International Workshop on Telematics, 17.-21.9.89, Denver, USA

Performance Evaluation of Protocol Mechanisms in an Internetworking Environment

Martin Bosch
University of Stuttgart
Institute of Communications Switching and Data Technics
Seidenstraße 36, D–7000 Stuttgart 1
Federal Republic of Germany

In an internetworking environment Interworking Units (IWUs) frequently represent bottlenecks of the distributed system due to their centralized position and the number of protocol entities, which must be handled by the processors of each IWU. Therefore, it is inevitable to optimize all adjustable parameters in protocol entities of IWUs. The influence of several protocol mechanisms on the performance of communication relationships via an IWU will be investigated in this paper. These protocol mechanisms are necessary for speed and packet size adaptation and may occur in various combinations with complex interactions, in order to contribute to a complete protocol conversion.

1. Introduction

During the recent years, most computer vendors have solved the problem of interconnecting their products using an adequate *Local Area Network* (LAN). Today, we have to take care of possibilities to interconnect these networks which frequently use different protocols with each other or with *Wide Area Networks* (WANs). Therefore, IWUs as shown in Figure 1 are necessary to provide the adaptation of the protocols involved, which may include adequate protocol mechanisms. IWUs may be classified according to the coupling layer at which the protocol conversion is done. In particular, if it is located at the network layer of the *Basic Reference Model* for *Open Systems Interconnection* (OSI) [4], the term *router* is usually used, and if it is located at the transport layer or above the IWU is called a *gateway.* An interconnection on lower layers with the help of *bridges* or *repeaters* is no main focus of this paper. The protocols in all layers above the coupling layer of the IWU must be identical in each of the interconnected networks due to their end–to–end significance, whereas the remaining protocols work station–to–station and may therefore be different. An example for an application layer gateway, a *MAP–Gateway* (Manufacturing Automation Protocol – Gateway), has been investigated and implemented. Its architecture and a performance evaluation have been presented in [1].

The effect of several protocol mechanisms in selected entities of Figure 1 on the performance of communication relationships via an IWU is subject of the following sections. It has been investigated with the help of an event by event simulation technique, as described in [2].

An isolated consideration of flow control mechanisms has been presented by Mühlenbein et. al. for DATEX–P in [5] and by Yamamoto et. al. for X.75 in [8]. Other protocol mechanisms have partly been considered by Vazquez et. al. and by Pach et. al. in [6] and [7], but not for interconnected networks via IWUs.

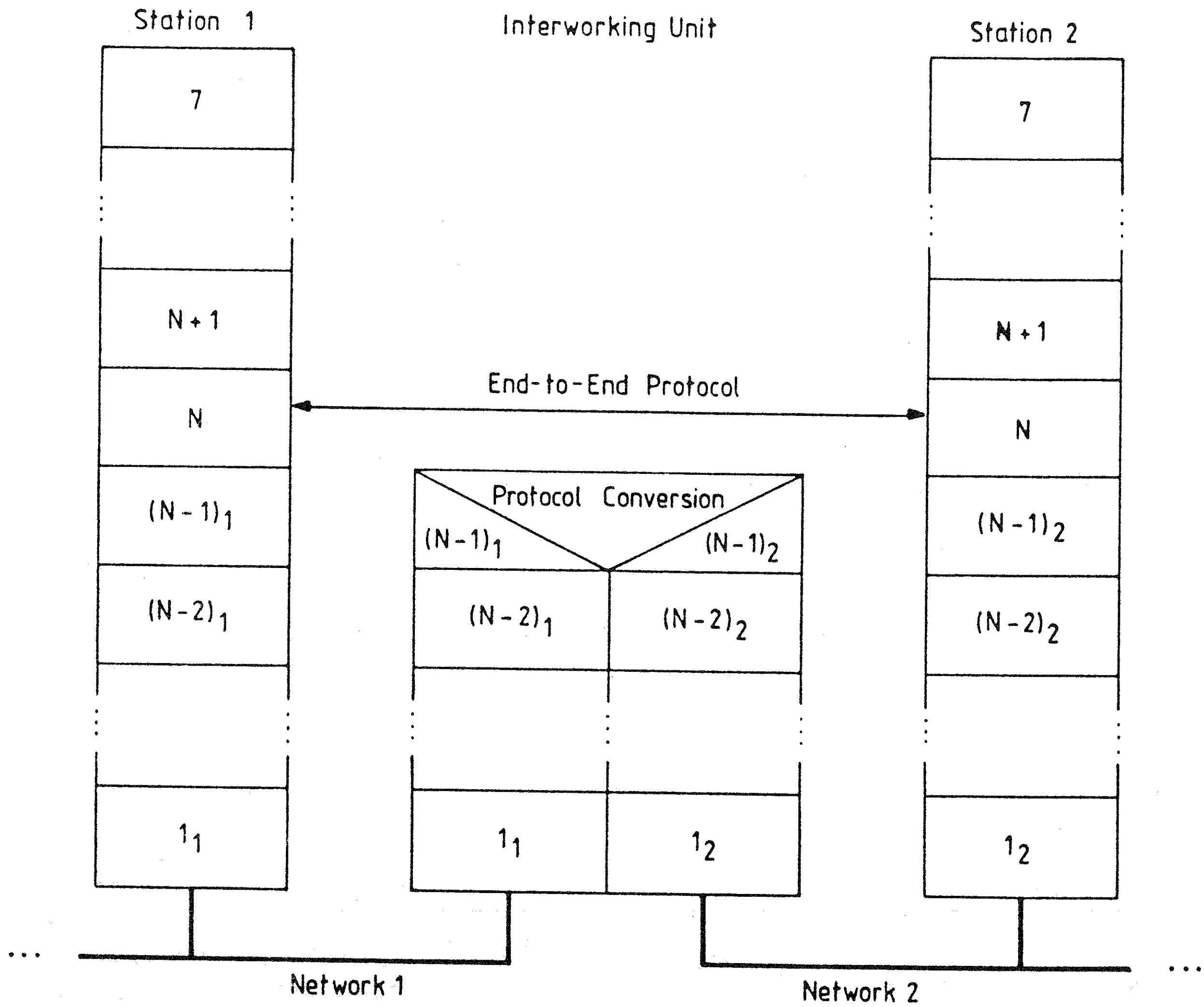

Figure 1 : Communication Relationship via an Interworking Unit

2. Characteristics and Models of Protocol Mechanisms

According to [4], various protocol mechanisms are allowed or even prescribed at each layer of the Basic Reference Model. Models, which are suitable for the use within a simulation program, have been developed for each protocol mechanism. The most interesting ones are depicted in Figures 2 to 6 for half duplex traffic.

The adaptation of speed mismatches in consecutive connection oriented protocols can be done by *multiplexing* or *splitting* of connections for the next lower layer. At the peer entities *demultiplexing* and *recombining* must be performed, respectively. In particular, if a flow control is used at a lower layer, significant effects may be expected due to a better or worse utilization of the window size.

Another possibility to compensate different speeds are adequate flow control mechanisms like *sliding window protocols* with sequence number control or pacing. The acknowledgement overhead of the sliding window protocol can partly be compensated by using the *piggybacking* mechanism, which means, that acknowledgements may be loaded on packets for the reverse direction. These acknowledgements must therefore be delayed to enable piggybacking. If a maximum delay is reached, a timer expires and a pure acknowledgement packet must be sent. The sequence number control of the sending station usually would start a timer for each packet being sent, which would be stopped upon receipt of the correct acknowledgement. If a timer would expire, the corresponding packet would be repeated immediately. In the simulation program no losses may occur, and therefore the time supervision is not necessary in this implementation.

In some cases a *sequencing* mechanism, as modelled in Figure 2, must be taken into account, especially after the recombining of connections with different speeds.

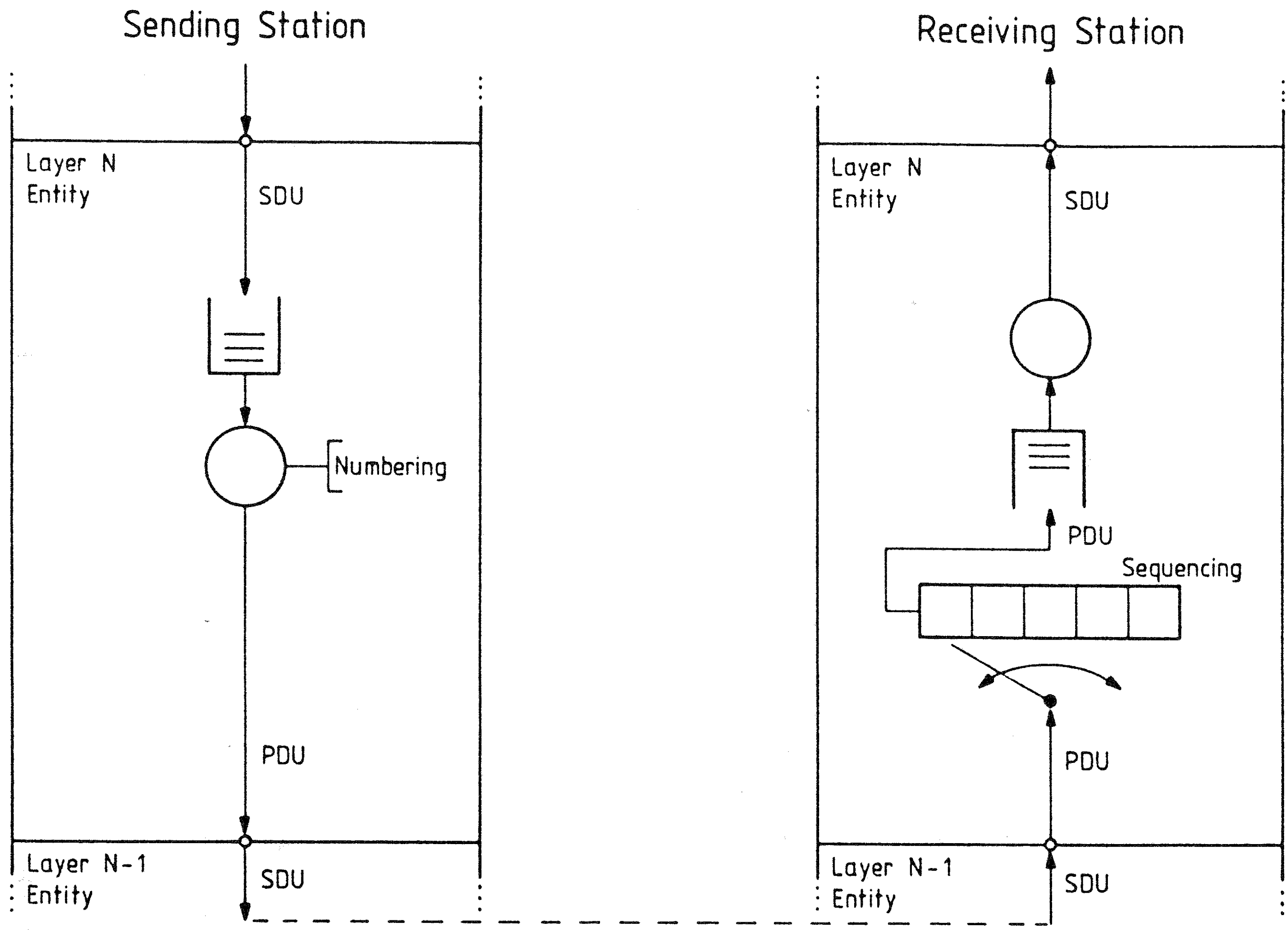

Figure 2 : Modelling of Sequencing

Different packet sizes must be adapted by adequate protocol mechanisms. The corresponding reverse protocol mechanisms have to be located at the peer entities.

Concatenation of multiple *Protocol Data Units* (PDUs) to one *Service Data Unit* (SDU) of the next lower layer may be used to improve the utilization of the window size of a lower layer flow control and to reduce the necessary processing time on these lower layers. Its reverse protocol mechanism is *separation*. PDUs have to be delayed for some time to allow a concatenation with subsequent PDUs, as depicted in Figure 3. If a maximum number of PDUs has been reached, or if the concatenation timer expires, the SDU for the next lower layer may pass its *Service Access Point* (SAP), which is drawn as a circle in Figure 3.

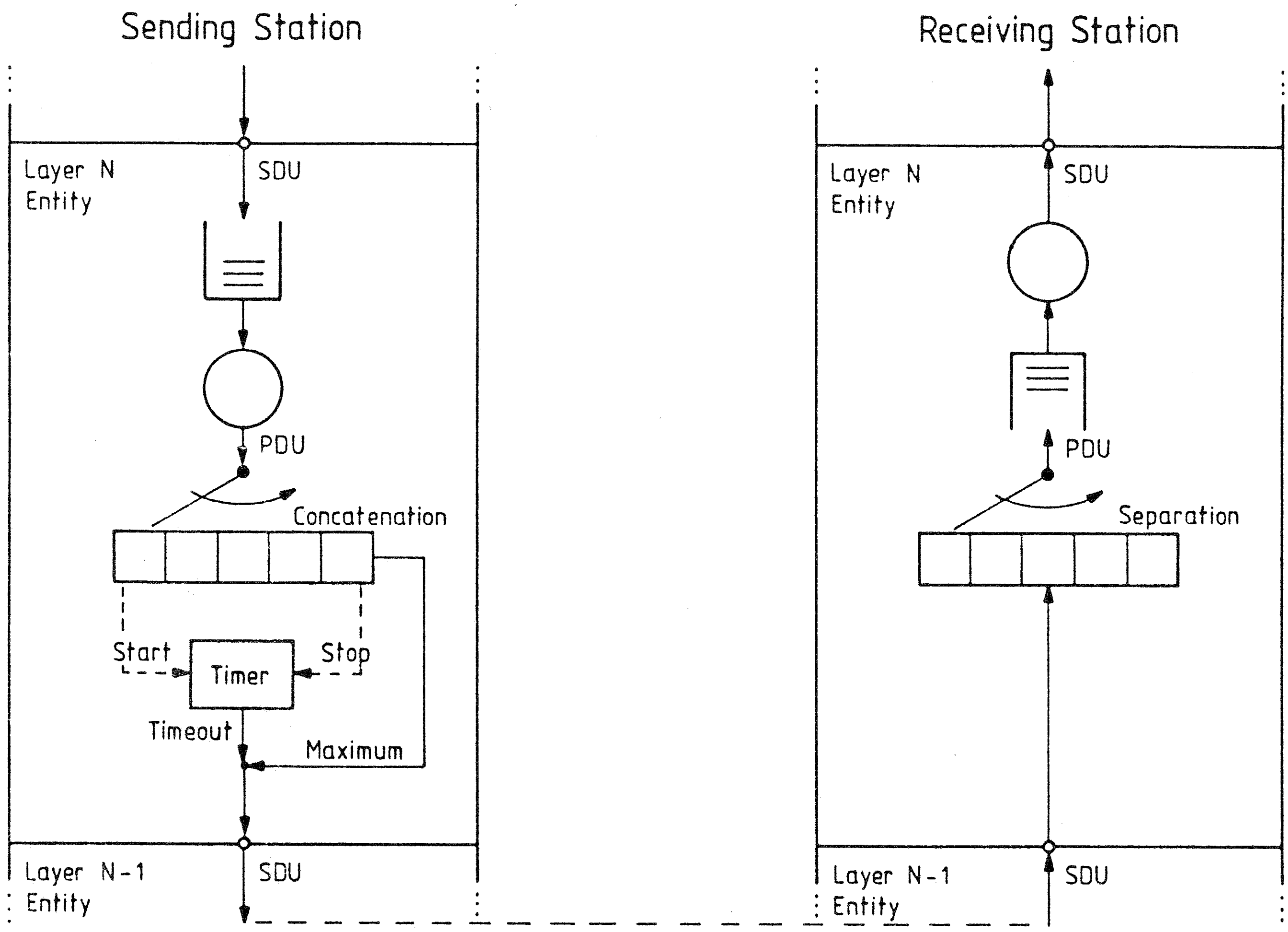

Figure 3 : Modelling of Concatenation and Separation

Blocking of multiple SDUs to one PDU has the same advantages as concatenation concerning the lower layers, but additionally for the considered layer itself. Its peer mechanism is *deblocking.* Blocking may be modelled and implemented similar to concatenation, as depicted in Figure 4.

Due to the results presented in section 4 (Figure 9), it seems to be more useful, at least concerning the transfer time, to implement blocking as modelled in Figure 5. In that implementation variant, the additional delay on top of the considered layer is removed. Instead, if the sending phase of the layer N entity is selected by the scheduling algorithm of its processor, it serves all waiting SDUs up to a maximum number. Consequently, single SDUs are served more frequently, which increases the processor utilizations, but decreases the transfer time in avoiding additional delays. To improve the chance of a successful concatenation, the sending phase of the layer N entity should have the lowest priority in the related processor.

Segmenting of one SDU into multiple PDUs may be necessary to adapt mismatches of packet sizes in adjacent layers. The peer mechanism *reassembling* is done with the help of a table, in which for all arriving segments an entry is being made. As soon as one SDU is complete, a total row of the table will be deleted. The corresponding model is shown in Figure 6.

All these models for protocol mechanisms have been inserted into the dedicated layers of a simulation model for the configuration depicted in Figure 1, as intended by the Basic Reference Model. The IWU may consist of three or four layers (router or gateway). In setting some processor phases constantly to zero, it is also possible to simulate a bridge. The data link layer has been subdivided into the *Media Access Control* (MAC) and *Logical Link Control* (LLC) sublayers. The application systems of stations 1 and 2 are represented by a set of traffic generators.

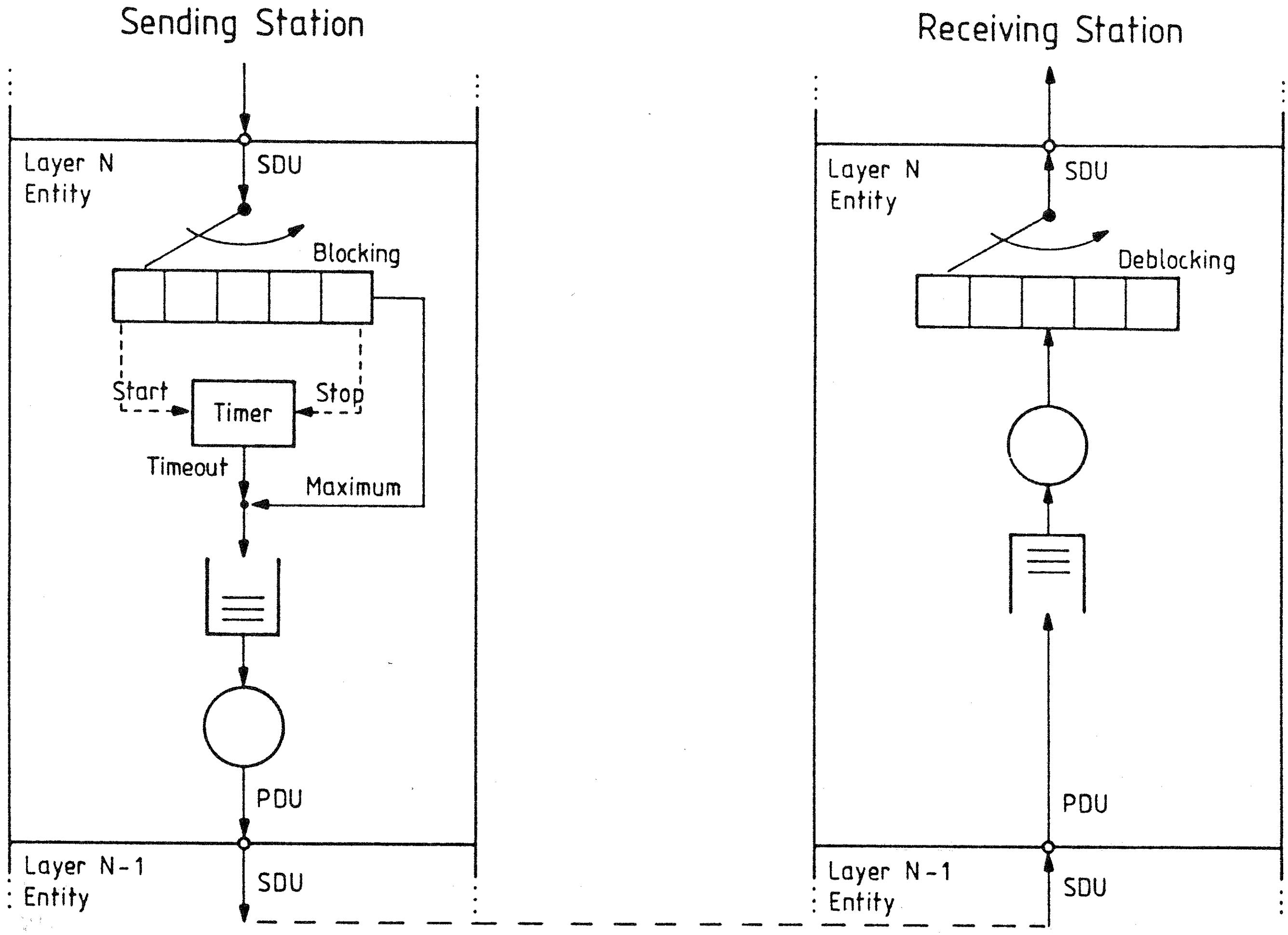

Figure 4 : Modelling of Blocking and Deblocking (Variant 1)

3. Implementation Aspects

The simulation program has been implemented in PASCAL on a VAX computer under the operating system VMS. Some standard modules of a simulation library for event by event simulations could be used. Dynamic data structures are central components of this simulation program. They allow an efficient implementation of all protocol mechanisms, avoid unnecessary transfers of data and therefore minimize computing time.

Communication relationships via the IWU may either be connectionless or connection oriented. The entities within each station are related to one or more processors. Various priorities may be assigned to connections and layers. Each entity adds its *Protocol Control Information* (PCI) to the received SDU to complete the PDU of that layer, and the peer entity will remove this PCI after its analysis. All allowed protocol mechanisms according to [4] may be activated or deactivated for each layer per input file.

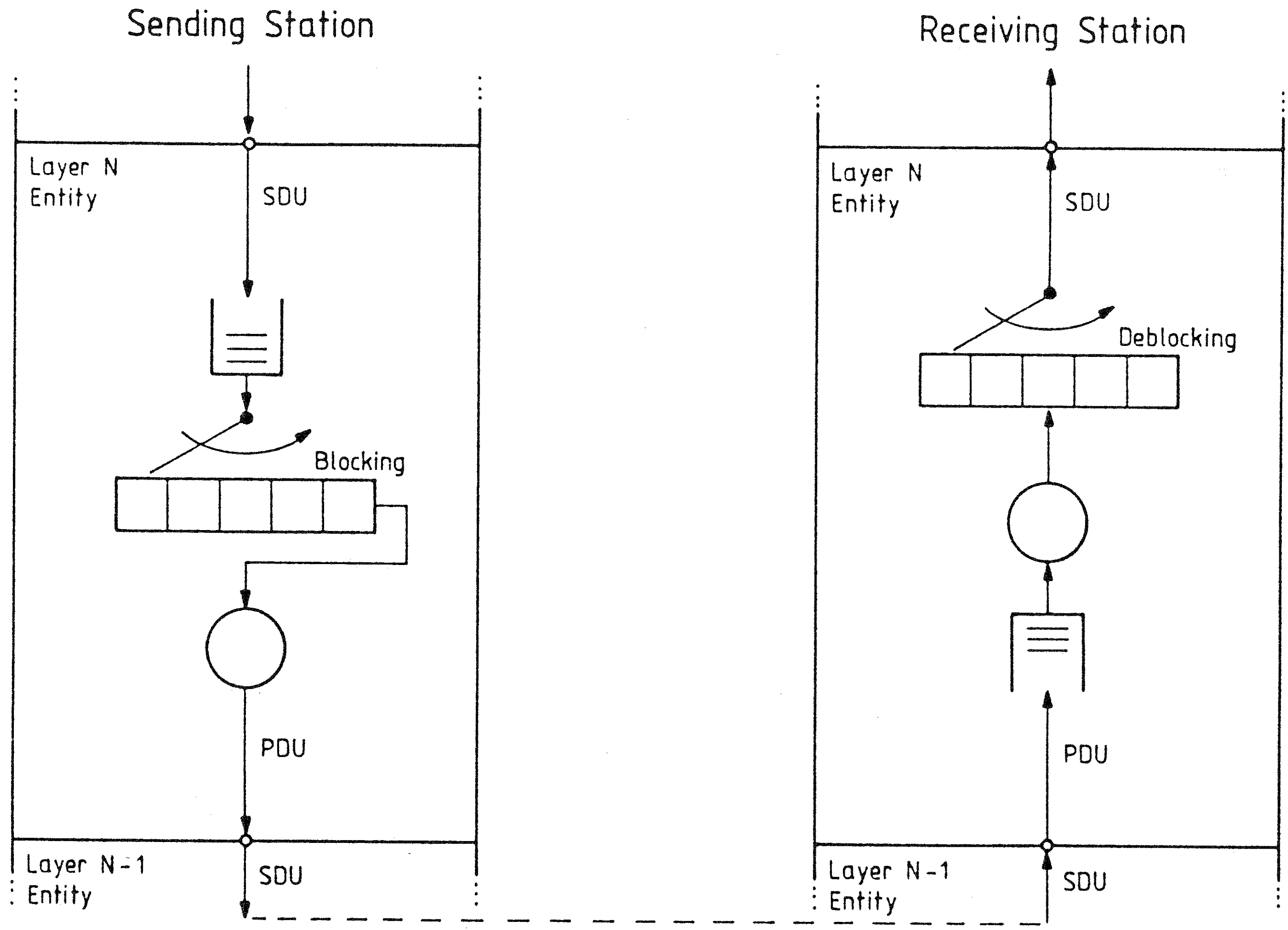

Figure 5 : Modelling of Blocking and Deblocking (Variant 2)

4. Performance Evaluation

Some representative simulation results are presented in this section. They have been obtained for a service time of 3 ms per packet in each entity and with an overhead time of 1 ms between two successive service times of one processor, representing task switches or other functions of the operating system. The transmission time on the medium between two stations has been assumed to be 3 ms. The transport protocol has an end–to–end significance in the simulated configurations. Layers 1 and 2a are assumed to be handled by a dedicated communication controller in each station. Layers 2b to 4 are usually implemented in firmware and run on a processor on the communication boards in stations 1 and 2. In the IWU (router) it is assumed, that two communication boards are used, which do not include the transport layer. The protocol conversion itself is done at the host processor. Priorities increase with the consumed processing time of each packet.

A comparison of transfer times (time from the transport SAP in station 1 to the transport SAP in station 2) for various flow control variants with sequence number control is depicted in Figure 7. The flow controls cause the boarders of stability to be below 50 packets per second, in contrast to Figures 8 to 11. Due to no possibility of stopping the traffic of station 1 in case of congestion of network 2, independent flow controls at the network layer should be avoided. Therefore, interconnected flow controls at the network layer represent an optimum. The window

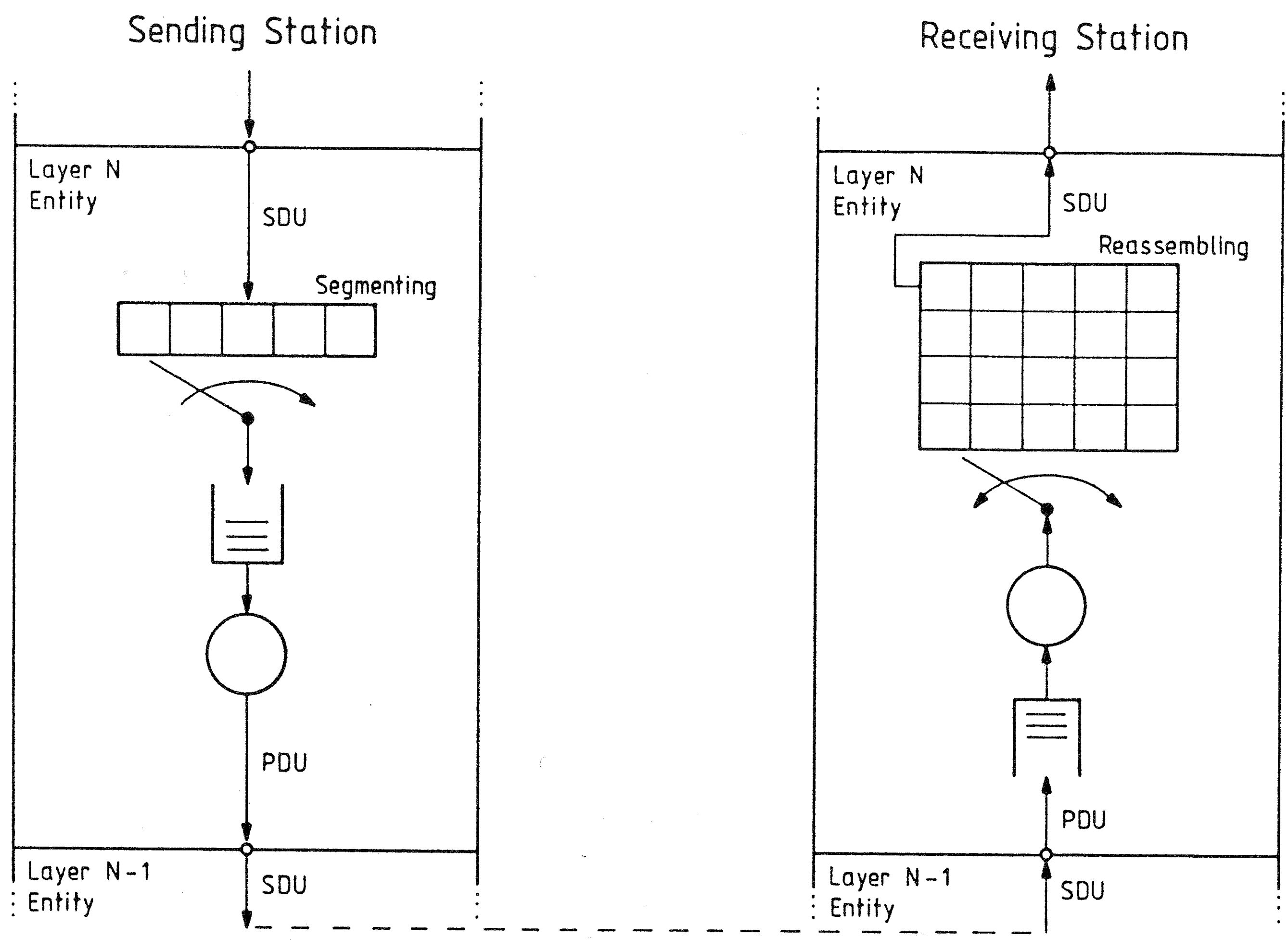

Figure 6 : Modelling of Segmenting and Reassembling

size in network 1 should be twice as high as the window size in network 2 because of a kind of end–to–end significance. An end–to–end flow control at the transport layer also needs a window of double size, due to the increased roundtrip delay via two networks, but it nevertheless leads to a loss of performance because of an increased overhead in processing acknowledgements at the transport layer entities. An end–to–end flow control is unavoidable, if the protocol conversion is done at the MAC sublayer, because of the first available flow control mechanism at the LLC sublayer. In this case a dynamic window size, as suggested in [3], should be used to avoid congestion in the bridge by throttling down the source. The worst performance has been observed for a combination of several flow controls due to the flow control of transport acknowledgements at the network layer in both networks.

Figure 8 shows the utilization decrease of the processor serving layers 2b to 4 in station 2, if blocking (controlled by a blocking timer, according to Figure 4) is used at the network layer of network 2. The maximum block size has been assumed to be five SDUs. The blocking timer value represents the maximum delay of the first arrived packet of a block. The processor utilization decreases with an increasing maximum delay because of the decreasing number of packets to be served. The minimal interarrival time of SDUs at the blocking mechanism in the IWU is 11 ms in the selected configuration. Therefore, no blocking is observed in the range from 0 ms to 11 ms of

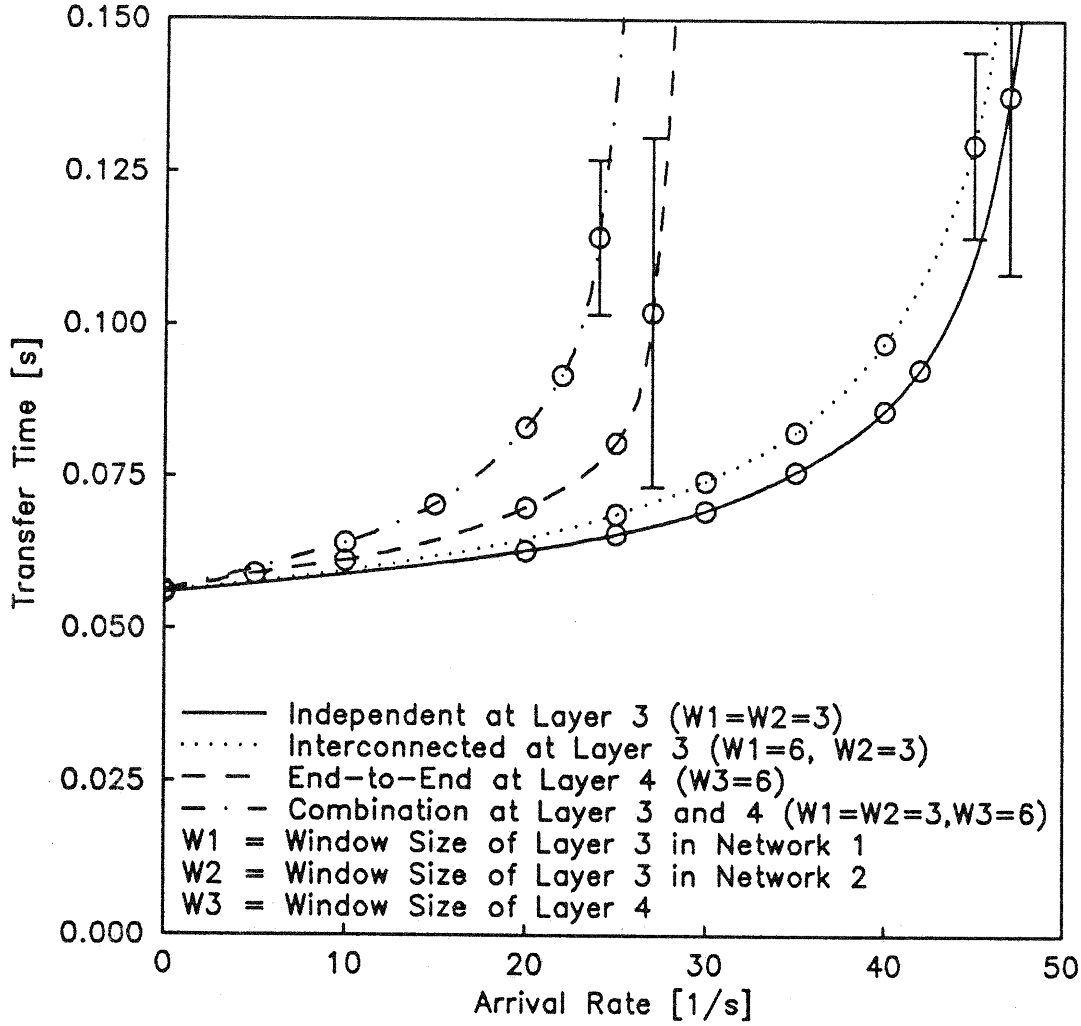

Figure 7 : Transfer Time for Various Flow Control Variants with Sequence Number Control

the blocking timer value. The blocking of three SDUs can only occur for a blocking timer value of at least 22 ms. Analogous arguments hold for the blocking of four or five SDUs. The probability for blocking increases with the arrival rate. Therefore, the expected effects for the blocking timer values of 11 ms and 22 ms are especially visible for the high arrival rate of 75 packets per second.

In Figure 9 it can be seen, that the additional blocking delay of this implementation variant 1 always dominates over the reduced waiting times in lower utilized processors, if the transfer time is considered. However, the blocking delay reduces with an increasing arrival rate. The reason of this behaviour is the fact, that for the considered configuration the bottleneck is not station 2 due to the reduced number of packets there, in combination with the nonexisting flow control in network 2. The simulation of a blocking implementation according to variant 2 (see Figure 5) yields the same results as shown in the curve without blocking in Figure 9, due to a lower cycle time of the corresponding processor, compared to the minimal interarrival time there. Blocking according to variant 2 will only occur, if it is done at a bottleneck processor.

Figure 10 shows the transfer time for an activated concatenation at the network layer of network 1 with a maximum number of five concatenated PDUs. Additionally, at the LLC sublayer of

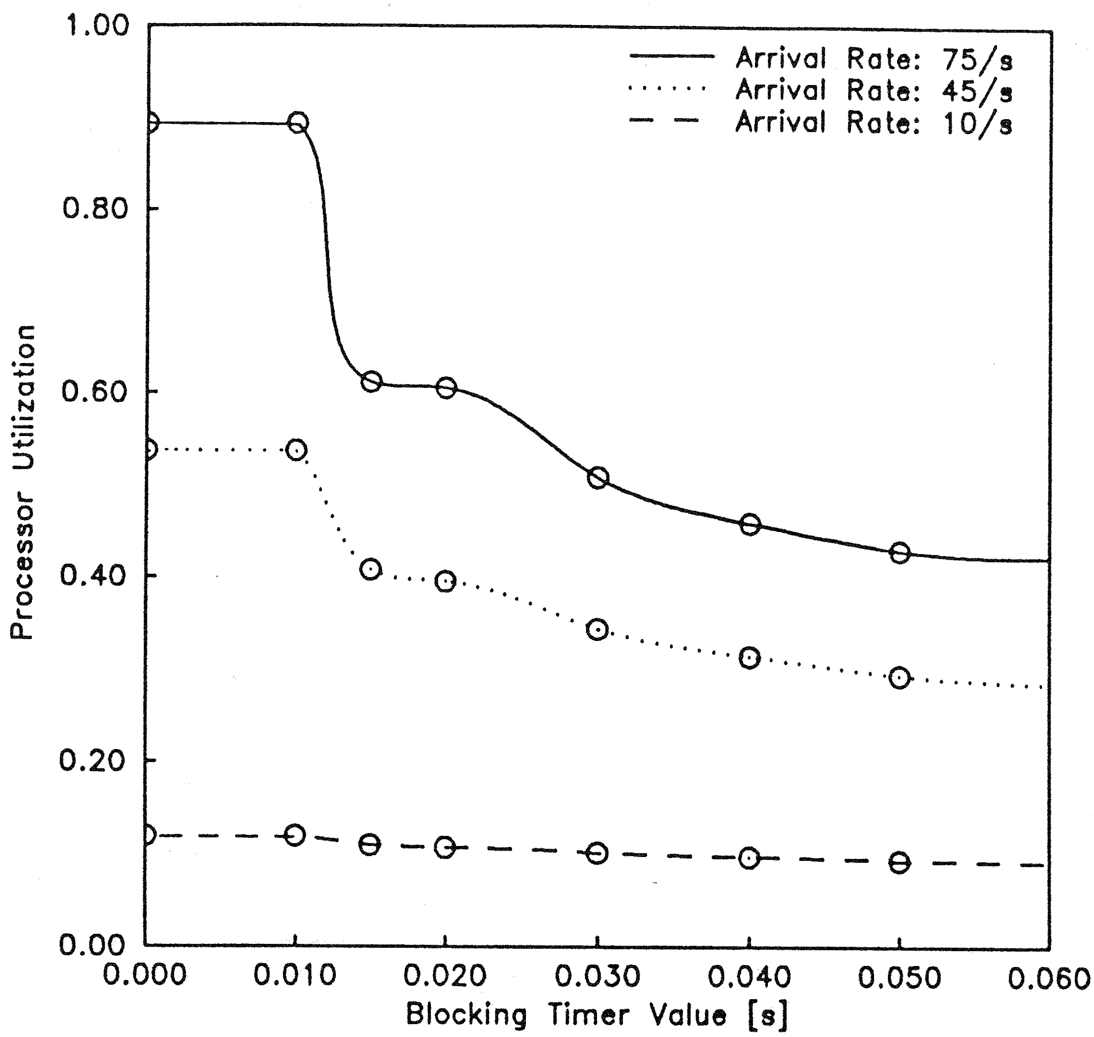

Figure 8 : Processor Utilization in Station 2 for Blocking at the Network Layer of Network 2

network 1 a flow control with a window size of two is used. For low arrival rates, the same effect as in Figure 8 can be observed: the transfer time is increased by the concatenation delay. For a high arrival rate, the bottleneck processor in station 1 and the flow control in network 1 lead to an improvement of the transfer time by the use of concatenation.

Segmenting at the network layer of network 2 leads to an increased transfer time and a lower boarder of stability because of the increased number of packets in network 2, as depicted in Figure 11. Therefore, segmenting should be used on the lowest possible layer, to limit the number of involved protocol entities. Additionally, the reassembly delay, which is necessary to reassemble all PDUs belonging to the same original SDU, contributes to the increased transfer time. Furthermore, segmenting leads to batch arrivals at the processing phase of the same entity, which is also disadvantageous for the behaviour of the system. The last two disadvantages can be avoided, if station 1 only produces packets of a minimal size, and therefore prevents segmenting in the IWU. This may be used, if network 1 is no bottleneck and the increase of the arrival rate can be handled there without any problem. However, the situation will be totally different, if a flow control is used on a lower layer of network 2.

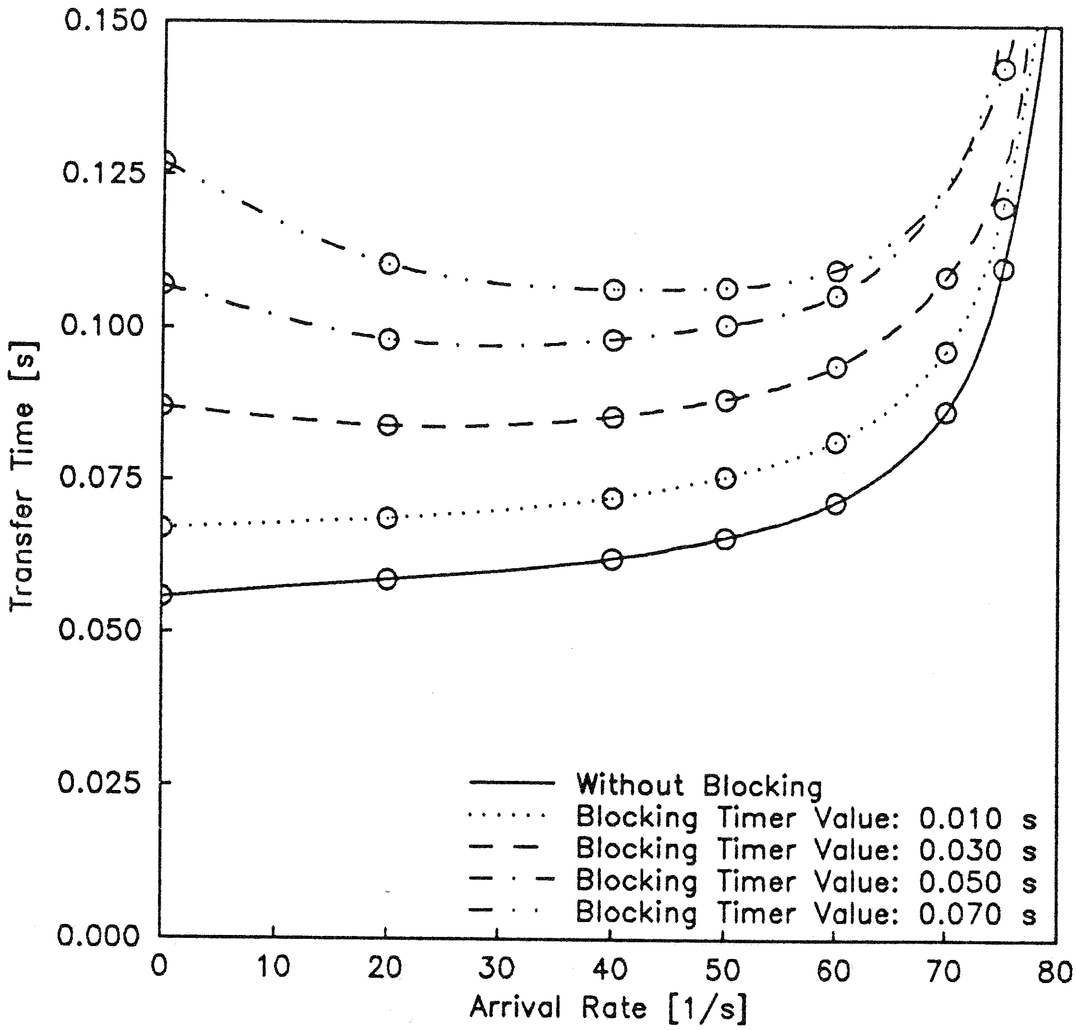

Figure 9 : Transfer Time for Blocking at the Network Layer of Network 2

5. Conclusion

A simulation program has been developed as an adequate tool to study the appropriate ranges of many adjustable parameters of protocol mechanisms, and to predict the performance to be expected. The optimal combination of protocol mechanisms in an internetworking environment can be determined for many specific interconnection problems.

Acknowledgement

The author would like to thank Dietmar Horst, Thomas Ohr and Michael Weinmann for the implementation of the simulation program.

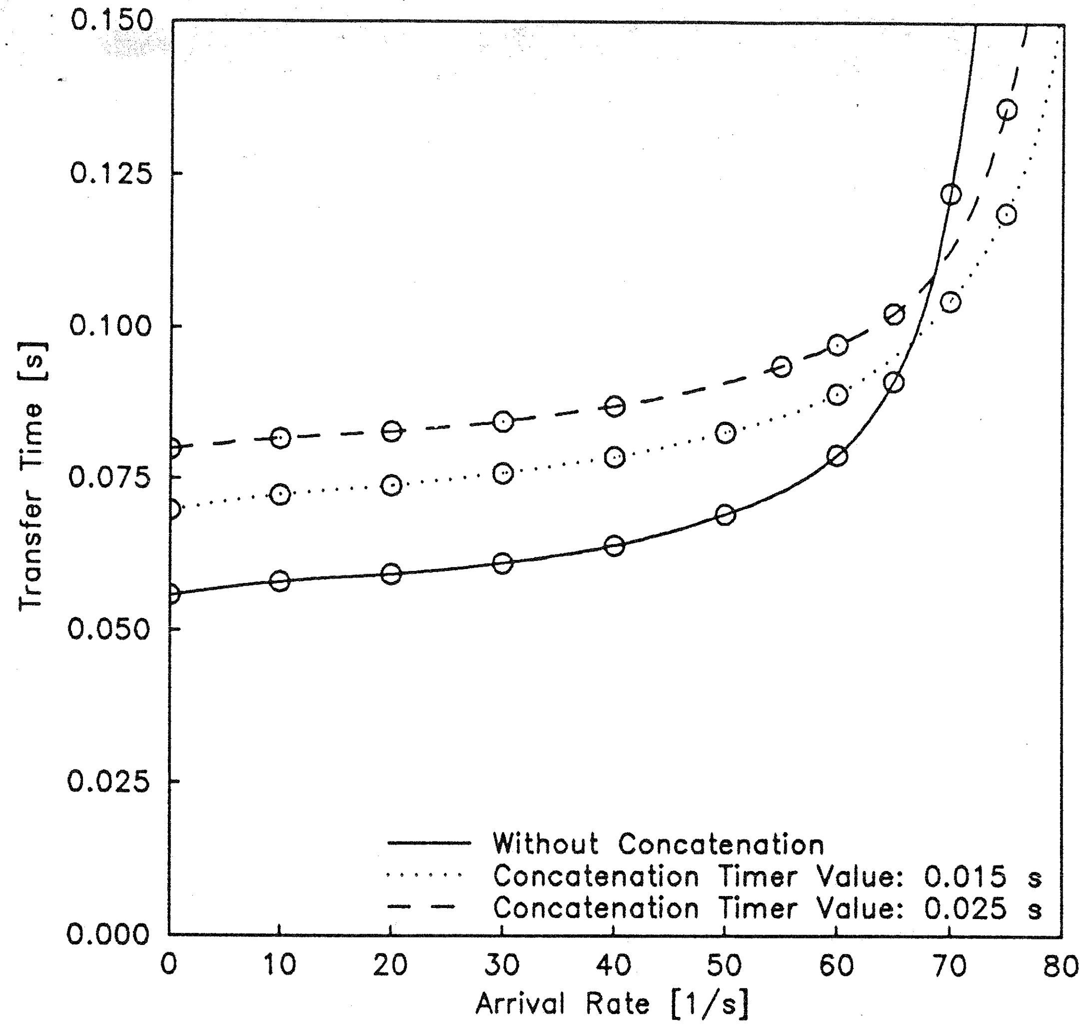

Figure 10 : Transfer Time for Concatenation at the Network Layer of Network 1

References

[1] M. Bosch, O. Gihr; "Architecture and Performance Evaluation of a MAP–Gateway", Proceedings of The Sixth European Fibre Optic Communications and Local Area Networks Exposition, Amsterdam, June 29 – July 1, 1988, pp. 354 – 360

[2] M. Bosch, O. Gihr; "Simulation of Application Layer Protocols for Factory Automation", Proceedings of The Ninth International Conference on Computer Communication, Tel Aviv, October 30 – November 4, 1988, pp. 320 – 328

[3] W. Bux, D. Grillo; "Flow Control in Local–Area Networks of Interconnected Token Rings", IEEE Transactions on Communications, October 1985, Volume 33, No. 10, pp. 1058 – 1066

[4] ISO 7498; "Information Processing Systems – Open Systems Interconnection – Basic Reference Model", November 1983

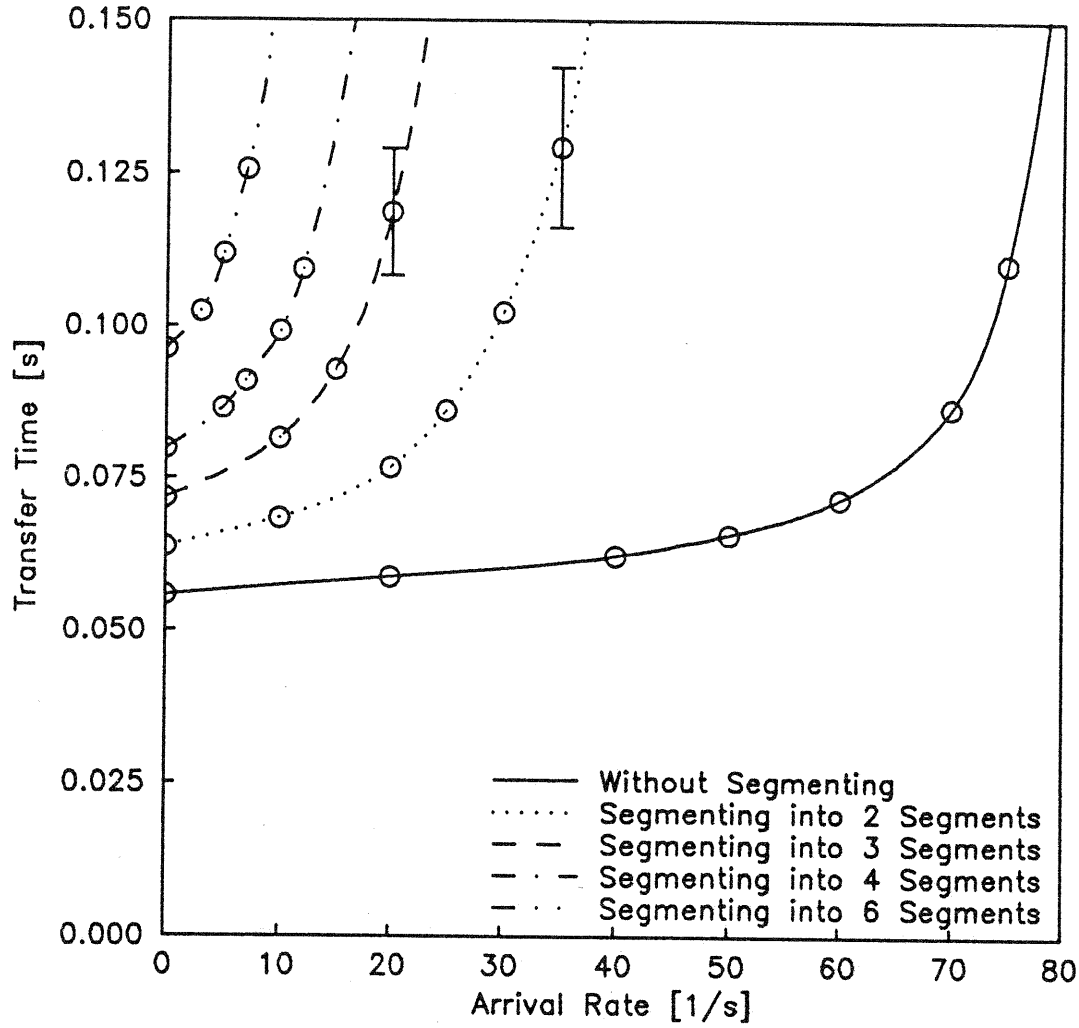

Figure 11 : Transfer Time for Segmenting at the Network Layer of Network 2

[5] H. Mühlenbein, F. Limburger; "Simulation of Flow and Congestion Control Algorithms", Proceedings of Performance of Computer Communication Systems '84, pp. 191 – 204

[6] A. Pach, P. Brown, B. Delosme; "Performance Analysis of the Class 4 Transport Protocol", Proceedings of the 12th International Teletraffic Congress, Torino, June 1 – 8, 1988, Volume 5, Paper 5.3B.5

[7] E. Vazquez, J. Vinyes, R. Colella; "Simulation of the X.224 Transport Protocol over X.25 Networks", Proceedings of the 12th International Teletraffic Congress, Torino, June 1 – 8, 1988, Volume 5, Paper 5.3B.3

[8] M. Yamamoto, I. Akiyoshi, H. Nakanishi, H. Sanada, Y. Tezuka; "Performance of Window Flow Control Scheme for Interconnected Packet Networks", Proceedings of the IEEE International Conference on Communications '88