Failure and Congestion Propagation Through Signaling Controls

D. J. Houck, K. S. Meier-Hellstern, F. Saheban, and R. A. Skoog

AT&T Bell Laboratories, 101 Crawfords Corner Road, Holmdel, New Jersey 07733, USA

We investigate failure and congestion propagation via signaling controls, and develop design principles for preventing propagation. The interaction between network management and network element overload controls is studied through a detailed simulation, and the impact of disabling the controls as a means of avoiding certain types of fault propagation is discussed in detail.

1. INTRODUCTION

As a consequence of the central role signaling networks play in teletraffic networks, signaling network problems can profoundly impact network performance. This was recognized in the development of common channel signaling and signaling protocols, and network architectures have been developed to provide highly reliable systems that can accommodate network element failures. What was not well recognized is the capability for a network element fault to introduce a failure mechanism that can propagate, by means of the signaling network, to other network elements and cause a widespread performance degradation. With the introduction of Signaling System No. 7 (SS7) networks, there have been a number of fault/failure propagations, making network providers painfully aware of this vulnerability. By examining events that have occurred and identifying potential problems, four generic mechanisms for propagating failures have been identified: (i) network management/control messages; (ii) invalid messages; (iii) congestion and flow control procedures; (iv) inadequate recovery procedures.

In this paper failure propagation through network management/control messages is investigated, and design principles for avoiding failures are identified. Section 2 describes the event that occurred in the AT&T network in 1990, and identifies the generic fault propagation mechanism and design principles to control this type of situation. Section 3 discusses overload control principles, a key aspect of dealing with and preventing fault propagation. Section 4 discusses the role of network management controls and the impact of disabling them as a means of controlling fault propagation.

2. FAILURE PROPAGATION VIA NETWORK MANAGEMENT/CONTROL MESSAGES

On January 15, 1990 an AT&T network failure occurred. The AT&T 4ESS™ switches are signaling points in the SS7 signaling network. The problem began after the 4ESS switch in New York (switch A) took itself out of service during normal recovery from a minor trunk interface problem, announcing this to adjacent 4ESS switches by network management (NM) messages. The adjacent switch processors made notations in their programs to indicate that Switch A was temporarily out of service. Upon completion of fault recovery, Switch A announced its recovery to adjacent switches by resuming call processing and sending call setup messages to adjacent switches; each adjacent switch then noted recovery of A by an appropriate update in its program. A software flaw leading to network...
failure caused the adjacent switches to be vulnerable to certain types of disruptions for a few seconds during the period when such updates were being made. The fatal disruption occurred when switch A sent two very closely spaced call setup messages (within an interval of 0.01 sec.) to an adjacent switch (switch B). This caused some data at switch B to be damaged so the processor at switch B tried to execute an instruction that did not make sense. The software then told switch B that its processor was faulty, so switch B took itself out of service. When switch B returned to service, it repeated switch A’s actions, disabling additional 4ESS switches for a brief period of time. A chain reaction was initiated causing a network instability that lasted for hours.

Also, when the 4ESS switch experienced a failure and was isolated, its home STP would return a NM message to the adjacent signaling points in response to every message destined for the isolated 4ESS. During the first 30 minutes of the Network Incident, about 98% of 114 4ESS switches were in and out of service. The STPs became heavily loaded with NM messages and circuit switched signaling messages associated with Dynamic Non-Hierarchical Routing (DNHR) [3] retries and crankbacks. The generic scenario is the following:

A trigger event, a failure of some type, causes a network element to temporarily go out of service, slow down, or do something that triggers network management messages to other elements so they can take some control action. Some of the other network elements have a software (or protocol) bug in their network management/control software that is activated on receipt of the network management message. This bug causes some of the notified elements to fail in a way that they send out the same network management message. This in turn causes another group of elements to activate the software (protocol) bug, some of them fail, and the network becomes unstable with faults continually propagating through the network.

A common proposal to avoid this situation is software diversity. However, software diversity introduces other compatibility problems, it can be very costly, and it does not eliminate the problem (there are still subsets of elements having common software). Also, software diversity does not address potential bugs in the protocol. The network design principle that is proposed to address this propagation mechanism is the following:

**PRINCIPLE 1**: There needs to be a capability to deactivate NM/control software, i.e., network elements should be able to stop sending and stop responding to NM/control messages. This should be done in a layered structure so all controls do not have to be turned off at the same time.

The justification for this principle is that normal call processing/message handling software works properly. The situation described above is triggered by an infrequent event that activates infrequently used control software. Principle 1 stabilizes the network by deactivating the defective software, without knowing what specific software is bad, and then turns the NM/control software back on to resume normal network operation. Since an infrequent trigger event is required to put the network back into an unstable mode, the problem is not likely to recur before the problem is diagnosed and fixed.

In order for this principle to be viable, a corollary principle MUST be adhered to; namely:

**PRINCIPLE 2**: Every processor based network element must have adequate overload controls so that it can maintain a reasonable level of performance when NM controls have been deactivated.

3. ADEQUATE OVERLOAD CONTROLS

Several important contributions to overload control have been reported in the literature, e.g., [4] [5] [6] [7]. The results indicate that complex overload studies are necessary to define optimum control actions. The methodology described in this section is intended to be a guide for them.
In ideal systems, throughput rises linearly as a function of the input traffic until the system capacity is reached and then levels off. In uncontrolled systems, the rise is linear for light loading but beyond the saturation point, the throughput usually declines. Pending work builds up, delays increase, and productive throughput drops. Also, overloaded systems often have to contend with a positive feedback loop whereby some of the unsuccessfully processed tasks reappear later as "new jobs" bidding for service all over again. To prevent (or reduce) this drop, the system should be provided with a variety of internal and external overload control mechanisms that reject some of the load. Ideally, just enough load is rejected at a cost of zero real-time, so that the system maintains maximum throughput and achieves reasonable delays, while performing sufficient audits to maintain system sanity. In real systems, this can never be achieved so throughput under overload is less than maximum.

In general, the optimum overload control actions depend heavily on the system design and the traffic range covered by the system in normal conditions. An efficient philosophy in overload treatment is to adapt system functioning to the new traffic situation and, only under very extreme conditions, to adjust traffic sources to system capacity. The goals of overload control can be summarized as:

- Maintain good throughput, even under periods of extreme overload.
- Assure tolerable message delays not much larger than those under normal load.
- Assure that there is no breakdown due to overload, i.e. processor queues being overwhelmed.
- Ensure system sanity and responsiveness.
- Provide a graceful degradation of service, if inevitable.

Overload control should initiate appropriate actions to eliminate or minimize the effect of the detected overload conditions. These actions should be a series of incremental controls to ensure a stable level of system performance with graceful degradation. The processing work of the system under consideration can be categorized into message processing due to input traffic (call processing), processing due to rejecting messages, processing due to non-essential work, and processing due to systems checks. When real-time overload is detected, the general strategy should be:

- Limit the input of new call/message processing work. Traffic is usually controlled via a combination of internal and external controls. Internal controls (e.g., admission rate throttles, queue length limits, message timeouts, etc.) are taken autonomously by the network element to defer or shed offered load. External controls represent actions taken by the network element to generate network management messages (e.g. code gap message, etc.) to block traffic. The system should be designed to protect itself if the external controls do not work.

- Ensure that the amount of processing capacity needed to reject messages is small compared to message processing. The amount of real-time spent for shedding traffic by the network element should be small relative to the real-time spent for processing input traffic. As much processor capacity as possible should be used for useful jobs. In many cases simply ignoring new calls during overload maintains throughput at close to maximum.

- Defer the non-essential processing work. Many administrative and/or maintenance tasks can be deferred during overload.

- Ensure that a minimum amount of essential system checks are run to maintain system sanity. A certain amount of real-time must be allocated for processes needed to maintain system sanity. Since call processing usually runs at higher priority and may squeeze out the lower priority work, overload controls must ensure this essential minimum allocation. The overload states and their transitions should be defined in terms of the amount of real-time available for low priority
processes, which are used to detect and recover from data errors so system performance is not adversely affected. Execution of these types of non-call processing activities are usually essential to maintain the sanity of a processor based system.

- **Avoid rapid oscillation of overload states due to statistical fluctuations of traffic.** In general some hysteresis should be built into the overload control mechanism so that normal traffic fluctuations do not trigger overload controls. For example, if the detection of overload is based on processor real-time, then the overload control should compare a moving average real-time usage over several time intervals to predetermined thresholds to compute the overload level.

- **Prevent throughput degradation in the network.** The overload control must be capable of dealing with an overload in one part of the system without causing it to stop or slow other processing to the extent that it would create congestion in an adjacent node.

4. **DISABLING NETWORK MANAGEMENT CONTROLS**

This section describes Principle 1 in more detail. For this purpose, we consider a telecommunications network consisting of a set of signaling points (switches and Signal Transfer Points (STPs)).

4.1 Disabling Switch Network Management Controls

A major goal of network management controls for switches is to inhibit switching congestion and prevent its spread. [12] [13] [14] This can be accomplished by blocking calls with a low probability of completion at the originating switch by using external network management messages. The Automatic Congestion Control (ACC) capability in SS7 is used for this purpose. Information in the ACC indicator of SS7 messages allows each originating switch to maintain a Congestion Level for each terminating switch. Congestion Level 1 (CL1) indicates moderate switch real-time congestion, while Congestion Level 2 (CL2) indicates heavy congestion. Hard-to-Reach (HTR) values may also be computed by each originating switch. A HTR code is a 3- or 6-digit destination code to which successfully outpulsed calls have a very small chance of completing. [15] Originating switches use this information to block calls destined for a congested switch. Typical blocking probabilities might be CL1 not HTR: 0, CL1 HTR: .75, CL2 not HTR: .75, CL2 HTR: 1. Traffic is also controlled depending on whether it is alternate routed or direct routed. [16] [17] [18] This effect is not considered here.

We examine network management and internal switch control interactions by considering a hypothetical internal switch overload control which defers non-essential work and delays new call admissions (in accordance with Section 3). New calls arrive to seizure queues. The switch maintains separate seizure queues for origination (calls from other networks) and terminations (calls from within the network). Internal to the switch, tasks are served cyclically. Once per cycle, calls are removed from the seizure queues and placed into the cyclic server. Tokens limit the number of seizures per cycle that can be placed into the internal queues. If all tokens are used, remaining seizures are placed into token queues to wait for an available token on a subsequent cycle. The number of tokens allocated per cycle is reduced as the overload level increases (as measured by the queue cycle time) until it finally reaches zero in the most severe overload level. The token queue for originating calls is unlimited, while the termination token queue is limited to 16. Calls that are blocked from the termination queue are returned to the originating switch where they are blocked. (The cost to the terminating switch of returning the call is negligible).

The key parameters and values which affect the interaction of switch and Network Management controls are shown in Table 1.
TABLE 1. Key Switch and User Parameters

<table>
<thead>
<tr>
<th>Switch Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>switch real-time for an origination</td>
<td>4.5 ms</td>
</tr>
<tr>
<td>switch real-time to terminate a call</td>
<td>8 ms</td>
</tr>
<tr>
<td>switch real-time for a termination</td>
<td>3.5 ms</td>
</tr>
<tr>
<td>switch real-time to block a call</td>
<td>2.25 ms</td>
</tr>
<tr>
<td>switch real-time to delay a call</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>retry probability given that a call is blocked</td>
<td>.8</td>
</tr>
<tr>
<td>mean retry time given that a call is blocked</td>
<td>30 sec</td>
</tr>
</tbody>
</table>

Table 1 does not include abandonment parameters. Originating abandonments do not strongly affect our results, and finite queues cause terminations to be blocked before they abandon.

To illustrate network interactions, we consider a focused overload in a 5-switch network. The Network Management Control is the ACC, with the typical blocking parameters listed above, the parameters in Table 1, and a Hard-to-Reach probability of .9. The ACC control is assumed to be active for 3 seconds. The exact switch capacity depends on the mix of originations and terminations, but for the parameters here, it is approximately 750,000 calls/hr. Four of the switches in our example are loaded to just under capacity at 720,000 calls/hr while the fifth switch is loaded to 900,000 calls/hour with a focused load from the four non-overloaded switches. Load is equally distributed among the non-overloaded switches. The focused overload is achieved by routing 37.75% of the traffic from each non-overloaded switch to the overloaded switch, and dividing the remainder of the load equally among the remaining non-overloaded switches. In the figures, switches 0-3 are non-overloaded, and switch 4 is overloaded. We used a detailed simulation to model the network behavior under overload. We have also used flow equations (not reported here) to generate similar results. Flow models [19] have the advantage of simplicity, but do not capture as accurately dynamic behavior such as oscillations.

Figures 1-4 show the overload behavior of this model network. Switch 0 is one of the "non-overloaded" switches and is representative of the other three, while Switch 4 is the "overloaded" switch. As can be seen in Figures 1 and 3, all switches become overloaded and a severe throughput degradation results. Figures 2 and 4 illustrate an instability in the real-time overload levels of the switches that helps the congestion propagate from Switch 4 to the non-overloaded switches. The instability is amplified by the virtual "all or nothing" blocking of traffic by the ACC control. The fact that the control lasts for several seconds contributes to the over-control. The overload propagation time appears to be on the order of several minutes. The main reason for the propagation of overload is that blocking a call costs about 1/2 of the real-time of a completed call. Assuming a retry probability of .8, there are about 5 retries per call that is continually blocked, so that a total of 5/2 times the real time of a "good" call is spent to block a call. This causes switches that are just below overload to become overloaded when they must block many calls due to ACC controls. The ACC control affects terminating calls to the overloaded switch, but note that these consume less real-time than originating calls.

Figures 5-8 show the effect of disabling the ACC control in this example. The finite termination queue at Switch 4 allows it to shed its excess load without any affect on its throughput (Figure 7). The terminations that were blocked at Switch 4 reduce the carried load at Switch 0 slightly, and increase the retries and thus the total offered load (Figure 5). As seen from Figures 6 and 8, the overload level at Switch 4 is fairly stable around Level 2, while the overload level at Switch 0 gradually moves up to Level 2. Overall, the network seems to be performing reasonably well without the ACC controls.

This illustrates the usefulness of Principle 2: the switch should be able to adequately protect itself under overload, and Principle 1: the network management controls should have the capability to be disabled.
To further investigate the cause of the undesirable performance of the ACC controls, we also considered the interaction of the ACC control and the size of the token queue. Recall that the termination queue holds a maximum of 16 calls, and when full, calls are returned to the originating switch for blocking. We observed the following performance under different configurations:

- Finite queue, ACC on (Figures 1-4): unstable overload levels, large throughput degradation;
- Finite queue, ACC off (Figures 5-8): stable overload levels, overload propagates to Switch 0, small throughput degradation;
• Infinite queue, ACC on: stable overload level at Switch 4, unstable overload level at Switch 0, medium throughput degradation;

• Infinite queue, ACC off: stable overload at Switch 4, no overload propagation to Switch 0, no throughput degradation.

Thus we observe that the interaction of the ACC control and the finite termination token queue adds to the network instability. Note that to fully consider infinite queues, it would be important to consider abandonments, which are not in the present model. This, as well as several other examples have suggested to us that there is a delicate interaction between switch and network management controls. Network management and individual controls can interact and need to be considered together. Given the size and complexity of the network, this poses a formidable challenge, but the potential benefits will allow us to avoid potential future network catastrophes.

These models indicate that the existing ACC controls have the potential to propagate overload conditions and that the networks worked better without the controls. The main problem seems to be the coarseness of the controls (only two levels) and the duration (3 seconds) that they are on. If each network element has appropriate overload controls to protect itself, then it might be more appropriate to apply network management controls only for long duration overloads and to give the controls more granularity. Thus a new Principle for reducing the spread of congestion in the network suggests itself.

**PRINCIPLE 3:** With adequate network element overload controls, NM congestion controls should only be applied during sustained overloads and should have a fine enough granularity to throttle the appropriate amount of traffic.

### 4.2 Disabling Signaling Network Management Controls

The purpose of signaling network management functions is to provide reconfiguration of the signaling network after failures and to control the traffic to prevent congestion.

Signaling Network Management functions can be divided into three classes: Signaling Traffic Management; Signaling Link Management; and Signaling Route Management. The signaling traffic management function is used to divert signaling traffic from a link or route to one or more different links or routes; or to temporarily slow down signaling traffic in response to signaling point congestion. The signaling link management function is used to restore failed signaling links, to activate idle (not yet aligned) links and to deactivate aligned signaling links. The route management function is used to distribute information about the signaling network status, to block or unblock signaling routes.

For this paper, only the signaling route management and signaling congestion control are most relevant. Signaling link management is used only for activating or deactivating a link. The effect of disabling Signaling Traffic Management (excluding congestion control procedures) on the performance of the network is negligible, since the emergency procedures can be used for diverting traffic. Emergency procedures do not require sending NM messages for diverting traffic.

**Disabling Transfer Controlled (TFC) Procedures**

**Normal Condition:** An STP sends, in response mode, TFC messages about a given destination to originating signaling points to notify them that they should no longer send messages with a given priority or lower to the controlled destination. When an originating signaling point receives a TFC message relating to a given destination, the signaling point checks the congestion status for the destination contained in the TFC message with its own record of the congestion status of the destination. If its current congestion status of the destination is not greater than the congestion status in the TFC message, then the originating signaling point updates the congestion status of the destination with the new value carried in the TFC message.

**Disabled Condition:** If an STP does not send a TFC message, then other signaling points will continue to send messages with priorities lower than allowed over the route. When the STP receives messages
with priorities below the current signaling route set status, the traffic builds up to the congestion level; then the STP discards the messages.

In a similar fashion, we can describe the impact of disabling the other Signaling Route Management controls; Transfer Restricted (TFR), Transfer Prohibited (TFP), Transfer Allowed (TFA), Signaling Route Set Test (SRST) messages, and Route Set Congestion Test (SRSCST).

In each case the impact of disabling signaling NM procedures is the possible loss of messages. We now describe the effect of lost messages on Plain Old Telephone Service (POTS) and noncircuit related services (e.g. 800 services, etc).

A detailed model with different call types is used to compute the impact of lost messages on calls. POTS calls are categorized into three types: call is answered and caller hangs up first; call is answered and called party hangs up first; and call does not go through, there is no answer, or it is busy. Furthermore, each type of call is categorized by the number of Initial Address Messages (IAMs) transmitted into the following classes: a call with one IAM transmission; a call with two IAM transmissions and no crankback message; and a call with two IAM transmissions and a crankback message.

For typical values of call fractions the probability that a lost message will kill a POTS call is about 0.3. This is due to the fact that loss of signaling messages may not necessarily kill a POTS call. For example, loss of Release and Release Complete messages do not kill the call. However, all calls requiring noncircuit related transactions will be killed if any of the the noncircuit related messages is lost. It should be noted that messages are likely to be lost only during the period of time between deactivation and activation of the NM/control software.

It is obvious that a detailed model of a complete signaling network, which takes all the influences of disabling congestion control software into account, is rather complex. In our approach to investigate the impact of disabling congestion control procedures we consider a simple model depicted in Figure 9 (actual networks have more redundancy to meet reliability needs). It consists of 12 originating signaling points, 12 destination signaling points and a pair of STPs which are connected by a pair of 56 kps links.

![Figure 9. Basic Model](image)

A detailed model comprising the relevant MTP levels 2 and 3 protocol functions is used as a basis for simulation. In summary, this model comprises all message streams, transmission channels, the congestion control procedures, and all other relevant parts of the signaling network. Our main interests are focused on the observation of the impact of disabling congestion control procedures on the throughput of the network. This model does not consider the impact of signaling end point real time consumption for blocked calls.
Figure 10. Network Performance With and Without Signaling Congestion Controls

The basic differences between the network with congestion control and without congestion control procedures can be appreciated by observing the simulated network performance shown in Figure 10. This figure shows the throughput as a function of offered load. As can be seen, the network, even in the absence of congestion control procedures, shows relatively robust performance. This figure shows the load from 0.8 to 1.6 Erlang (per link). As the load is increased from 0.8 to 1 Erlang, the two curves almost exactly coincide. (The effect of real time consumption of blocked call versus losing the call in the signaling end points is not considered). When the load is increased between 1 and 1.15 Erlang, the case without TFC shows a better throughput. This shows a network over-reaction under congestion with generation of TFCs. When the load is increased from 1.15 to 1.6 Erlang the two cases show similar throughput with the TFC case being slightly better. The improvement is because of elimination of ineffective call attempts by the TFC procedures. As stated earlier, the impact of disabling congestion control procedures on the real time consumption of signaling end points is not considered. These results also demonstrate the validity of Principle 3.

5. SUMMARY

In this paper the propagation of congestion through network management messages was investigated, and principles for network design were developed that should help eliminate this cause of major network outages. A key element in this area is the interaction between network management controls and network element overload controls. Adequate overload controls were shown to be a key aspect of dealing with and preventing fault propagation. With adequate overload controls in place, it is argued that NM congestion controls need to be applied only for sustained overloads and with a finer granularity.
6. ACKNOWLEDGEMENTS

The authors would like to acknowledge helpful discussions with many colleagues, especially Don Mintz, Robert Fishell, and Howie Kerr. We are grateful to William Mayer for the examples in Section 4.2.

REFERENCES