Profit Optimal Congestion Control in Intelligent Networks

Åke Arvidsson[a], Stefan Pettersson[b], and Lars Angelin[b]

[a]Dept. of Telecommunications and Mathematics, University of Karlskrona/Ronneby, S-371 79 Karlskrona, Sweden

[b]Ericsson Software Technology, Research, Box 518, S-371 23 Karlskrona, Sweden

Current developments in technologies and markets stress the importance of flexible and robust performance optimisation and congestion control. In intelligent networks, congestion control has traditionally taken a mainly technical view and focused on protecting individual nodes from harmful overloads. We take a profit oriented, network wide view and develop a congestion control mechanism to ensure profit maximisation under real time performance. Numerical studies in a simulator show that the proposed mechanism is robust and, in particular in overloaded states, provides a substantial improvement in comparison to conventional load control mechanisms.

1. Introduction

1.1. Background

The telecommunications scene is presently undergoing a period of fundamental changes such as increasing transmission rates, user mobility, and service sophistication. At the same time we witness a growing usage of telecommunications, both in terms of number of users and applications and in terms of information volumes. This leads to a higher degree of dependence on telecommunications, which results in increasing demands on availability from businesses, governments, and the public. Furthermore, in the face of growing competition in the telecommunications industry following removal of monopolies, customers are successively learning to become less loyal to network operators and service providers with which they are discontent. Summing up, providing quality of service (QoS) is becoming more difficult in terms of conditions and expectations, and more important in terms of business survival.

1.2. Intelligent Networks

The IN [1,2] is a distributing and centralising framework of telecommunication services whereby new services may be introduced rapidly and cost effectively. Service Control Points (SCPs) contain the service logic that controls the IN-based services and form the core of an IN. SCPs may be assisted by Service Data Points (SDPs) which provide data about customers and the network, and Intelligent Peripherals (IPs) which support special functions like automatic voice announcements. End-user access to the IN-services is provided through Service Switching Points (SSPs) which contain the logic required to detect service access codes and send the requests to SCPs. SSPs and SCPs can be interconnected via a number of Signaling Transfer Points (STPs) which merely switch messages.

The mechanism of communication between SSPs and SCPs is by exchanging signals over the signaling network. Typically, ITU-T Common Channel Signaling System No. 7 (SS7) [3,4] is deployed. In SS7, signaling information is transmitted as Message Signal Units (MSUs), which by the signaling network essentially are treated as packets in a packet switched network. SS7 has a structure similar to the OSI-model, though for our purposes it is sufficient to distinguish between two layers: The Message Transfer Part (MTP), which provides reliable connectionless transport of signals, and User Parts (UPs), where the he actual signaling takes place.

1.3. Overload Control

INs are engineered to meet relevant real time demands. Above the engineered levels, however, response times may exceed time outs of signaling system protocols or not correspond to human expectations. This may cause repeated attempts which in turn increase loads further, and in the worst case a spiral of increasing loads is started which eventually may lead to complete service disruption. The literature contains a number of examples of this and other kinds of disasters, *e.g.* [5]. A whole range of signaling network congestion prevention mechanisms have been proposed, *e.g.* [6–10]. Most of these have been inherited from general packet switched networks, *e.g.* throttling and flow control, or from circuit switched networks, *e.g.* call gapping.

Our approach is based on the recognition that single signals have little or no value on their own since customers neither accept nor pay for anything but actually delivered services, and that signaling sessions may differ in value or urgency [11–13] depending on the service they support. We wish to ensure that each decision of whether to accept, delay, or reject a request [14,15] is made such that all accepted service requests can be successfully completed and that important ones are given priority. Our method is to make explicit use of some features which are exclusive to signaling traffic as compared to general data traffic, *viz.* that the traffic essentially consists of a large number of more or less predictable sequences each of which span over a relatively short time.

All numerical results presented herein are obtained from simulation. The work is a condensation and extension of earlier works, [16–18] and others, by the same authors.

2. Preliminaries

2.1. Model

We consider signaling networks with N nodes and M links. All nodes are identical and encompass the functionalities of SSPs, STPs, and SCP/SDP/IPs. Network topologies are symmetrical to allow simple parametrisation. S distinct services are supported, each with its own, specific *signaling sequence* which is executed each time the service is requested. Such an execution is referred to as a *signaling session*. The number of signals in each session and the list of nodes invoked may vary from one service to another (*e.g.* calling a fixed or mobile subscriber), and also from one session to another depending on the outcome (*e.g.* answer, no answer, and busy).

Let o and d denote any two distinct nodes $o, d : o, d \in \{1, \ldots, N\} \wedge o \neq d$. Requests for services of class $s : s \in \{1, \ldots, S\}$ are received from users at node (SSP) o at random according to a Poisson process of rate $\lambda_{s,o}$. Service execution is modelled as follows: A request for an s-service at o results in at most L_s signals being sent from o to arbitrary nodes d_l, $l : l \in \{1, \ldots, L_s\}$. The probability of the lth signal being sent to node d is $\alpha(l, s, o, d)$, where we for simplicity have used $\alpha(l, s, o, d) = 1/(N-1)$. All signals from o will lead to a response and we may thus speak of *signal pairs*. For simplicity we have assumed that the response to signal l to d_l is sent from d_l without further signaling, but the model allows for any node to respond after any sequence of subsequent signaling.

The nodes have two parallel and independent processors: One for the MTP functions and one for the UP functions. For convenience, we refer to them as the MTP and UP processor (MTPP and UPP) respectively. All incoming signals are processed by the MTPP to handle the transmission and switching protocols. It can then pass the signal on to either the UPP of the same node, or the MTPP of another node, depending on the destination of the signal. The former case applies to signals terminating at the current node (in which case the UPP handles the service protocols including further signaling), and the latter to signals which are bound for another node (in which case the next MTPP is obtained by the fixed routing protocol deployed).

Both processors have infinite buffers and the processing times are assumed to consist of two components, a constant one and one which is drawn from an independent negative exponential distribution. The averages of the two components are the same. We have tried two different values for the ratio η between MTPP and UPP times, 1:1 and 1:10. Transmission times are considered to be included in the MTPP times.

2.2. Congestion Control

To each service class s we associate a maximum signaling session completion time T_s which is set with respect to protocol standards and users' expectations [11–13] or, for pure load control purposes without stringent real time requirements, simply engineered to match completion times for nearly congested networks. We say that a request for an s-service is successful only if the completion time of the associated session is less than or equal to T_s, otherwise the request is unsuccessful. To model a properly dimensioned network, the processing resources and routing strategies of our sample networks are set so that all units are equally loaded and all requests successful under normal work loads. Hence this study does not explicitly consider other kinds of congestion, *e.g.* in the trunk network.

Assuming that only successful sessions generate revenues, it is immediately observed that unsuccessful ones, occurring *e.g.* during equipment failures or unexpected demand peaks, represent a double nuisance: Not only do they fail to generate an income, but the fact that they consume processing- and transmission resources while in the network, means that they actively contribute to delaying other sessions and thus further increase the number of unsuccessful ones. Clearly, the only way to completely eliminate unsuccessful sessions, is to expand network resources or in some other way remove the overload. This is also the natural option for repeated or permanent overloads. However, for unexpected and exceptional occurrences of overload, current equipment and load conditions set the limits, and the best one can do is to maximise the number of successful sessions. To do so, network resources should be spent on useful work, *i.e.* successful sessions, while unsuccessful ones should be removed from the network. This is also the aim of our congestion control mechanism: To predict the outcome of a session and reject the ones which eventually will become unsuccessful.

3. Prediction

3.1. Round Trip Delay

It is well known that successive observations of queuing system delays, *e.g.* in discrete event simulations, exhibit some degree of correlation. Using the correlation to our advantage, we propose a simple estimator which predicts round trip delay (RTD) for the k signal pair from the observations made at instant $k-1$. To account for the fact correlation drops with time, we let the estimator alter between two states, one where correlation is used and the other where it is not. In the former state are predictions equal to the last observations, and in the latter to the reasonably invariant signal pair processing times (which forms a minimum for RTD) scaled by the current network load. Processing times are approximated as the minimum RTDs and we let $t_{\min}(o, d, k-1)$ denote the minimum RTD observed for signal pairs from node o to node d up until instant k. To allow scaling are network loads characterised by the RTD they cause and expressed as the ratio of total delays (actual values) to processing times (minimal values). More precisely we define a measure $\rho = \rho(o, k)$ of the present overall network load from the perspective of an originating node o from at instant kth as

$$\rho(o,k) = \sum_{d=1}^{N} \tilde{t}(o,d,k-1) / \sum_{d=1}^{N} t_{\min}(o,d,k-1) \tag{1}$$

where $\tilde{t}(o, d, k-1)$ is the prediction of the RTD between o and d based on the information available at instant $k-1$. Note that $\rho(o, k) : 1 \leq \rho(o, k) < \infty$ and thus not identical to the conventional notation of load as system utilisation. The lower bound is obtained when the network is so lightly loaded that the total RTDs equal processing times, while higher values reflect the impact of delay. Also note that we have chosen a ratio of sums rather than a sum of ratios in order to reduce the impact of each single destination d.

Coming to the details of the predictor, we add one state to the ones previously introduced. The new state represents something in between the first two states. The purpose of the state is to exploit the extra information available by signal pairs in transit. The predictor thus consists of three states and three transitions as follows

State 1: The link has been idle for such a long time that the most recent RTD is no longer valid for prediction. We then set $\tilde{t}(o, d, k) = t_{\min}(o, d, k)\rho(o, k)$.

Transition 1: A signal is sent from node o to node d and State 2 is entered.

State 2: The signal causing Transition 1 has not completed the round trip. We set $\tilde{t}(o, d, k) = \tau(o, d, k) + t_{\min}(o, d, k)\rho(o, k)$, where $\tau(o, d, k)$ is the time so far consumed by the kth signal pair.

Transition 2: The signal causing Transition 1 has completed its round trip and State 3 is entered.

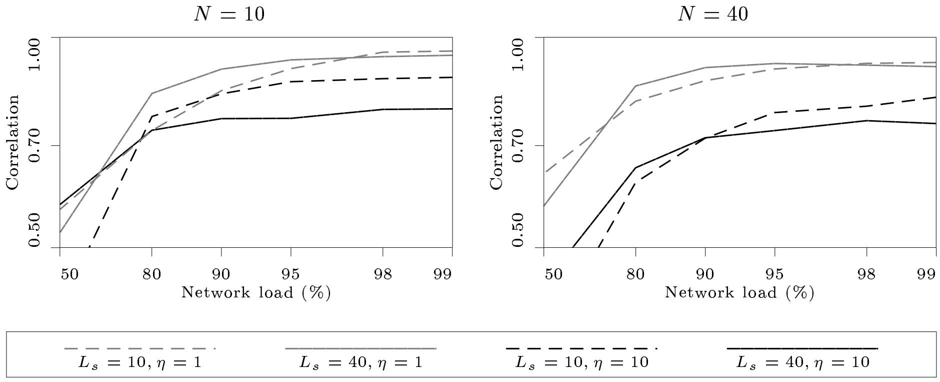

Figure 1. Correlation between predicted and actual session completions times vs. network load for various network sizes, sequence lengths, and processing ratios.

State 3: The time consumed by the kth signal pair is a recent RTD, $t(o,d,k)$, which can be used for prediction. Here we set $\tilde{t}(o,d,k) = t(o,d,k)$. Note that $t(o,d,k)$ is updated for every new round trip completion.

Transition 3: Too long time has elapsed since the kth signaling event and State 1 is entered. This happens when $u(o,d,k) = t(o,d,k)\rho(o,k)$, where $u(o,d,k)$ is the time elapsed since signaling event k between o and d.

Note that the signaling event counter k is unique to each state machine, one per origin-destination (OD-) pair, hence $k = k_{o,d}$ but we omit these subscripts for convenience.

3.2. Completion Times

The completion time for a class s signal session originating in node o for which l of the L_s signal pairs are already completed may now be estimated as

$$\tilde{t}_{L_s}(o,k,l) = \sum_{\ell=1}^{l} \tau_\ell + \sum_{\ell=l+1}^{L_s} \tilde{t}(o,d_\ell,k) \tag{2}$$

where τ_ℓ is the actual time consumed for signal pair ℓ and d_ℓ is the destination of the ℓth signal. Of particular interest is $\tilde{t}_{L_s}(o,k,0)$, the initial prediction for an s-session made before the first signal of the session has left the originating node.

To assess the power of our predictor (2), we study the correlation between predicted $\tilde{t}_{L_s}(o,k,0)$ and actual $\tilde{t}_{L_s}(o,k,L_s)$ completion times. Figure 1 show the results for network sizes $N = 10$ and 40, sequence lengths $L_s = 10$ and 40, and UP/MTP processing ratios $\eta = 1$ and 10. The most important observation is the high correlation in the region of interest, *i.e.* high loads, for all combinations of N, L_s, and η. The values indicate that even load levels down to 80% permits predictions of very good accuracy.

3.3. Control

Using our new estimator (2), we may apply load control by accepting a signaling session if $\tilde{t}_{L_s}(o,k,0) \leq T_s$ and rejecting it if $\tilde{t}_{L_s}(o,k,0) > T_s$. To evaluate the effectiveness of this

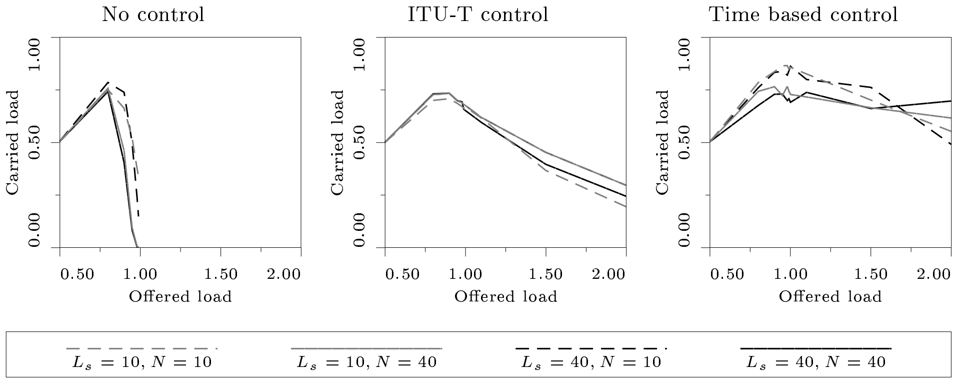

Figure 2. Effective throughput vs. input for various network sizes and sequence lengths.

strategy we study carried load, *i.e.* successful signaling sessions which are completed in time, as a function of offered load.

Figure 2 gives the results for $N = 10$ and 40 and $L_s = 10$ and 40. To limit the number of curves we only show the least favourable of the two processing cases above, *i.e.* $\eta = 10$ for which correlation is the smallest. The three diagrams refer to no control, a control mechanism similar to the one used by ITU-T [19,20][1], and our mechanism. It is seen that the latter improves real time performance for all values of N and L_s, in particular in comparison to no control at all, but also in comparison to the ITU-T control scheme. (Clearly the latter is neither designed to take a session perspective nor to consider real time performance, and that is precisely our point: The results we achieve with respect to *completed sessions* and *real time performance* could not have been obtained by means of conventional mechanisms. Although comparisons to other mechanisms with a scope similar to ours would be interesting, we have not found a standard candidate with standard settings that allows a simple and fair comparison.)

4. Decision

4.1. Network Profit

We distinguish the three distinct outcomes of a signaling session, *viz.* successful, unsuccessful, and rejected. *Successful sessions* generate revenues to operators in terms of call or service charges. The income associated with class s sessions is denoted by $I(s)$. The costs for successful sessions consists of a fixed processing cost $C_{\mathrm{pc}}(s)$ plus a variable

[1]In our implementation we have modelled the Discard and Abatement levels and the corresponding Transfer Prohibited and Transfer Allowed messages of the ITU-T recommendation as follows: The MTPPs have two threshold values with respect to the contents of their buffers, which we refer to as hold and resume respectively. When the buffer level of a node n reaches the hold level, are all downstream nodes notified that they should hold packets to n, and when the buffer level of n falls down to the resume level are all downstream nodes notified that they may resume sending packets to n. These messages are immediately acted upon, while in reality processing and transmission times would slow down communication between two nodes. Moreover, an effort has been made to optimise the settings of the two levels for optimum performance at 95% load. This means that the two parameters take different values for each N and L_s.

transport cost $(C_{\mathrm{t}} + C_{\mathrm{o}})t_{L_s}$. C_{t} is the cost for using transmission equipment *etc.*, and C_{o} is the marginal cost the session represent in terms of increased completion times and possible failure for other sessions. *Unsuccessful sessions* do not generate any revenues. The costs are the same as for successful sessions, but an extra bad will cost $C_{\mathrm{bu}}(s)$ for unsuccessful sessions is added. *Rejected sessions* are similar to unsuccessful ones in that they do not generate any revenues. The costs, however, are smaller since there are less processing costs $C_{\mathrm{pa}}(s) < C_{\mathrm{pc}}(s)$ and no transport costs. Moreover, the bad will cost $C_{\mathrm{br}}(s)$ for rejected sessions might be lower than the one for unsuccessful sessions if, *e.g.*, the user immediately is informed by some kind of announcement or message that the service request can not be accepted due to temporary overload.

Cost parameter values may be modified as desired over time. For customers such as other operators and service providers may costs be determined in direct negotiations, but for the large majority of customers the settings must be made from estimations and measurements. One factor often used is churn, *i.e.* the number of customers leaving the network divided by the average number of subscribers in a given period. Churn may be caused by customers finding more attractive offers with competitors or feeling mistreated and dissatisfied. Each customer that churns means lost revenues, *i.e.* costs for the operator, and the impact of poor QoS on churn is reflected as bad will costs.

4.2. Optimal Decision

We now apply Bayesian decision theory to combine satisfaction/profits with uncertain predictions in order to determine the most profitable action for any service request. The most profitable action is the action a_j that maximises the expected value of the gain function

$$E[w(\Theta, a_j)] = \sum_{\forall \theta_i} w(\theta_i, a_j) g(\theta_i) \tag{3}$$

where $A = \{a_1, a_2\}$ is the set of possible actions to be taken (reject and accept respectively), $\Theta = \{\theta_1, \theta_2\}$ is the set of possible outcomes (successful and not successful respectively), $g(\theta_i)$ is the probability of outcome θ_i, and $w(\theta_i, a_j)$ is the gain obtained in state θ_i for action a_j.

The probabilities $g(\theta_i)$ can formally be written as the probabilities that the completion time is less than or equal to the maximum completion time and greater than the maximum completion time respectively. Assuming that the estimation error $\tilde{t}_{L_s} - t_{L_s}$ is Gaussian with mean zero and standard deviation σ, it follows that

$$g(\theta_1) = \Phi\left(\frac{\tilde{t}_{L_s} - T_s}{\sigma}\right) \tag{4}$$

$$g(\theta_2) = 1 - \Phi\left(\frac{\tilde{t}_{L_s} - T_s}{\sigma}\right) \tag{5}$$

where Φ is the probability function of a standard (0,1) normal distribution. The derivation of σ involves two models, one of the prediction state machine, and another one of the network. The details are found in [18]. The gains $w(\theta_i, a_j)$ for the two actions and states are written as

$$w(\theta_1, a_1) = -C_{\mathrm{pa}}(s) - C_{\mathrm{br}}(s) \tag{6}$$

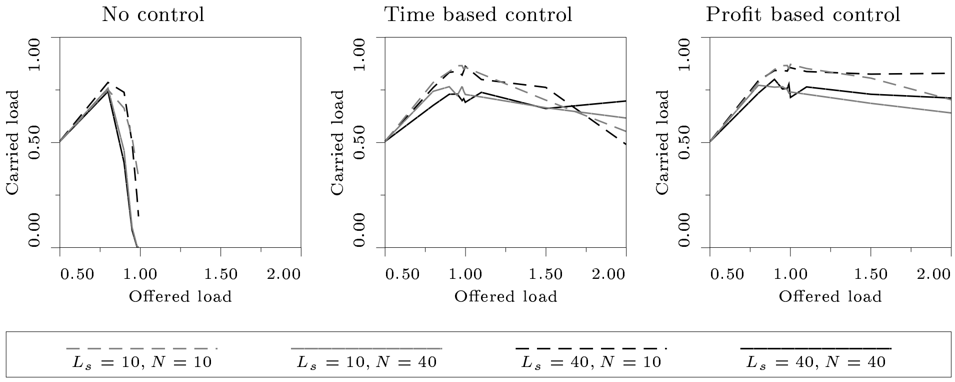

Figure 3. Effective throughput versus input for various network sizes and sequence lengths.

$$w(\theta_1, a_2) = I(s) - C_{\text{pc}}(s) - (C_{\text{t}} + C_{\text{o}})t_{L_s} \tag{7}$$
$$w(\theta_2, a_1) = -C_{\text{pa}}(s) - C_{\text{br}}(s) \tag{8}$$
$$w(\theta_2, a_2) = -C_{\text{pc}}(s) - C_{\text{bu}}(s) - (C_{\text{t}} + C_{\text{o}})t_{L_s} \tag{9}$$

using the notation of section 4.1.

4.3. Control

To evaluate our decision technique, we again study carried load *vs.* offered load. Figure 3 gives the results for $N = 10$ and 40, $L_s = 10$ and 40, and $\eta = 10$. The three diagrams represent no control, control with decisions based on completion time estimates (2), and with decisions based on expected profits (3). No specific costs are assumed but the gain for a successful session is equals to the loss of a rejected or unsuccessful session. It is seen that the latter performs the best, *i.e.* gets the most sessions through, though the degree of improvement varies from one set of parameters to another. The low improvement noted for low loads simply reflects the fact that low loads mean no congestion and hence no control mechanism should be active.

5. Application

Finally we apply our results to a simplified, typical application scenario. We consider networks of $N = 40$ nodes with non symmetrical topologies. The networks support $S = 3$ distinct services, the characteristics of which are given in Table 1. The values are somewhat arbitrarily chosen following discussions with operators. The nominal network load is 25% but due to some incident, *e.g.* a massive call in, this suddenly goes up to 200% in a particular node over a period of time.

With bursts of overload, the purpose of an overload control mechanism is then to ensure that network profit is kept at its maximum level during this period and that the network immediately returns to normal operation when the overload disappears.

The results are shown in Figure 4 as network revenues per time unit as a function of time for three distinct topologies. The curves refer to decision based control and time based control respectively, the case of no control is not shown since loads of 200% drive

Table 1
Service attributes used in Figure 4.

	1	2	3
Profit if successful (money units)	5	5	5
Cost if rejected (money units)	5	15	20
Cost if unsuccessful (money units)	10	20	30
Length (signal pairs)	15	25	30
Real time requirement (time units)	3	5	6
Fraction of offered traffic (%)	30	35	35

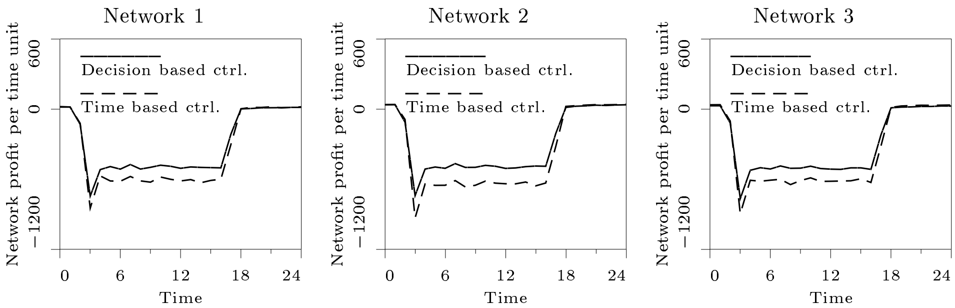

Figure 4. Impact of sudden, local overload in non-symmetrical networks.

the system unstable. It is seen that both mechanisms can handle the overload gracefully, but that the decision based mechanism clearly is the better of the two.

6. Conclusions

Current developments in telecommunication technologies and markets point at increasing demands on INs and signaling networks. These demands can partly be translated to requirements on system engineers to design robust control mechanisms that maximise a possibly weighted throughput of service requests under real time considerations. We have developed a congestion control mechanism which aims at maximising network profit expressed as a weighted sum of successfully completed service requests under all load conditions which can be used both for pure overload protection and for overload protection with real time performance requirements. The numerical results indicate that the mechanism offers considerable improvements in throughput, in particular under severe overloads.

Current issues include interaction with node based load controls, routing mechanisms, and service logic placement.

REFERENCES

1. J. Harju, T. Karttunen, O. Martikainen (1995) *Intelligent Networks*, Chapman & Hall.
2. J. Thörner (1994) *Intelligent Networks*, Artech House.

3. B. Jabbari (1991) Common Channel Signalling System Number 7 for ISDN and Intelligent Networks, in *Proc. IEEE*, **79**, 601–17.
4. A. Modarressi and R. Skoog (1990) Signaling System No. 7: A Tutorial, *IEEE Commun. Mag.*, **28**, 19–35.
5. D. Houck, K. Meier-Hellstern, F. Saheban, and R. Skoog (1994) Failure and Congestion Propagation Through Signaling Controls, in *Proc. 14th Int. Teletraffic Cong.*, Juan-Les-Pins, 367–76.
6. B. Banh and G. Anido (1994) Signaling Network Design Aspects For Mobile Services, in *Proc. Australian Telecommun. Netw. and Appl. Conf.*, Melbourne, 695–700.
7. P. Kühn (1995) Overload Control For Signalling Networks, in *Proc. 2nd ITC Reg. Sem.*, St. Petersburg, 1.
8. U. Körner, B. Wallström, and C. Nyberg (1994) The Impacts of New Services and New Control Architectures on Overload Control, in *Proc. 14th Int. Teletraffic Cong.*, Juan-Les-Pins, 275–83.
9. H. Nyberg and B. Olin (1994) On Load Control of an SCP in The Intelligent Network, in *Proc. of Australian Telecommun. Netw. and Appl. Conf.*, Melbourne, 751–56.
10. J. Zepf and G. Rufa (1994) Congestion and Flow Control in Signaling System No. 7 — Impacts of Intelligent Networks and New Services, *IEEE J. on Sel. Areas in Commun.*, **12**, 501–09.
11. B. Cotton and T. Pappas (1992) Effect of Service Completion Time and Network Interactions on the Acceptability of Advanced Intelligent Network Services, in *Proc. XIV Int. Switching Symp.*, Yokohama, B7.2.
12. R. Hallows (1995) *Service Management in Computing and Telecommunications*, Artech House.
13. D. MacDonald and S. Archambault (1994) Using Customer Expectation in Planning the Intelligent Network, in *Proc. 14th Int. Teletraffic Cong.*, Juan-les-Pins, 95–104.
14. M. Rumsewics (1993) Analysis of the Effects of SS7 Message Discard Completion Rates During Overload, *IEEE/ACM Trans. on Networking*, **1**, 491–502.
15. M. Rumsewics (1995) A comparison of SS7 Congestion Control Options During Mass Call-In Situations, *IEEE/ACM Trans. on Networking*, **3**, 1–9.
16. Å. Arvidsson, S. Pettersson, and L. Angelin, (1996) A Congestion Control in Intelligent Networks for Real Time Performance and Profit Optimisation, in *Proc. 10th ITC Spec. Sem.*, Lund, 347–58.
17. L. Angelin (1996) On the properties of a congestion control mechanism for signaling networks based on a state machine, *Technical report*, Univ. of Karlskrona/Ronneby, Karlskrona, ISRN HKR-RES–96/10–SE,1996.
18. S. Pettersson (1996) Some Results on Optimal Decisions in Network Oriented Load Control in Signaling Networks, *Technical report*, Univ. of Karlskrona/Ronneby, Karlskrona, ISRN HKR-RES–96/11–SE,1996.
19. W. Klein and R. Kleinewillighöfer-Kopp (1991) Performance Analysis of a Large Scale Common Channel Signalling Network, in *Proc. 13th Int. Teletraffic Cong.*, Copenhagen, 73–8.
20. K. Lee and Y. Lim (1989) Performance Analysis of the Congestion Control Scheme in Signaling System No. 7, in *Proc. IEEE Infocom '89*, Toronto, 691–700.