Annales Univ. Sci. Budapest., Sect. Comp. **9** (1988) 151-170

SIMULATION STUDY OF THE CCS NO.7 PROTOKOL SPECIFICATION IN CONGESTED SITUATIONS

DIETER HOGREFE

Abstract. This paper reflects a complete study of the CCS no.7 protocol in congestion situations. It gives an example of a simulation of a nontrivial problem associated with a complex system. Therefore the basic model design method and the model itself are presented so that it can serve as an example to anybody who has to do with similar studies. Also some new results about the congestion behaviour of the CCS no.7 protocol are presented. The results were achieved by a three years project which the author was responsible for at a large manufacturer.

1. Introduction

The CCS no.7 protocol /CCa 85/ is recommended by the CCITT as an internationally standardized general purpose common channel signalling system. It is especially suitable for operation in digital telecommunication networks with stored program control exchanges. It provides a reliable means for transfer of information in correct sequence and without loss or duplication. A main application of the protocol is in the field of modern telephone switching networks.

An important aspect of such a complex system is its behaviour in congestion situations. The respective specifications /CCa 85/ and /1R7 85/ do not fully specify the complete congestion behaviour but leave some freedom for the implementor.

Thus there is a variety of possible ways to indicate congestion and to handle it. In this paper there are basically two congestion indication methods considered for investigation, and they are combined with various congestion handling methods. The results were obtained by simulation.

2. Model description

2.1 Basic protocol structure

The CCS no.7 protocol is constructed in a modular manner and described top-down in /CCa 85/. The following describes the modular construction of the protocol specification.

Fig.1 shows the structure of the protocol where only two signalling points are involved with one connection between them.

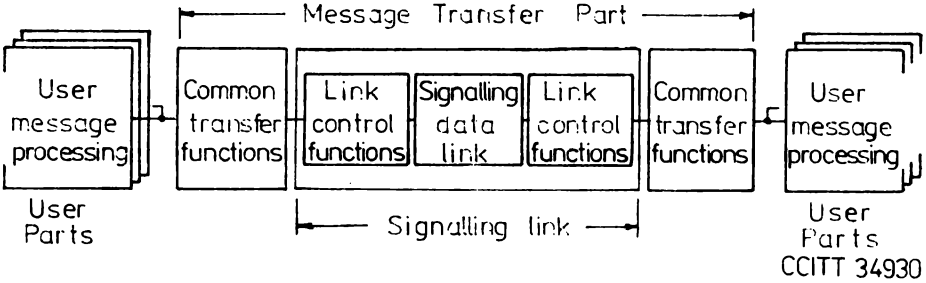


Fig.1

The protocol is decomposed into user parts (level 4) and a message transfer part (level 3-2-1). A user part is a functional unit using the message transfer part for message exchange with other user parts.

Fig.2 shows a refinement of the left part of Fig.1. It can be recognized that the message transfer part basically consists of three parts. It should be mentioned here that this specification is not a description of an implementation. Thus the borders of the blocks do not have to be physical borders in a later derived implementation. The decomposition of the basic functions into blocks is only a recommendation which can serve as a basis for the implementation. Thus the channels between blocks do not have to be physical channels. The exchange of signals may as well be made by function calls or use of shared variables.

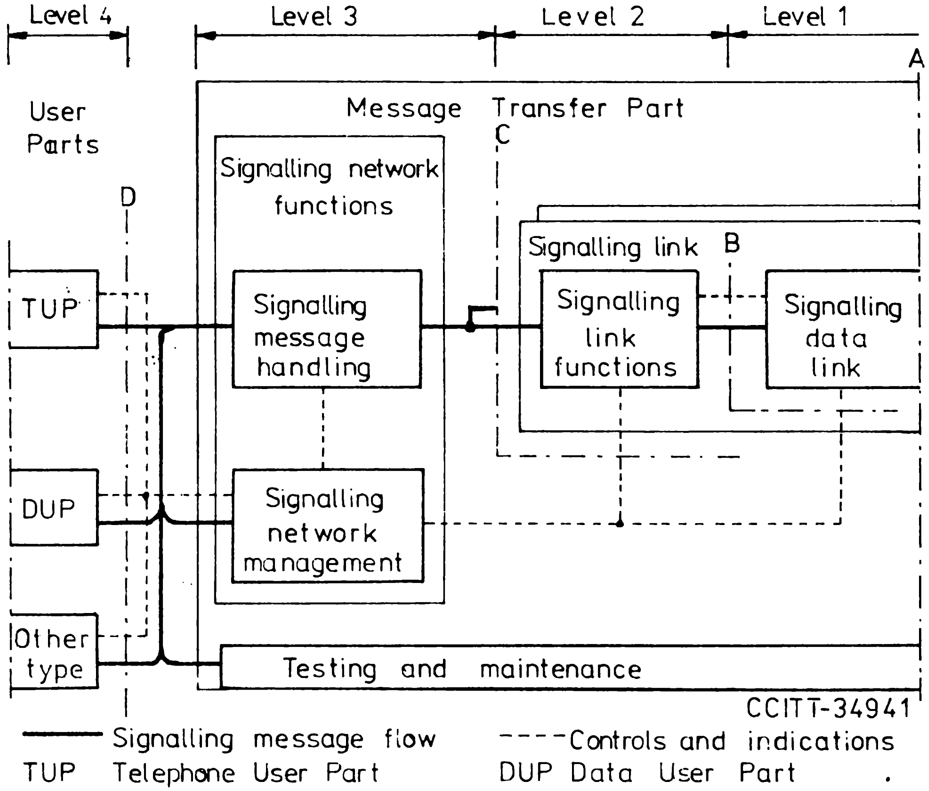


Fig.2

The following briefly describes the functions of the various levels of the protocol. Details of these functions can be found in the respective specifications /CCa 85/.

Level 1 (signalling data link functions)

Level 1 defines the physical, electrical and functional characteristics of a signalling data link and the means to access it. A signalling data link is a bidirectional transmission path for data which consisits of two data channels operating together in opposite directions at the same data rate.

Level 2 (signalling link functions)

Level 2 describes the functions and procedures that are concerned with the transmission of messages on an individual signalling data link. The level 2 functions together with the level 1 functions quarantee a reliable transfer of messages between two directly connected signalling points.

The messages are transmitted as variable length signal units. Besides the pure information for transmission, a signal unit also contains various control information for the reliable transmission.

Level 3 (signalling network functions)

Level 3 consists of those transport functions which are universal and independent of an individual signalling link. They are devided into

- signalling message handling functions and
- signalling network management functions.

The signalling message handling functions direct a message for transmission to the proper signalling link. On the other hand, they direct a recieved message to the proper user part.

The signalling network management functions control the message routing and the network facitilies on the basis of information about the network status. In case of a change of the status they control reconfiguration and other activities to reestablish the normal message flow.

Level 4 (user part functions)

Level 4 consists of various user parts, TUP (telephone user part), DUP (data user part), ISDN-user part,... . Each user part defines functions and procedures that are particular to a certain type of network user. In the case of congestion appropriate actions for traffic reduction are taken.

2.2 Basic model structure

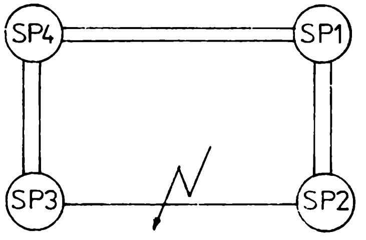
The simulation for the investigation of congestion situations was made with a simulation model consisting of four signalling points. This basic model structure is shown in Fig.3. The internal structure of a signalling point is an abstraction of the structure of the protocol as described in the previous section. It will be described later. The configuration with four signalling points was chosen as a compromise between oversimplification and making the model too complex in order to study it in appropriate time. Experience showed that the model was on the upper limit of complexity which could be handled. But it also showed that the model was sufficient to simulate situations of rerouting messages with two signalling transfer points involved. Therefore, it was possible to simulate realistic congestion situations.

Fig.4 shows the basic message flow in the model. At the beginning of the simulation SP1 sends traffic to SP3 where SP4 serves as a transfer point. SP2 sends traffic to SP3 on the direct way. After a certain amount of time the link SP2-SP3 failes and traffic has to be rerouted as shown by the dotted line in Fig.4. This results in an additional load to be handled by signalling transfer point SP4. In the simulation model SP4 was supposed to have limited capacity such that a congestion situation arises.

There were two congestion indication methods to be investigated. In both cases the model was the same. With the first method which is proposed by /1R7 85/, the congestion is recognized by the buffer contents. The other method, which performed better in some respect, recognized the congestion situation checking the processor load every once in a while. When the processor load exceeds a certain limit, e.g. 85% of its maximum capacity, certain traffic reduction measures are taken by the system. Both methods are further explained later.

2.3 Basic model design method

The signalling system no.7 is constructed in a modular manner. Each function within a network is responsible for certain situations. Only part of the functions are responsible for "normal" message flow conditions. The other functions are activated in case of exception, e.g. when a signalling link fails and rerouting of messages becomes necessary. This means that the simulation model could be implemented and tested stepwise and part by part. This common software implementation method becomes easily applicable because of the modular nature of the SDL specification of the protocol.



Simulation model

Fig.3

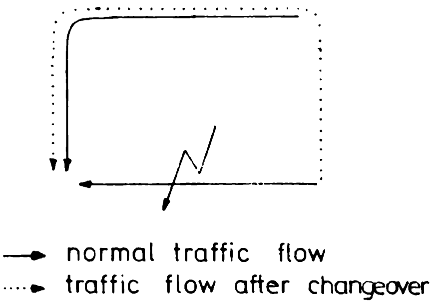


Fig.4

First step

In the first step only those functions were implemented that are responsible for the "normal" network state where the messages are routed on their predefined routes. The interfaces of these "normal" functions to the "abnormal" functions were satisfied by dummy functions which had the correct interface but no executable code.

Further steps

For the simulation of "abnormal" network situations the dummies had to be filled with functions. For each simulation experiment it had to be assured that only those functions were concerned which were already implemented in the model.

The stepwise proceeding in the model implementation had two significant advantages:

- for each simulation experiment the model was only as complicated as really necessary
- the testing was simplified

Implementation of the functions

With the specification and description language SDL the structure of a system is specified with the language element "block". A system is a block. Every block can be further divided into interconnected blocks. The leaf blocks (the bottom level of this hierarchy of blocks) are filled with processes. The interconnected blocks such as those in Fig.2 determine the structure of a system, where as the behaviour is determined by the processes. For further details about the language SDL the reader is referred to the appropriate literature /CCb 85/.

For the purpose of simulation the blocks and processes have to be incorporated into a simulation program. The translation of the SDL specification into a program was supported by the simulation system BORIS /SIE 86/ which is based on Pascal. Of course not all blocks and processes were incorporated into the program. Those functions that were not relevant for reaching the simulation goal were extracted. But the others were translated nearly one to one. The implementation of the block structure is quite straightforward. It was realized with the usual input/output relations, a feature which is supported by BORIS. The way from a process diagram to an executable Pascal code is more complex, and the idea is given in the following.

Simulation of a process

The processes are implemented as Pascal procedures. Communication is made by parameter passing. For every input signal type of a process and its data there is a call-by-value parameter in the simulating procedure. For every output signal type and its data there is a call-by-reference parameter. The data type of a parameter is a compound of the data types of the data that are passed with the signal.

The procedure body contains the actual state transition behaviour of the simulated process.

For further details /HoZ 86/ may be referenced.

Simulation of the time behaviour

There are basically three situations where time has to be expressed in the simulation model:

1) activation of a process at a certain moment in time (This is done by using the concept of a timer)

2) time consumption caused by the state transition of a process

3) time consumption on a channel caused by the transmission of a signal.

Especially the times in 2) and 3) are very hard to estimate. It is nearly impossible to give a time for every processes' state transition. The same holds for the transmission of a signal. The latter, for example, depends on how a channel is realized in a real implementation of the system. It further depends on how long a signal has to wait in an input queue of a process. /Hog 87/ processes an informal method to solve this problem by restricting the attention to some basic features.

2.4 Model structure

The basic model structure has already been explained in a previous chapter as far as the overall configuration of signalling points are concerned. Fig.5 now shows the configuration at one signalling point. The signalling point (SP) itself is coupled to the network through a special concentrator model component (CLS).

In the most general case many linksets (LS) can lead to one signalling point. In our model every signalling point has only two adjacent points. Thus the concentrator has only to concentrate two linksets. In the following the structures of the model components SP and LS are further explained.

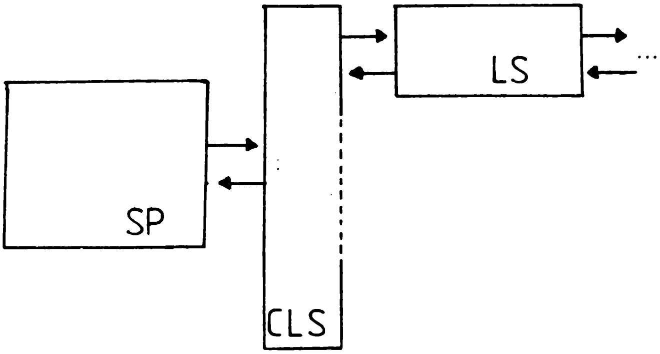


Fig.5

2.4.1 Signalling point (SP)

The model component SP consists of one level 4 component (L4) and one level 3 component (L3). L4 creates messages which are sent to L3 and recieves messages from L3. L4 simulates the functions of the level 4 as far as they are relevant for the simulation goal. L3 routes messages and takes care of some network management functions.

Level 4 (L4)

The model component L4 simulates the user part of the CCS no.7. But the user part functions, described in /CCa 85/, were as much abstracted as was possible to achieve the simulation goal. L4 consists of the components REC, GEN, SETGEN and CONG as shown in Fig.6.

REC recieves incoming messages and initiates the resulting creation of backward messages. GEN creates messages according to a certain frequency on the basis of random number generation. The frequency can be set at the beginning of the simulation or dynamically during the simulation through model component SETGEN. CONG takes care of some traffic reduction measures in case of congestion. The latter component will be further explained later.

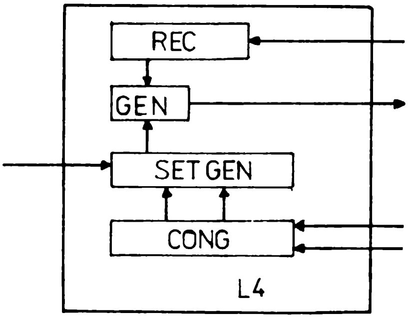


Fig.6

Level 3 (L3)

L3 simulates the signalling network function of Fig.2. Most of the functions, described in /CCa 85/, are precisely represented, because they were needed throughout the simulation. L3 is responsible for discrimination, distribution and routing of the messages (signalling message handling) and also controls the network configuration (signalling network management).

The signalling points in the simulation model have a limited capacity. They can only handle *n* messages per second, where *n* was different for each signalling points. The capacity limit *n* is simulated by only taking *n* messages per second from the input buffer of the signalling point. This input buffer is called reception buffer and is located at a signalling link.

2.4.2 Linkset (LS)

The communication between two signalling points is established by a signalling linkset which is a bundle of signalling links.

Model component LS simulates such a bundle. For the purpose of the simulation LS did not actually have to consist of a bundle of links. For the simulation it makes no difference whether there are 5 links transmitting with 64 kbit/s or one link transmitting with 5*64=320 kbit/s, if the sizes of the buffers within the links are multiplied with the same factor. Thus, in the simulation model a linkset can be as a link. And the link can be regarded as four buffers with two delay components between them as shown in Fig.7.

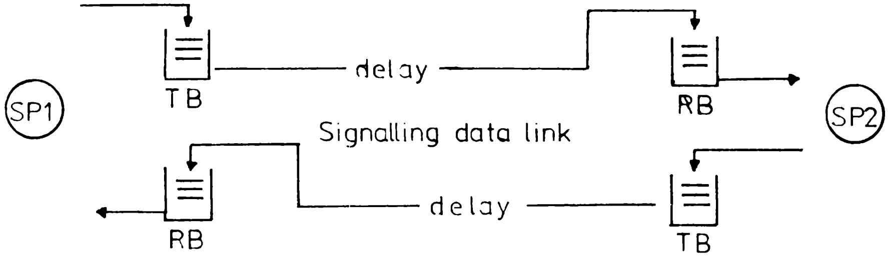


Fig.7

There is a reception and a transmission buffer for each direction. The transmission time between sender and receiver on the actual connection is simulated by the delay component. The messages for transmission are first put into the transmission buffer (TB). Usually after transmission and delay time the transmitted messages are put into the reception buffer (RB). If the receiving side of the link cannot take the messages, i.e. the RB is filled, the messages are collected on the transmitting side in the TB. More about these congestion situations will be said in the chapters 2.6 and 2.7.

2.5 Load model

The load is generated in the model component GEN. The generation is explained in the following with a model of two signalling points shown in Fig.8. It is an abstraction of the procedures, described in /1R7 85/ for the telephone user part (TUP).

GEN in SP1 generates "calls" with randomly produced time intervals between them. The mean value of the random time intervals can be set dynamically by parameter. A call is a sequence of 6 messages which are sent in intervals of approximately 1.5 sec. to a remote signalling point.

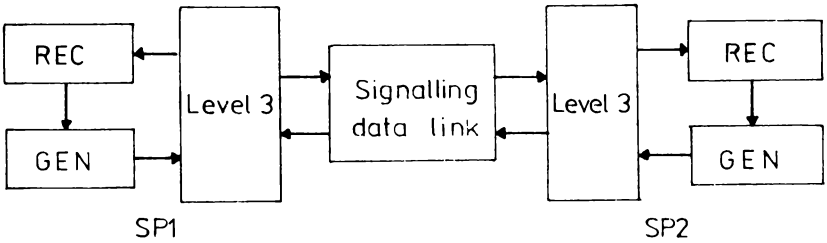


Fig.8

REC in SP2 receives the messages of the calls. After reception of the last message of a call, GEN in SP2 sends a (call-complete-) message back to SP1 after approx. 1 sec. After approx. another 15 sec. GEN in SP2 sends another message to SP1.

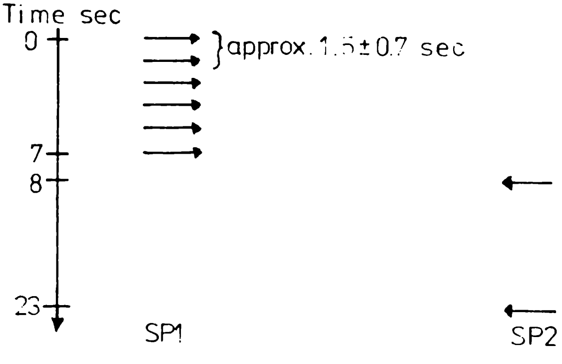


Fig.9

This load model only considers the connection establishment which was sufficient to study congestion situations.

2.6 Congestion indication and handling

Many methods for congestion avoidance, indication, and han-

dling have been proposed in the literature /BaD 79/, Dav 72/ and investigated by simulation /Pri 77/, /Hea 73/, /PrC 75/. Some of these methods are quite similar to those presented here.

There are basically two congestion indication methods that were investigated by this simulation:

- congestion indication by supervising the buffer contents (suggested in /1R7 85/) and

- additional indication by supervising the processor load.

Both methods will be briefly described in the following.

2.6.1 Buffer content indication

Each direction of a signalling link basically consists of two buffers as shown in Fig.7. The RB on the right side is emptied by the processors of SP2. If the processors in SP2 are in a congested state, RB will be filled up to a certain level. When this level is reached, the transmitting side of the link is told to send no more messages. That means TB will fill up. TB is supervised by the level 3, thus the level 3 functions will recognize when a certain upper limit is reached.

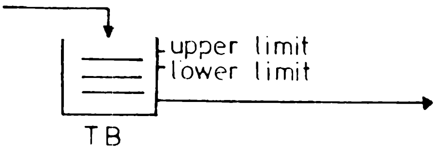


Fig.10

The level 3 functions will then take certain actions to reduce the traffic carried by the envolved route. After a while, hopefully, the traffic reduces and the buffer will be emptied again. When it reaches a certain lower limit, the level 3 functions will recognize that congestion is dissolved and the traffic can be brought up again.

It should be mentioned here that congestion is recognized at the transmitting side of the link, whereas the congestion situation itself has occured at the receiving side. Therefore this method imposes a delay on the congestion recognition.

2.6.2 Processor load indication

Congestion situations can also be recognized by supervising the level 3 processor load at the receiving side of the link. When the load exceeds a certain limit, e.g. 85% of the processor's capacity, traffic reduction actions are taken by the level 3 functions. This method is based on calculating a processor's load every once in a while, e.g. every second. When the processor load falls under a certain lower limit, the traffic can be brought up again.

2.6.3 Congestion handling

Level 3 and level 4 functions are responsible for the traffic reduction, once congestion is recognized by one of the methods described in the previous chapter.

Level 3 sends transfer control messages to those signalling points that cause the congestion. Those signalling points tell their user parts (level 4) to reduce stepwise the traffic for the concerned direction. If the user part is causing the congestion, it is also told to reduce the traffic. Informing the user parts is done by the level 3 with congestion indication primitives containing the destination point code (DPC) for the congested direction. The primitives are repeated every once in a while on a basis which is further explained in /1R7 85/.

Upon reception of a congestion indication primitive the level 4 initiates actions to reduce the traffic. The traffic is reduced for one step, e.g. 25%, for the indicated direction. A timer TS1 is started. All further primitives received for the same direction are ignored. Together with TS1 another timer TS2 is started which is much longer compared to TS1. The reception of a primitive after TS1 has elapsed, but before TS2 has elapsed causes another reduction step, e.g. another 25%. For each destination point code there is a step counter in the level 4.

If no primitive arrives within TS2 the traffic can be increased by one step. With respect to /1R7 85/ the values for TS1 and TS2 were

$$TS1 = 300\ ms$$
$$TS2 = 4300\ ms$$

Throughout the simulation they were varied to obtain some better performance, though.

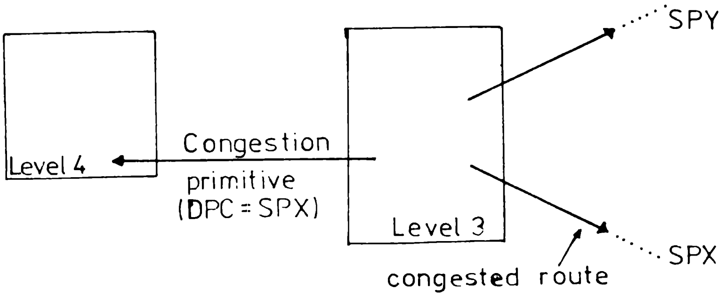


Fig.11

3. The experiments

The stepwise reduction of the traffic was made by trying various step schemes. The first and most unsophisticated scheme was the reduction by 4 steps (congestion levels), 25% each:

congestion level	permitted percentage of new calls
0	100%
1	75%
2	50%
3	25%
0	0%

Throughout a series of experiments, other step schemes were tried, some of which performed better than this very primitive one. More information can be found in /Hog 86/.

3.1 Exemplary run of a simulation

Fig.3 shows the basic model structure. SP1 and SP2 each sent $150M/s$ (messages per second) to SP3. The signalling traffic from SP1 to SP3 ran through SP4. The capacity of the different signalling points was different. SP1, SP2, and SP3 could each handle $700M/s$, whereas SP4 could handle $300M/s$.

After 10 seconds of simulation the linkset SP2-SP3 failed. Thus messages from SP2 addressed to SP3 had to be rerouted to SP1, from there to SP4 and then to SP3. The same rerouting took place for backward messages from SP3 to SP2. Thus SP4 now had to handle:

$$\begin{array}{rl}
150\ M/s & \text{from SP1 to SP3} \\
150\ M/s & \text{from SP2 to SP3} \\
50\ M/s & \text{backward messages from SP3 to SP1} \\
\underline{+\ 50\ M/s} & \text{backward messages from SP3 to SP2} \\
=400\ M/s &
\end{array}$$

After rerouting, a congestion occured at SP4, because it could only handle $300M/s$. The congestion was recognized by one of the congestion indication methods of chapter 2.6. The traffic was stepwise reduced until the congestion was dissolved. Then, according to the rules of chapter 2.6.3, the traffic was stepwise increased again. Therefore, after a while a new congestion came up, and the same procedure was repeated. Fig.12 shows an abstracted curve of the overall traffic at SP4 in M/s with respect to simulation time.

3.2 The simulation goal

The problem that arises from the protocol behaviour shown in Fig.12 was the oscillation of the traffic flow rate. Thus the resulting traffic actually transported by SP4 was not as much as it could have been. The maximum capacity of SP4 was $300M/s$. This value was only reached at the peakes of the oscilating curve. Thus most of the times SP4 transported much less messages. The average transported traffic could be much more if the curve was flattened a bit. This phenomenon is well known in telephone networks /Gim 74/, /Agn 76/.

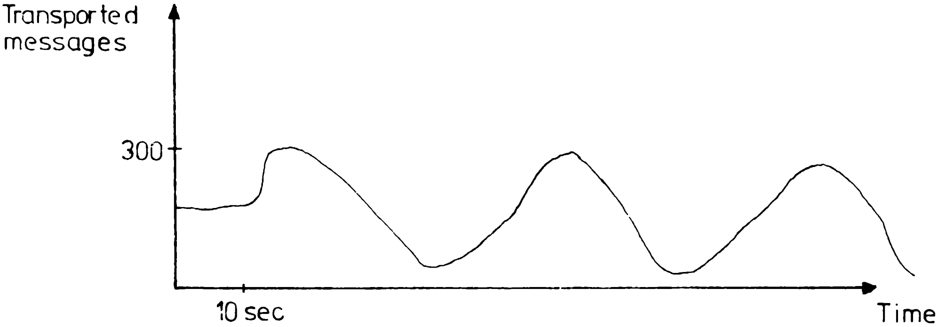


Fig.12

One of the goals of the simulation was to study the curve in Fig.12. The two different congestion indication methods were tried with different parameters and with different step schemes in the level 4 as described in the previous chapter.

Another goal was to study the average waiting time of messages in the concerned buffers and to try out methods to reduce these delays in message transportation.

3.3 Some results

The experiments showed that the second proposed congestion indication method, i.e. indication by supervising the processor load, has some advantages. Both the overall transported messages and the delays caused by the buffers could be improved. The reasons for this can be seen in the following.

Buffer content indication

The buffer supervising method takes some time to recognize the congestion. This is the time between the congestion causing event and the exceeding of the upper buffer limit. The concerned transmission buffer fills up, because the load cannot be handled by the concerned processors. This filling-up of the buffer is not recognized until the upper limit is exceeded. Thus no traffic reduction actions could be taken in the meantime. When the congestion is recognized after all, it still takes some time before the reduction methods become effective. During this time the buffer is filled even more so that the messages have to wait proportionally longer.

When the traffic has been reduced, the buffer has to be emptied by the concerned processors. This also takes some time. Thus it takes some time until the traffic reduction will be recognized by the level 3 functions. Until then further steps of traffic reduction have possibly been initiated although they would not have been necessary. This results in unnecessary little traffic carried by SP4 as shown in Fig.12.

The average transported traffic by SP4 in the simulation was about 68% to 80% of the maximum capacity of SP4. The average waiting time in the transmission buffer varied from 130 msec to 180 msec.

Processor load indication

The processor load supervising method is able to recognize the congestion much earlier, before the buffer has filled up. Thus the traffic reduction actions can be initiated much earlier so that there is almost no delay caused by message buffering.

Also the reduction of the actual traffic is recognized much earlier so that the traffic reduction actions can be stopped before reducing the traffic too much.

One disadvantage is attached to this method. The level 3 functions indicate congestion when the load exceeds X% of the maximum capacity, $X < 100$, e.g. $X = 85$. Thus the processor will never handle more than 85% although, theoretically, it could handle 90% or even 100%. This can be a disadvantage if the processor is driven around its capacity limit.

In the simulation with this method, the average transported traffic could be raised to about 82% when the congestion indication limit was 85%. The average waiting time was less than 10 msec.

References

[1] AGNEW, C., Dynamic modelling and control of congestion-prone systems, Operations Research, May 24, 1976.

[2] Anwendungsspezifikation für das CCITT-Zeichengabesystem Nr.7 im nationalen Netz der Deutschen Bundespost, FTZ, 1R7, Ausgabe 3, 1985.

[3] BARBER, D. L. A., and DAVIES, D. W., Computer networks and their protocols, John Wiley & Sons, 1979.

[4] BORIS Benutzerhandbuch, Siemens, 1986.

[5] CCITT: Specification of Signalling System No.7, Recommendations Q.701-Q.795, ITU, Geneva, 1985.

[6] CCITT: Specification an Description Language SDL, Recommendations Z.100-Z.104, ITU, Geneva, 1985.

[7] DAVIES, D. W., The control of congestion in packet switching networks, IEEE Trans. on Com., 1972.

[8] GIMPELSON, L. A., Network management: design and control of communication networks, Electrical communication, 49, 1974.

[9] HEALEY, R., Computer network simulation study, NPL rep. COM64, 1973.

[10] HOGREFE, D., and ZORN, S., Simulation of SDL specified models (in S. Shoemaker: Computer networks and Simulation III), North Holland, Amsterdam, 1986.

[11] HOGREFE, D., Untersuchungen am CCS7-Protokoll, SIEMENS-report no. ZTISOF/313/010/Hog, 1986.

[12] HOGREFE, D., Simulation of large SDL systems, Third SDL Forum, The Hague, 1987.

[13] PRICE, W. L. and COWIN, G. C., Simulation studies of the effect of link breakdown on data communication network performance, NPL rep. COM77, 1975.

[14] PRICE, W. L., Data network simulation: experiments at the National Physical Laboratory, 1968-76, Computer Networks, 1977.

(Received June 27, 1988)

DIETER HOGREFE
Fachbereich Informatik Universität
D-2000 Hamburg 13, Rothenbaumchaussee 67/69
GERMANY