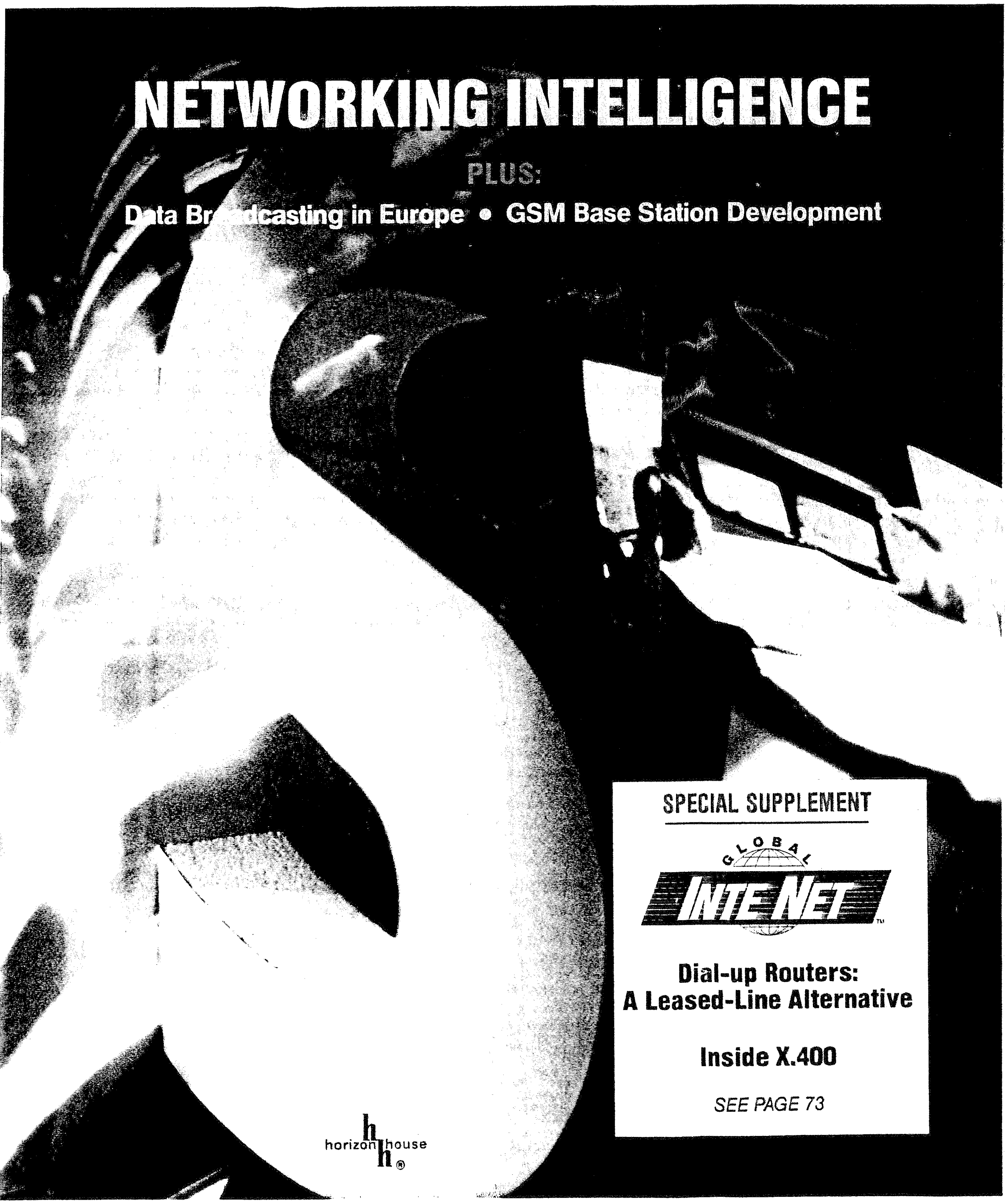
INTERNATIONAL EDITION

TELECOMMUNICATIONS®

TELECOM
BUSINESS
ASIA SUPPLEMENT

DATA + VOICE + IMAGE = NETWORKING ☐ MAY 1992



NETWORKING INTELLIGENCE

PLUS:

Data Broadcasting in Europe • GSM Base Station Development

SPECIAL SUPPLEMENT

GLOBAL INTE NET™

Dial-up Routers:
A Leased-Line Alternative

Inside X.400

SEE PAGE 73

horizon house®

Overload Control Strategies for Packet Switching

A number of dynamic control strategies for short-term overload situations on packet-switching networks are under examination.

Joachim Zepf, Manfred N. Huber, and Bernd Dasch

The increasing need for packet communication has led to a second generation of switching nodes that are characterized by high throughput and a modular structure. To cope with performance requirements, it is necessary that the behaviour of these systems should not be impaired dramatically by unforeseen heavy-load situations. This requires different levels of traffic control. This article focusses on call-control strategies during dynamic overload situations for the packet-switching system, EWSP.

THE EWSP PACKET-SWITCHING SYSTEM

EWSP is a high-performance packet-switching system with an expected throughput of more than 40,000 data packets per second and the capability to handle up to two million calls per hour (**Reference 2**). The modular design allows for the realization of small systems with approximately 100 lines, as well as very large systems with more than 10,000 lines, at reasonable costs. **Figure 1** shows the basic structure of EWSP consisting of four global functional blocks: the ring unit (RU), switching unit (SU), termination unit (TU), and management unit (MU).

The RU is the central entity of the system. It is a high-speed redundant ring network, and is used for interconnecting all other blocks. Each TU includes several line termination units (LTUs) that interface the switching node to subscriber lines and internodal trunks. A TU performs basically the functions of Layer 1 and 2 of the ISO reference model. Layer 3 functions are implemented in the SU. This unit processes call setup (i.e., routing), clear-down, and data packets (DPs). In addition to these functions, each SU performs safeguard tasks as well as front-end tasks towards the high-performance management system. The MU supports all management functions such as operation and maintenance (O&M), journalling, statistics, and alarming.

In order to achieve a high degree of reliability, each line is administered by the so-called default SU and the standby SU. If the default SU fails, the administration tasks will be performed by the standby SU. Each default SU is, at the same time, a standby SU for another set of lines.

The TU forwards an incoming call setup message to its default SU, called SU_IN (see **Figure 2**). SU_IN determines the outgoing line (routing) and transfers the call request to the destination TU via the associated SU (SU_OUT). The call confirmation packet uses the same path but in the opposite direction. The admission control procedure determines whether or not a call can be accepted and which SU will be involved in the data transfer phase. Only one SU (SU_ACT) handles the connection in an active manner. The other SU (SU_BACK) is used as a backup processor for this connection. All DPs, flow control messages of Layer 3, and other data transfer-specific packets are processed by SU_ACT (switching). In **Figure 2**, the transfer of a DP is also shown (in this example, SU_IN is the active SU and SU_OUT is used as SU_BACK).

GENERAL ASPECTS OF TRAFFIC CONTROL

EWSP is designed to transport a variety of traffic classes satisfying user and network requirements. To cope with these requirements, traffic control capability at the connection, admission control, and flow control levels is necessary. Connection admission control is the set of actions taken by the SUs at the call setup phase in order to determine whether or not a new virtual connection can be accepted. A call will be rejected if line-specific or global node-specific congestion occurs. If the various load thresholds in one or both SUs are not reached, the call can be accepted. For the determination of the active SU, two possible strategies can be used:

- For each new connection, SU_IN will become SU_ACT if its load level will not exceed the given threshold. Otherwise, SU_OUT will become SU_ACT.
- The load levels of both SUs are compared and the SU carrying the lower load becomes SU_ACT.

This article deals with the dynamic overload control of a single SU in order to cope with short-term overload situations caused, for example, by bursts of DPs.

(Continued on page 83)

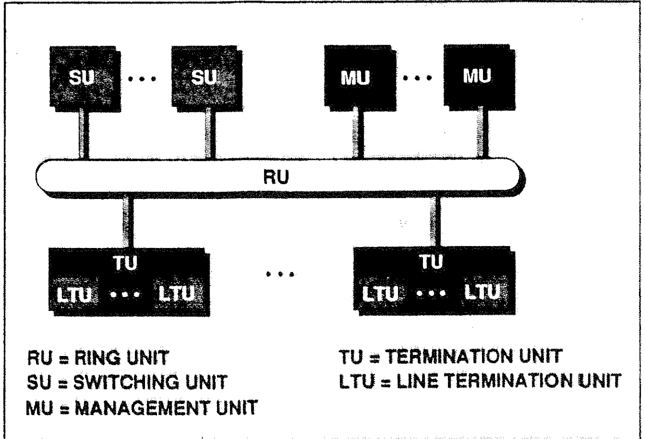


Fig. 1 Basic structure of EWSP.

(From page 47)

Long-term congestion situations are not taken into account. As already mentioned above, a SU processes call setup packets and DPs on a number of LTUs. Both types of packets will be stored in a common first-in first-out (FIFO) buffer. If the filling of this buffer exceeds a predefined overload threshold O, fast and efficient emptying of the buffer is required in order to avoid loss of packets. To prevent the system from changing too frequently between the normal state and the overload state, a hysteresis is provided. Normal operation is resumed when the filling of the buffer reaches the abatement threshold A, where A < O.

CONTROL STRATEGIES

In the following sections, different call control strategies, their modelling, and their performance are described. The proposed control strategies and their main principles in case of dynamic overload situations, can be summarized as follows:

- strategy 1: normal operation;
- strategy 2: call discarding;
- strategy 3: immediate call rejection;
- strategy 4: delayed call acceptance;
- strategy 5: delayed call rejection.

It is assumed that new connections might be accepted by the global call admission control, which means that the system resources to accept the new calls are available. Strategy 1, called normal operation mode, processes the incoming calls independently from the actual load state. This actual load state is represented by the buffer occupation of the common buffer for call requests (CRs) and DPs. This strategy is taken as a reference for comparison with the other methods. Due to the time-consuming processing time of CR packets compared to the switching time of DPs of an established virtual connection, the buffer occupation will increase quickly in the case of burst arrivals of DPs.

The intention of strategy 2 is to gain processing capacity for the DPs by discarding the time-intensive CRs in the case of dynamic overload situations. Due to the short processing time of the DPs, a fast and efficient emptying of the buffer should be achieved. The drawback of this method is the discarding of the CR packets. The Layer 3 protocol of the calling user must recognize the loss of the CR by means of a timer control. As an appropriate action, the user will repeat the call after the timer expiry. These follow-on calls will bring additional loads into the system.

Strategy 3 aims also at gaining processing capacity for the DPs, but now by rejecting the CRs. The time needed for rejecting a call is much less than for accepting it. A normal clear-down of the call is performed by sending a call clearing message. As above, the drawback of this method is that the users will repeat their calls. In the case of automatic call repetition, the follow-on calls can occur very quickly after the reception of the call reject. Therefore, the immediate call rejection can lead to a worse situation, if the system is flooded with follow-on calls.

The intention of strategy 4 is to set aside new calls in overload situations and to process these calls later, in the hope of an early termination of the overload situation. For the moment, only a short time is needed to set aside the call. Hence, the buffer occupation will decrease quickly. The delay of the call processing is assumed to be timer-controlled. After the timer expiry, the call is accepted. The idea of strategy 5 is to set aside new calls in a similar way to strategy 4, but to reject the call until after the timer expiry. To clear the call, the same procedure as described for strategy 3 is used. In comparison with strategy 3, this method, using delayed call rejection, avoids too many follow-on calls caused by an immediate call reject. Nevertheless, follow-on calls will occur.

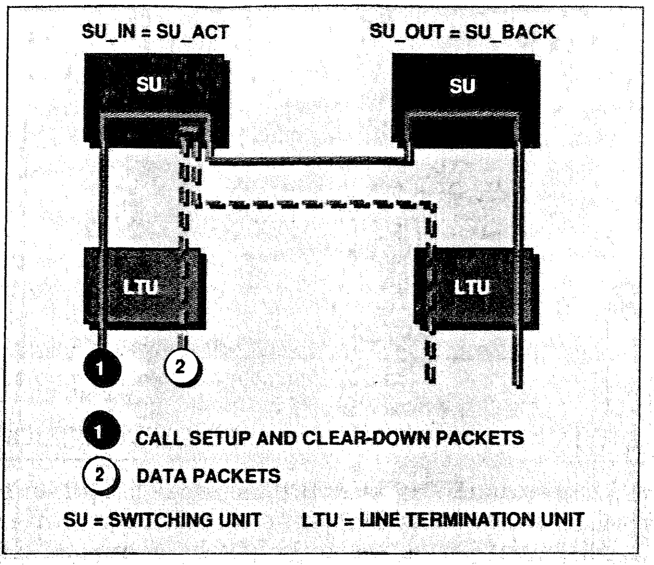


Fig. 2 Routing and switching procedures.

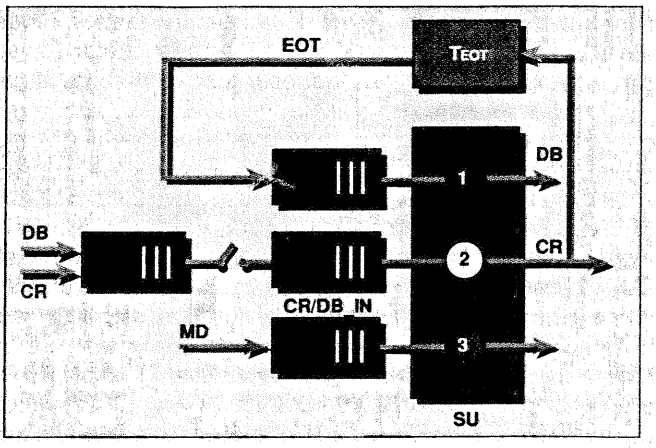


Fig. 3 Model 1 (normal operation).

MODELLING

As mentioned above, only one SU with the relevant components is considered. Therefore, the central component is the single processor of the SU performing several tasks. These tasks are call establishment and clearing; switching of the DPs; processing of acknowledgements generated by the transport protocol over the RU; and processing of management data (MD) (i.e., alarms and metering).

Due to the fact that both CRs and DPs are stored in a common buffer, in our model the processor consists of three phases:

- phase 1: processing of acknowledgements (end of transmission, EOT);
- phase 2: processing of CRs and DPs;
- phase 3: processing of MD.

Each phase gets its information out of a separate FIFO buffer. The CRs and DPs or data blocks (DBs) are previously written into a separate buffer, which is transferred cyclic and completely to the input buffer of phase 2. These facts are symbolized by a switch between both buffers. To decide about the overload situation, the two thresholds O and A performing a hysteresis are implemented at the input buffer of phase 2 (CR/DB_IN).

In the following, five models are introduced, each representing the appropriate overload control strategy (model 1 represents strategy 1, model 2 represents strategy 2, etc.). With the remarks above, the simplest model 1 for strategy 1 is depicted in **Figure 3**. Each phase is processing the appropriate information mentioned above. The loopback in **Figure 3** is caused by the transport protocol between SU and TU over the RU. Each transfer of a DB from a SU over the RU (out-transfer) leads to an acknowledgement EOT, which must be processed by the same SU (for details see **References 1** and **2**). Hence, in our model, each outgoing DB comes into the system again as an EOT after some delay time (T_{EOT}).

Model 2, which represents the strategy of discarding CRs in overload situations, is shown in **Figure 4**. The only difference to model 1 is a second loopback, which stands for the follow-on calls. If a CR is discarded, a timer of the Layer 3 protocol of the calling user will expire and the call will be repeated. In the model, each discarded CR comes into the system again as a new CR (follow-on call) after the delay time (T_{FOC}). This time, T_{FOC} includes the interval until the timer at the user side expires, and some additional delay caused by processing and transmission times.

Model 3, which is depicted in **Figure 5**, looks just like model 2. The only difference is that the follow-on calls are caused by rejected calls, which are cleared down with a clear message (CLR). Therefore, the delay time T_{FOC} between the CLR and the new CR now represents only processing times of the calling user and transmission times.

For the last two models, which represent the delayed processing of CRs, a new processor phase (phase 4 — processing of delayed CR), has to be introduced. In an overload state, the CRs are set aside and the processing is delayed under control of a timer T_1. After the expiry of T_1, the delayed calls are sent to a separate buffer, which is served by phase 4. Depending on the strategy chosen, the calls are either accepted or rejected by phase 4. The model 4, which is based on a delayed call acceptance, is shown in **Figure 6**. In model 5 (see **Figure 7**), the loopback for the follow-on calls caused by the delayed call rejection is just the same as in model 3 using immediate call rejection.

Finally, the scheduling of the processor phases is based on a polling mechanism with priorities, additionally interrupted by a cyclic timer T. Phase 1 always has the highest priority and serves the queue exhaustive (memory space can be given free). Phase 2 has the next lower priority. The lowest priority is assigned to phase 3. After the interrupt of timer T phase 1 is started, the remaining time is used by phase 2. This cycle proceeds until both input buffers are empty. If neither phase 1 nor phase 2 are active, or if under heavy-load conditions, the cycle has been passed n times (n = 9), phase

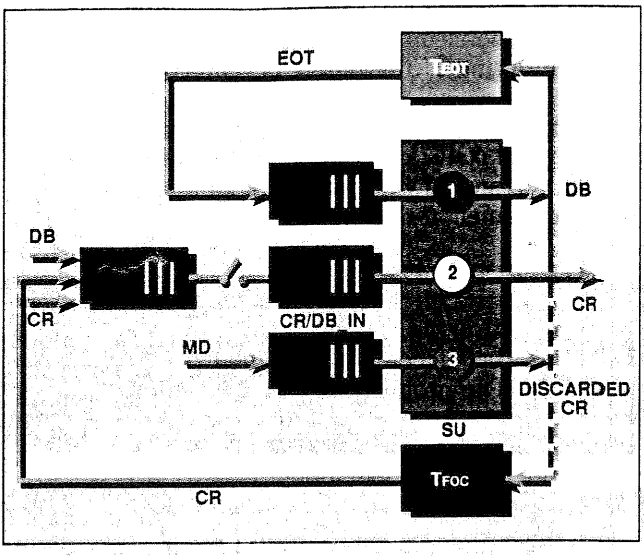


Fig. 4 Model 2 (call discarding).

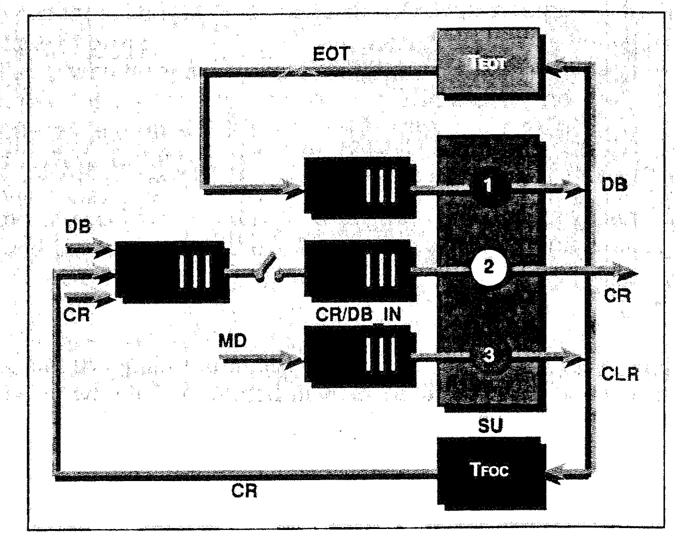


Fig. 5 Model 3 (immediate call rejection).

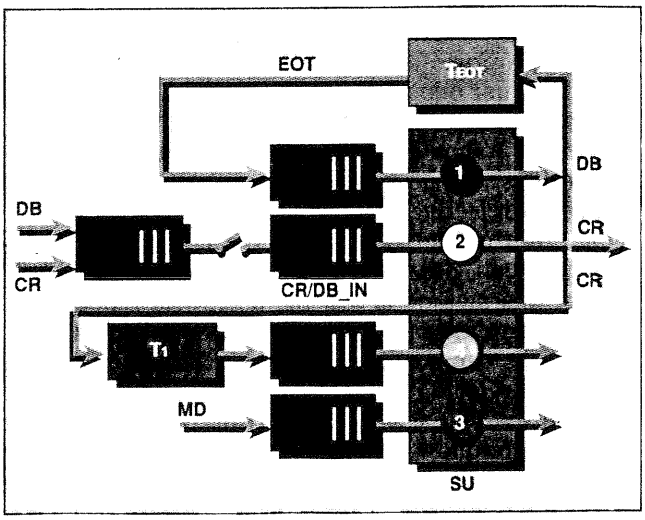


Fig. 6 Model 4 (delayed call acceptance).

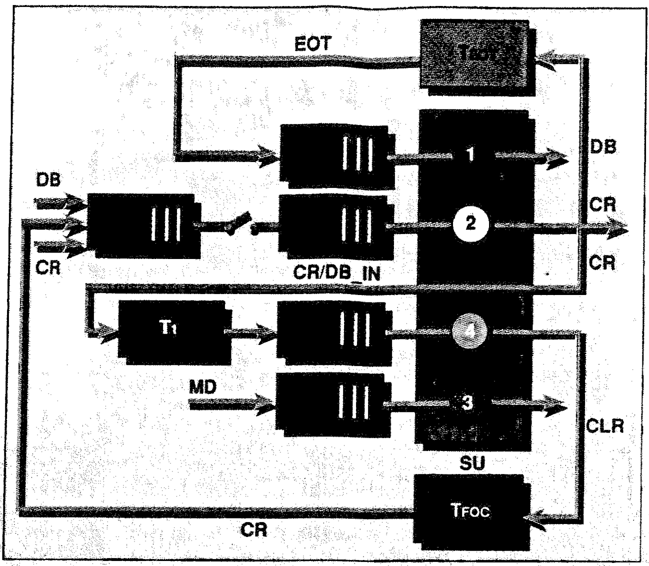


Fig. 7 Model 5 (delayed call rejection).

3 is started until the next timer interrupt (forced activation). If, additionally, phase 4 is existing, the priority of phase 4 is set between phase 2 and phase 3. The forced activation of phases 3 and 4 under heavy-load conditions is done alternately.

NUMERICAL RESULTS

The performance evaluation of the various overload control strategies is realized by discrete event simulation. The models introduced above were implemented in a computer simulation program. CRs, DPs, and MDs are generated by traffic generators with individual and variable interarrival time distributions. The generation of the dynamic overload situations is done by switching between two load states, a normal load state and a heavy-load state. In the heavy-load state, the arrival rate for DBs is increased for modelling of the bursts.

Some results obtained by these simulations are presented here. These results are based on the following assumptions for the mean processing time per message of each phase:

- phase 1 = 0.7 ms per EOT;
- phase 2 = 1 ms per DB/20 ms to accept a CR/4 ms to reject a CR/1 ms to set aside a CR;
- phase 3 = 10 ms per MD;
- phase 4 = 20 ms to accept a CR/4 ms to reject a CR.

The mean delay times and timer values are chosen as follows:

- T_{EOT} = 20 ms;
- T_{FOC} = 150 secs for model 2;
- T_{FOC} = 1050 ms for model 3 and 5;
- T_1 = 1 sec and the scheduling timer;
- T = 25 ms.

The maximum length L of the queue CR/DB_IN is set to L = 200 waiting places, the overload threshold O to O = 72, and the abatement threshold A to A = 56.

Under normal load conditions, the carried load r of the processor is assumed to be r = 0.7. With the processing times above and a given traffic mix, the mean interarrival times can be derived as 3.4 ms for DBs, 42.5 ms for CRs, and 360 ms for MDs. All arrival processes are assumed to be Poisson arrivals. The overload bursts are simulated by doubling the normal DB arrival rate, which means a mean interarrival time of 21.25 ms for DBs. These bursts have a constant duration. Nevertheless, they are varied, for the simulation runs between 0 and 4000 ms.

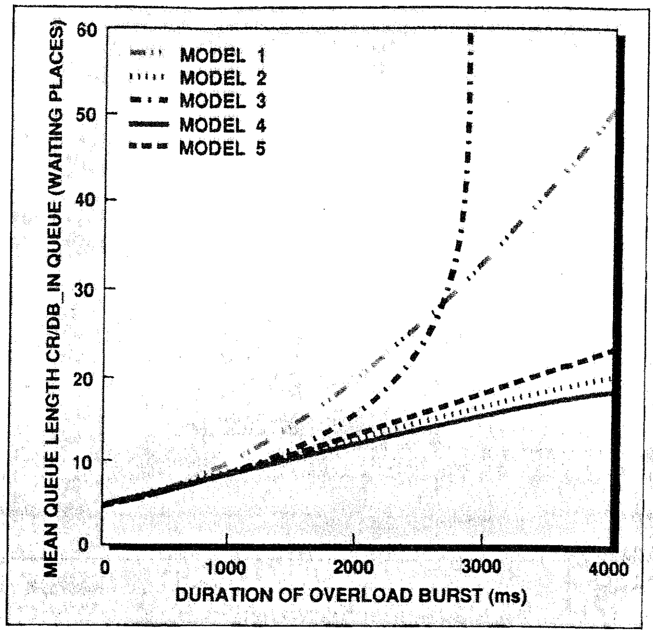


Fig. 8 Mean CR/DB_IN queue length vs. duration of overload burst.

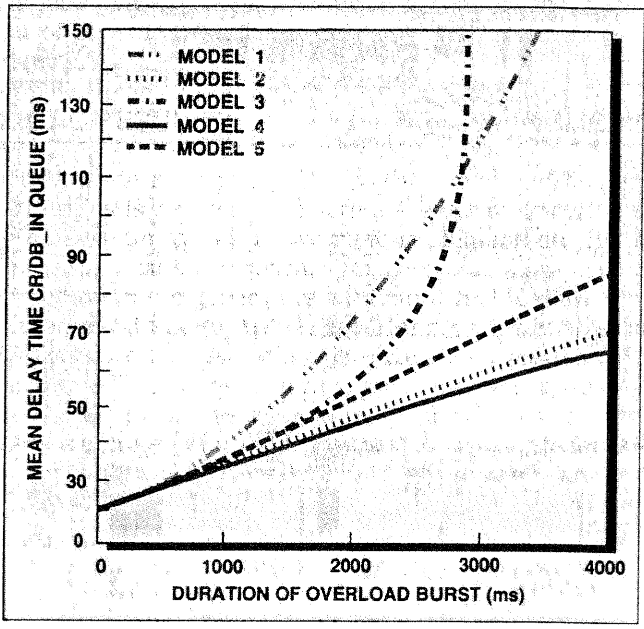


Fig. 9 Mean CR/DB_IN delay time vs. duration of overload burst.

In **Figure 8**, the mean queue length of the CR/DB_IN is depicted versus the duration of the overload bursts. The graphs show that with increasing length of the bursts the queue length increases. As expected, the queue length for model 1 increases more than the other ones, because the immediate processing of all CRs costs much time. It is interesting that the curve of model 3 (immediate call rejection) crosses the curve of model 1 early. Although the processing time for rejecting a call is much smaller than for accepting it, the system is flooded with follow-on calls, and this leads to a dramatic increase of the queue length. The behaviour

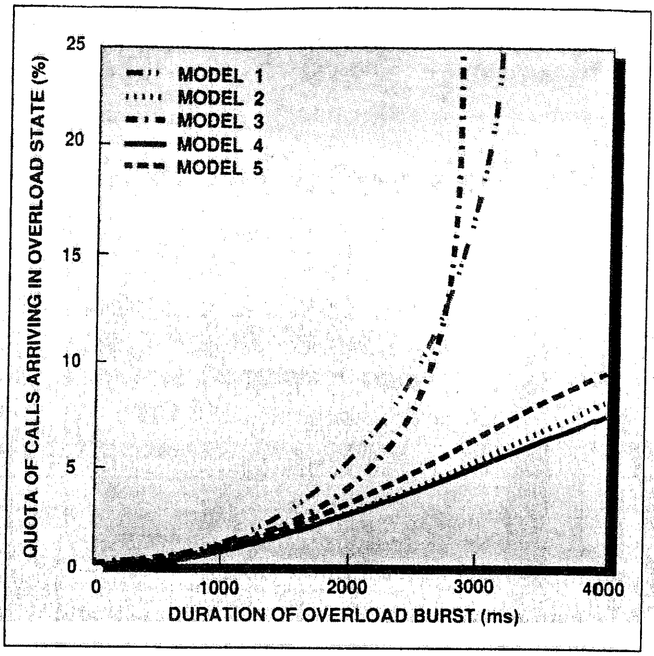


Fig. 10 Calls arriving in overload state vs. duration of overload bursts.

of models 2, 4, and 5 does not differ significantly among them, but it is much better than with model 1 or 3. Although not shown here, note that packet losses occur with model 1 and 3. Similar results are illustrated in **Figure 9**, where the mean delay time of the CR/DB_IN queue is shown versus the duration of the bursts.

Finally, in **Figure 10**, the quota of calls that are arriving during an overload situation is depicted versus the duration of the overload bursts. Generally, the picture shows the same tendencies as the figures before. Note that CRs during an overload situation are only rejected or discarded in models 2, 3, and 5. Using the models 1 or 4 means, nevertheless, that the concerned CRs are accepted. To summarize the results above, it can be easily recognized that strategy 3 (immediate call rejection) is the poorest method. The drawback of strategy 2 is the intentional discarding of messages and, hence, the intentional usage of protocol error correction mechanisms. The delayed acceptance and rejection of calls used in models 4 and 5 yields good performance values that do not differ significantly. ☐

REFERENCES:

1. Dasch, B.: *Ueberlaststrategien in der Paketvermittlung.* Telcom Report 13 (4/90), Siemens AG, 1990, pp. 144 - 147.

2. Mair, E., Hausmann, H., Naessl, R.: *EWSP - A High Performance Packet Switching System.* Proceedings of the 8th International Conference on Computer Communications, Munich, 1986, pp. 359 - 364.

3. Huber, M.N., Kiesel, W.M., Kuehn, P.J., and Raith, T.: *Performance Modelling of a Highly Modularized Packet Switching Node.* Proceedings of the 8th International Conference on Computer Communications, Munich, 1986, pp. 600 - 605.

Joachim Zepf *(Dipl.-Ing) is from the University of Stuttgart Institute of Communications Switching and Data Technics.*

Manfred N. Huber *(Dr.-Ing) and* **Bernd Dasch** *(Dr.-Ing) are from Siemens AG.*

Is this subject of interest to you?
Yes: Circle 217 No: Circle 218