transmission rates for uncoordinated and coordinated transmissions. The simulation framework is based on our institutes's simulation library IKR Simlib. We make use of the *Network Simulation Cradle* [14] to account for the influence of TCP and run the TCP/IP stack of the Linux kernel 2.6.26 on a 32 bit system. The stack is configured to use TCP CUBIC with default parameters.

Our simulation setup is illustrated in Fig. 7. User data transmission is organized in TTIs of 1 ms length. In uplink and downlink direction we configured a propagation delay of 10 ms. The transmission rate in the uplink direction is constant and equals 0.5 Mbps. The round-trip-time is thus at least 20 ms. We use the batch-means method with 10 batches and a duration of 3h per batch. The results are presented with 0.95% confidence intervals.

The traffic generator is implemented as a client process that requests web pages from a server process. The web pages consist of main objects and embedded objects. The client is

---

[2] Given that CoMP gains in literature vary largely depending on which scenario is assumed, we arbitrarily chose $g = 1.3$ to illustrate our results. For other values of g, the absolute numbers differ but the principal behavior remains the same.

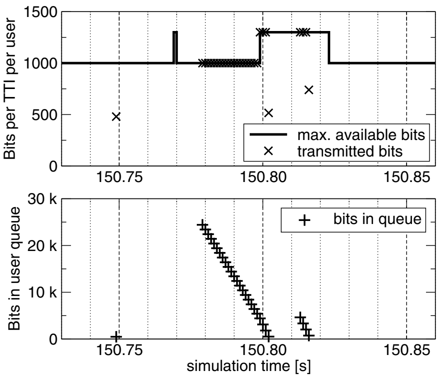


Fig. 8: Example of coordinated transmission of a web page in variant A

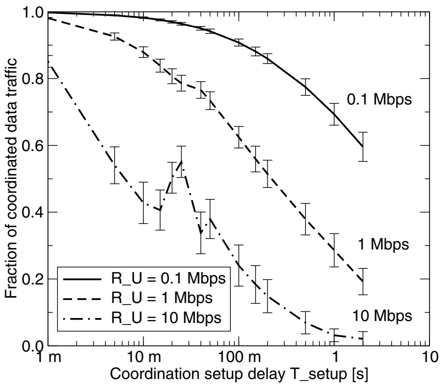


Fig. 9: Coordinated data volume for variant A from simulations

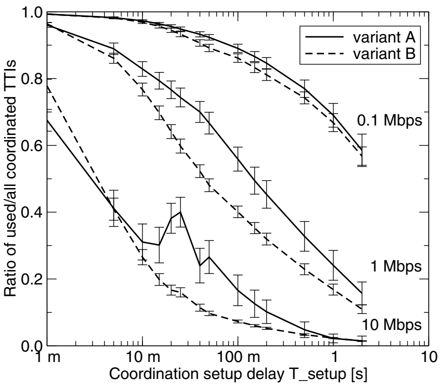


Fig. 10: Coordination overhead as fraction of unnecessarily constrained TTIs

configured to open a new TCP connection for every new web page to sequentially download main and embedded objects, according to [7]. We have chosen the synthetic web traffic model instead of the a-b-t model of the UNC traces, because we expect the smaller object sizes of this model to be more realistic than the UNC data of fixed networks[3].

B. Simulation Results

1) Effective coordination gain: Figure 8 depicts an example of the transmission of a single web page for variant A for $T_{\text{setup}} = 20\,\text{ms}$. The lower graph shows the number of bits in the user buffer over the simulation time. The upper graph gives the number of bits that could be sent to the user in a single TTI (solid line) and the number of bits that have actually been transmitted (X symbols). Around $t = 150.75$, a *SYN + ACK* packet for TCP connection setup has arrived at the eNodeB. T_{setup} later, a TTI for coordinated transmission was scheduled but not used. Around $t = 150.78$, the HTTP response arrives at the eNodeB. It can be seen that only a fraction of the object is being transmitted at the higher rate R_C. Because $T_{\text{setup}} = T_{\text{stop}}$, the coordinated transmission cannot be stopped immediately after the first object was transmitted. In this example, the coordinated interval lasts long enough to also include the transmission of the second object after $t = 150.81$.

Figure 9 shows the overall percentage of the data volume being transmitted on coordinated resources for coordination variant A. The simulation result corresponds to the analytic result in Fig. 6. The discontinuity around $T_{\text{setup}} = 20\,ms$ is a result of the round-trip-time being equal to T_{setup}. In this case, the probability that a response object partially falls in the first coordination interval after connection setup is higher. Nevertheless, for coordination delays around 20 ms, the effective coordination gain is reduced to 80% for 1 Mbps and 50% for 10 Mbps average per user rate. Hence, it can be concluded that coordination delay has significant influence on the effective coordination gains of variant A for moderate to high per user average transmission rates.

2) Transmission delay: We measure the additional delay introduced by variant B in terms of the *transaction finish time*. We define the transaction finish time as the time required to completely download a web page, which is the duration from the web page request of the client until the delivery of the last embedded object of the corresponding web page, measured at the client side. Figure 11 shows the difference and the ratio between the transaction finish times for variant B and for variant A. It can be seen that the overall delay is substantially larger than the coordination setup time $T_{\text{setup}} = 10\,\text{ms}$. This is due to the TCP slow start and the structure of the web page downloads. A web page download consists of multiple request-response actions. Depending on the gaps in between and depending on the chosen coordination interval length T_{int} and coordination stop time T_{stop}, a coordinated transmission might have already been stopped before the download of the web page is complete. The downloads thus suffer from multiple coordination setup delays, which increase the RTT seen by TCP. This in-turn leads to a slower increase of the congestion window size as compared to variant A and further slows the downloads down.

Figure 11 shows that the ratio is largest for small transaction sizes and gets smaller as transaction size increases. For modern highly interactive web applications (e.g. web mailer, web office applications, gaming etc.), the additional delay for small messages might result in a less responsive behavior of the application and thus in a reduced quality of service. For large transaction sizes, the ratio is smaller but the absolute difference is in the order of several hundred milliseconds, which might well be noted by a user. A possible way to improve the transmission delay of variant B is to avoid buffering of small TCP packets, such as acknowledgments and SYN packets.

3) Coordination overhead: Coordination usually generates overhead on backhaul links. Depending on the coordination scheme, overhead might also be generated on the air interface, if additional measurements of the mobile are required. In addition to this overhead, agreements between eNodeBs on time/frequency resources and antenna configurations put

---

[3] Due to a lack of up-to-data traffic measurements in wireless networks, we currently cannot verify this assumption and therefore plan to repeat the experiment with other trace data as well.

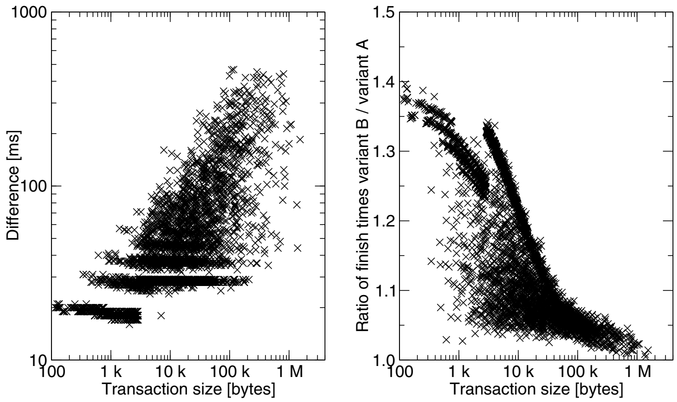


Fig. 11: Transaction finish times of variants A and B

constraints on local scheduling decisions. If, for instance, a coordinated transmission was scheduled for a certain user i and this user does not have any data to transmit anymore, the same resources cannot just be used for another user j, because j might require other antenna settings, violating the constraints that have been agreed on in the coordination process. Whether it is possible to use the resources planned for user i for another user or not depends on the distribution of users in the cell, their radio channel characteristics and their current buffer fill levels. Hence, the cost of these unnecessarily constrained resources is unknown and its quantification requires further research.

Nevertheless, to assess the overhead of these constrained resources in our model, we count the number of TTIs that were planned for coordinated transmission to a certain user, but not used because the user queue is empty. Fig. 10 shows the number of TTIs with coordinated resources that were used for data transmission vs. the number of used and unused coordinated TTIs. For variant A, the shape of the curves obviously follows the shape observed in Fig. 9. Although the fraction of coordinated traffic is 1 for variant B, the coordinated TTI usage ratio is similar to variant A. The ratio declines with increasing coordination delay and with increasing data rate. The reason is the way how coordinated transmission is stopped: as soon as the user queue runs empty, the base station stops the coordination process with its neighbor base stations. However, due to the time $T_{\text{stop}} = T_{\text{setup}}$, the coordination will not stop immediately, but continue for at T_{stop} plus the remaining coordination interval length. Clearly, a more intelligent coordination scheme, which does not just consider the current buffer fill level of a user but extrapolate the time at which the buffer will run empty, could stop the coordination process earlier and thus improve this ratio.

VI. CONCLUSION

Our analysis revealed that traffic characteristics can have significant impact on the achievable higher layer gains of various Coordinated Multipoint (CoMP) transmission schemes, especially if coordination setup time is large. Obviously, coordination gains can be preserved if data is buffered until the coordination process among the eNodeBs is complete. However, as we have shown, this might result in a degradation of the quality of service of delay-sensitive applications. Consequently, user traffic characteristics and quality of service requirements should be taken into account in the decision about whether and how to use CoMP schemes for a given user data traffic.

We consider our work to be a first step to understand possible interactions between Internet data traffic characteristics and CoMP algorithms. Understanding these interactions is vital to ensure that costly investments in CoMP-enabled infrastructure actually pay off and that applications can really benefit from higher transmission rates provided by CoMP. The algorithms on how to initiate a coordinated transmission that were presented here are quite simplistic approaches. There is still a lot of room to reduce coordination overhead and to improve higher layer performance. Our future work and hopefully also work of other researchers will continue in this direction.

ACKNOWLEDGMENT

The author wants to thank my colleagues Christian Blankenhorn and Thomas Werthmann for their continuous support on the simulation environment and the NSC.

REFERENCES

[1] 3GPP TR 36.814, "Further advancements for E-UTRA physical layer aspects," 3GPP Technical Report, Release 9, Mar. 2010.

[2] M. Sawahashi, Y. Kishiyama, A. Morimoto, D. Nishikawa, and M. Tanno, "Coordinated multipoint transmission/reception techniques for lte-advanced," *Wireless Communications, IEEE*, vol. 17, no. 3, 2010.

[3] V. Jungnickel, L. Thiele, T. Wirth, T. Haustein, S. Schiffermuller, A. Forck, S. Wahls, S. Jaeckel, S. Schubert, H. Gabler, C. Juchems, F. Luhn, R. Zavrtak, H. Droste, G. Kadel, W. Kreher, J. Mueller, W. Stoermer, and G. Wannemacher, "Coordinated multipoint trials in the downlink," in *IEEE GLOBECOM Workshops*, 2009.

[4] P. Marsch and G. Fettweis, "On downlink network mimo under a constrained backhaul and imperfect channel knowledge," in *IEEE GLOBECOM 2009*, 2009.

[5] R. Ballentin and M. Doll, "Downlink coordinated scheduling - simulation results," ITG 5.2.4 Workshop on IMT-Advanced, available at http://www.ikr.uni-stuttgart.de/Content/itg/fg524/Meetings/2010-07-08-Heidelberg/index.html, July 2010.

[6] R. Chakravorty, S. Katti, I. Pratt, and J. Crowcroft, "Using tcp flow-aggregation to enhance data experience of cellular wireless users," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 6, 2005.

[7] NGMN Alliance, "Radio access performance evaluation methodology," available online at http://www.ngmn.org/, June 2007.

[8] B. Mah, "An empirical model of http network traffic," in *Proceedings of IEEE INFOCOM '97*, vol. 2, 1997.

[9] California Institute of Technology, Networking Lab, "Traffic traces for tcp evaluation," available online at http://wil.cs.caltech.edu/suite/TrafficTraces.php, 2008.

[10] F. D. Smith, F. H. Campos, K. Jeffay, and D. Ott, "What tcp/ip protocol headers can tell us about the web," in *Proceedings of the 2001 ACM SIGMETRICS conference*, USA, 2001.

[11] F. Hernandez-Campos, "Generation and validation of empirically-derived tcp application workloads," Ph.D. dissertation, University of North Carolina at Chapel Hill, Departement of Computer Science, 2006.

[12] R. Pries, F. Wamser, D. Staehle, K. Heck, and P. Tran-Gia, "Traffic measurement and analysis of a broadband wireless internet access," in *IEEE 69th VTC Spring*, 2009.

[13] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, "Host-to-host congestion control for tcp," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 3, 2010.

[14] S. Jansen and A. McGregor, "Simulation with real world network stacks," in *Simulation Conference, 2005 Proceedings of the Winter*, 2005.