Dynamic Congestion Control in Circuit-Switched Telecommunications Networks

F. Langlois, J. Régnier
BNR, Canada

Abstract

This paper presents an adaptive, near-real time traffic congestion control system for modern telecommunications networks. The system centers around a network processor to which switches report, every 10 seconds, their status and the status of the network elements which they can directly monitor. When needed, the network processor communicates to the switches directives regarding the volume of call attempts that should be allowed into the network towards specific destinations. The performance of the system for different types of measurements and controlled resources and the dependance of performance on system parameters are evaluated via computer simulation.

The results demonstrate that the system effectively prevents congestion, that it manages well different types of network resources, and that it is robust with respect to inaccuracy in the setting of its parameters. Due to common architecture, measurements and complementarity of functions, the system may naturally be developed as a complement to centralized adaptive call routing systems such as Dynamically Controlled Routing [1].

1. Introduction

It is well known that, when left uncontrolled, complex telecommunication networks may perform poorly under overload conditions [2]. The overhead for partial treatment of call attempts which will eventually be rejected reduces the capacity that resources may devote to successful attempts, and may in extreme cases drive a network into a totally ineffective standstill. Traffic congestion control is aimed at preventing such situations.

In telecommunications networks, traffic congestion control is implemented mainly as explicit controls under the supervision of network managers, but also implicitly in network equipment. Explicit controls throttle the admission of call attempts or divert call attempts from certain routes; e.g., code block, call gap, cancel to, cancel from, skip route and re-route controls. Implicit controls limit the attempt rate on resources as they become congested by increasing the delays for reaching them; e.g., inter-office call set up delay with in-band signalling. Explicit controls are typically overrides altering the planned internal network operation. Their effectiveness depends on the rapid collection of operational measurements and on the experience, insight, and promptness of network managers for rapid problem identification and translation into appropriate contingency plans.

The above approach to traffic congestion control is based on the assumption that manual analysis and intervention is sufficiently fast to preempt or cope with crisis. The telecommunications environment traditionally satisfied this premise because the customer base and behavior were stable and predictable, and because network equipment configuration was typically a slow off-line provisioning process. Furthermore, failure of individual network components could not rapidly propagate to degrade the operation of other network equipment. However, technological innovation, the nature of the customer demand, and competition have significantly changed this environment.

Technological innovation has increased by orders of magnitude the capacity and speed of both network equipment and networking support infrastructure. Digital switching, fiber optic transmission, digital cross-connection, and common channel signalling are ubiquitous examples of this evolution. Telecommunications networks have become topologically smaller and denser, relying on fewer but larger network equipments, and hence are now more vulnerable to sudden and intense traffic stresses due to failure of individual components. Furthermore, better networking allows propagation of these stresses far faster than before. This last observation is well exemplified by the elimination of "choke" networks by common channel signalling. These imposed an implicit limit on the inter-office attempt rate as a consequence of the long in-band call set-up and holding times. By reducing these times, particularly when the busy tone is generated by the originating office, common channel signalling leads to far greater attempt rates on an office subject to mass focused calling.

Customers have benefited just as much as the telecommunications operating companies from technological innovation. This, in the process, has changed the nature of their demand. New services have appeared, imposing significantly more pressure on network resources than the traditional telephone service. Facsimile, data communication, automatic credit checking, electronic fund transfer in the retail industry, ISDN, answering machines and call services such as automatic re-dial have already had a significant impact on both the call attempt volume and on resource consumption per call. Emerging technologies such as fiber to the home and low power mobile communication may only further contribute to this pressure.

Not only has the nature of the customer demand evolved, but the customer base itself has also evolved. Indeed, deregulation and bypass of their service offerings by equipment and third party service providers have transformed the telecommunications operating company traditionally captive customer base and, as a result, have transformed as well its business motivation for responsiveness and service quality. Grade of service and reliability are now becoming competitive advantages, not regulatory constraints. Furthermore, telecommunications service providers must now shield their customers from the mishaps of other service providers, a constraint they never had to cope with before. For instance, by provoking a massive influx of traffic, the failure of a provider's fiber facility may jeopardize the grade of service another provider can provide to its regular customers, no matter how efficient its own facility restoration plans are. Overall, competition has clear implications on traffic management: it makes the customer base more volatile yet it makes it imperative to satisfy customer demand well.

Adaptation to these new technological and market pressures can be achieved to a large extent by adding redundancy, flexibility, and responsiveness to the network. Products such as SONET rings provide by means of redundancy full immediate protection against fiber facility failures, while products such as add-drop multiplexers and digital crossconnects, provide an immediate reconfiguration capability. Intensive capital expenditure in network equipment is, however, neither economically justifiable nor alone sufficient. The network management support infrastructure must be modernized as well. Hence traditionally manual operations such as alarm-to-service correlation [3] and adaptive traffic re-routing [1,4,5] are being embedded in network equipment, and operation systems relying on manual intervention are being developed or upgraded for providing better responsiveness. For instance, centralized monitoring and control systems from which control actions can be taken on a 5-minute basis have been implemented in several networks [6,7,8]. Current proposals or modernization efforts under way for these systems call for even more stringent responsiveness requirements, such as a 1 or 2-minute delay [9,10].

In line with this modernization of traffic management through the migration of manual operations into the network equipment and the
centralization of monitoring and controls, this paper proposes a traffic congestion control system which automates traditionally manual traffic overload controls with a centralized network element. The system builds on the near-real time centralized control structure developed for Dynamically Controlled Routing, and naturally complements it by adding load admission to its existing load balancing function.

The paper is organized as follows. Section 2 discusses the main objectives and constraints which have guided the design of the congestion control system. Section 3 presents the system, and evaluates via computer simulation several alternatives for the choice of measurements, controls and feedback control process. Section 4 discusses implementation issues in conjunction with Dynamically Controlled Routing. Finally, Section 5 concludes with a summary of the system's features and simulation results.

2. Objectives and Constraints

2.1 Network Integrity versus Admission Management

Our focus for congestion control is to preempt or limit the propagation of crisis resulting from traffic congestion. These may occur either as a result of the volatility of traffic demand, or as a result of network equipment failures. The objective is to design a system which can rapidly detect such crisis and then protect the network integrity by immediately throttling the traffic demand admitted by the network to safe levels. Hence, the primary goal is the efficient utilization of the network.

Considerable attention is currently being devoted to admission control, particularly in the context of ATM switching and SONET transmission [11]. The premise of the efforts is that network resources such as bandwidth, buffers, and processing capacity are scarce. Intelligent admission policies are devised to manage fairly the allocation of these scarce resources amongst competing services while maximizing the expected reward to the telecommunications network operator. We do not attempt here to follow this reasoning. Rather, we assume that network provisioning is sufficient to make admission control an irrelevant issue under all but very exceptional circumstances. This is supported by the business rationale that a telecommunications service provider cannot survive in the marketplace otherwise, particularly if dissatisfied customers can walk away to a competitor. From this viewpoint, we assume that traffic congestion is an exceptional event, hence under which fairness or service quality cannot be the main issues as they have become unacceptably bad anyway. We assume that the main issue must then be to keep the network serving as much demand as possible.

2.2 Crisis Management or Post-Catastrophe Recovery

Systems currently supporting traffic congestion control centralize a large collection of operational measurements in a network management center and allow remote activation of network element controls [6,7,10]. The operational measurements are analyzed by these systems, typically via threshold mechanisms, and potential problems are flagged to network managers. Sometimes, corrective actions are also suggested, but decisions generally lie in the hands of network managers.

The extensive operational measurements and human analysis limit the response time of these systems to the order of minutes. Considering the evolution of the telecommunications environment, this makes them good post-catastrophe recovery support systems, but they increasingly cannot preempt or cope with congestion as it develops. Such fast reaction requires that human intervention be eliminated from the active control process. This can increase the speed of the feedback process from the level of human decision making to that of electronics, hence matching the time scales at which network elements may generate and propagate congestion. It requires also that operational measurements be dispatched in near-real time. Otherwise, reacting fast is purposeless, and near-real time comprehensive operational measurements would impose too much burden on the network elements which must deliver them.

Note that part of the actual motivation for comprehensive operational measurements is to compensate for their slow delivery. Comprehensive data help network managers fully assess a situation, and hence make better decisions. This is important if the individual pieces of data are old, hence potentially inaccurate, and if the decisions remain enforced for a significant time before new data is available to allow corrections. Comprehensive operational measurements are, however, not critical if their delivery and the feedback process are near-real time because any imprecision can be quickly corrected. Note also that automating the active congestion control process does not mean eliminating human intervention. Human knowledge is mandatory for setting and supervising its behavior.

2.3 Centralization or Decentralization

A congestion control system imposes processing overhead on the elements that it controls. Since under congestion this overhead translates directly into a loss of equivalent useful work, it must clearly be minimized. This can be achieved first by placing the burden of congestion control away from the controlled elements and, second, by eradicating congestion as close to the source as possible, so that a minimal number of elements are affected. This latter requirement implies either rapid communication from every network element where congestion may be experienced to every network element which may have to be throttled to control it, or centralization so that controls at a given network element can immediately reflect data gathered at another. Given the overload on the controlled elements and the more stringent communication requirement for decentralization, centralization appears the most reasonable, perhaps the most cost-effective, solution.

Besides, a centralized system may be supervised by network managers from a single access point. This is a major advantage from an administrative point of view, as witnessed by all major existing network management systems.

2.4 Reliable Measurements

Knowledge of the volume and distribution of call attempts is useful for congestion control. It helps pinpoint the sources of the congestion, and allows controls to be immediately and precisely tailored to the desired level of throttling. However, almost by definition, the volume and distribution of call attempts in congestion are unknown. Since they differ significantly from those for which the network is engineered, network planning or servicing data are inadequate to estimate them. Of course, given enough time and assuming that call attempt arrival processes are stationary, reliable estimates can be constructed from measurements. This is fine if one is prepared to wait while the congestion worsens, but it is incompatible with immediate reaction. Furthermore, assuming stationary call attempt arrival processes is highly questionable in congestion; they may change quickly as the congestion evolves. This makes past measurements rapidly obsolete for estimating current call arrival statistics, and thereby precludes accurate estimation. Overall, although desirable, accurate knowledge of call arrival statistics does not appear feasible in near-real time.

Contrary to the volume and distribution of call attempts, the holding time or in general the resource consumption per successful attempt is stationary under congestion. People may call to inquire about relatives in the advent of a disaster, book tickets for a popular event, or any other matter, but their purpose is well-defined. As this purpose may differ from that for which the network is engineered, the holding time may be initially unknown, and hence may have to be measured. However, it is worthwhile to do it, or at least assume that the resource consumption per successful attempt is constant.

† This argument does not consider the cost of the centralized structure itself, however. We will show in Section 4 that centralized structures developed for traffic routing purposes may essentially be re-used for congestion control purposes, at no extra cost.
3. Technical Description

3.1 Overview

As discussed in the preceding section, the objective of the congestion control system is to achieve an optimum use of both the network and of the bottleneck resources in event of congestion. That is, the system must regulate the excess traffic demand and admit into the network a sufficient yet not excessive call flow. Following [12], the architecture we suggest allows the close control of this call flow through centralized monitoring and near-real time feedback from the potential bottleneck elements. Figure 1 illustrates the architecture and congestion control process, and Figure 2 illustrates the timing of operations. Every feedback cycle, the network processor (NP) collects congestion control process, and Figure 2 illustrates the timing of operations. Every feedback cycle, the network processor (NP) collects utilization estimators from selected network elements, either in the form of direct occupancy figures or as indirect measurements like call overflow rate. In Figure 1, the examples of a trunk group to another network and a group of lines to a commercial subscriber are given as potential bottleneck elements, but the elements could as well be switch CPUs, SS7 channels, etc. If the occupancy or overflow of one or some of these elements is found to be above a predefined threshold, an alert state is initiated. When the alert is on, a call flow control parameter is calculated, whose value is a function of the measurement just made and of the last control applied. The control action, called call gapping, is of an absolute nature: it is a minimum inter-arrival time imposed between calls admitted into the network; it is never relative to the offered flow (e.g., a percentage). The control parameter, once calculated, is communicated to the switches where it is selectively applied to the calls outgoing to the congested destination. The whole process of data collection, parameter calculation, and control application lasts about two seconds within a 10-second feedback cycle (also called update cycle). It is applied continuously, until the level of offered traffic no longer threatens the network integrity.

In the following subsections, we describe the feedback loop: data collection, the calculation of the control parameter, and the application of the control at the switch level. Then, results from computer simulation are presented to motivate several of the decisions which have guided the design as well as to illustrate the performance of the system under various conditions.

3.2 Data Collection

The first step of the feedback loop is to collect the resource utilization measurements from the potential bottleneck elements. In principle, this data can be collected on a permanent basis, whether or not the elements are saturated. In practice, however, the data may be filtered by the switches and be only communicated on an exception basis, when the switches detect (via threshold crossing) saturation and danger of congestion. The switches then would send the relevant data to the NP until the crisis is considered resolved.

As mentioned in Section 3.1, resource utilization estimators may be of two kinds:

- **Direct occupancy**: switch CPU consumption figure, SS7 channel consumption figure, etc.
- **Overflow**: calls overflowing a target subscriber's line(s) (the target subscriber being a radio station, a ticket sales agency, etc.), a failed trunk group, or a trunk group towards external switches.

To understand the choice of these estimators and the operation of the system around them, consider first the case where the bottleneck resource is a group of lines of a commercial subscriber. Figure 3 shows the average occupancy $\rho$, the overflow rate $\omega$, and the probability that an incoming call is accepted $P_a$ as a function of the incoming call flow $\lambda$, for a group of 48 lines with a mean call holding time (MHT) of 180 seconds. From this graph, it can be seen that the subscriber lines can be kept practically 100% busy if they are offered a flow $\lambda > 0.5$ calls/sec. Such a level of utilization is obviously desirable for links (either subscriber lines or trunk groups) as long as it does not mean the waste of other resources in the network, as it becomes the case when $P_a \to 0$. Now, $\rho$ is almost insensitive to $\lambda$ at this level, which makes it a poor source of feedback for the tight control of $\lambda$. The overflow curve, however, shows that $\omega$ does become sensitive to $\lambda$ as $\rho$ comes to saturation. It is therefore a good estimator of $\lambda$ in the desired region of operation, and a target overflow rate, $\omega_T$, can thus be fixed in advance in a region where $\partial \omega / \partial \lambda \equiv 1$ (the maximum) and $\lambda$ is not large enough to impact negatively on the remaining of the network. The target overflow rate will be the point of operation of the system for this particular resource, and the system will try to keep the actual overflow rate around this value in times of congestion.

![Figure 1](image1.png)

**Figure 1** Architecture and data flow of the congestion control system: (1) occupancy or call overflow data is collected, (2) call flow control parameter is calculated, (3) parameter is sent as a recommendation to the switches.

![Figure 2](image2.png)

**Figure 2** Timing of the congestion control operations. $T_{FU}$ is the feedback cycle (10 seconds); $T_D$ is the time for computation and for communication of the congestion control recommendations (2 seconds).

![Figure 3](image3.png)

**Figure 3** Occupancy ($\rho$), probability that a call is accepted ($P_a$), and overflow ($\omega$) as a function of the call flow ($\lambda$) offered to a group of 48 lines. The offered traffic is Poisson and call holding times follow an exponential distribution with mean 180 seconds.
Consider now the case of switches. In a stored-program controlled switch, only a part of the CPU time can be devoted to call processing, the remainder being consumed by various background tasks. The switch will operate efficiently as long as the call processing occupancy is kept below a maximal level (85%, for instance). Since this point is below saturation, contrary to the case of lines and trunk groups, the call processing occupancy $p$ is sensitive enough to the call flow to become a good source of feedback $(\delta p/\delta \lambda > 0)$. A call processing target occupancy $p_T$ can be fixed, equal to the maximum call processing allowable $(p_T = 0.85$, to take the same example as above), and the system will try to keep the operation of the switch around this value in times of overload.

In summary, the choice for the source of feedback in the congestion control system presented here relies on the observation that the data must be sensitive to the offered traffic in the region where the traffic should be throttled. This is achieved by using occupancy for resources which must be maintained below saturation and overflow for resources which operate best when saturated.

### 3.3 Control Parameter Calculation

As soon as an occupancy or overflow measurement for a resource reaches a predefined threshold, the NP recognizes that the resource is under congestion and sends to the switches an initial value, $T_0$, for the call gap. The evaluation of $T_0$ should be matched to the target occupancy or overflow rate of the specific resource. However, it is subjected to two uncertainties: (1) the number of switches participating in the congestion, and (2) the mean call holding time. Both parameters may vary widely from one congestion situation to another, so they cannot be fixed in advance. Therefore, in order to bring the system in its region of operation, $T_0$ will be based on extreme values of the unknown parameters. That is, if $\lambda_{\text{max}}$ is the call flow necessary to obtain the occupancy $p_0$ (or overflow rate $\alpha_0$) under the smallest possible MHT (say, 10 seconds), and considering that this call flow can come from only one switch, then:

$$ T_0 = 1/\lambda_{\text{max}} $$

is a minimum assessment of $T_0$, allowing the system to always start its feedback in the planned region of operation. Note that in certain cases (for instance, the CPU load of a switch) the MHT is irrelevant, which simplifies the estimation of $T_0$.

Once the switches have received and implemented $T_0$, the call flow to the congested resource is under control and must be maintained around its optimum value throughout subsequent feedback cycles. The cycle-to-cycle calculation of the inter-arrival time depends on the measurement which applies to the resource. Take first the case of direct occupancy measurement. Suppose that at cycle $n$ the NP receives $p_n$ as the measured occupancy of the congested resource. At cycle $n-1$, the NP had sent to the switches $T_{n-1}$ as the call gap control. Assuming (as in Section 2.4) that each successful call imposes a constant load on the target resource, $p_T$, can be considered as inversely proportional to $T_{n-1}$. Then the next inter-arrival time can be simply calculated as:

$$ T_n = \text{max}(T_{n-1}p_n/p_T, T_{\text{min}}) $$

Experiments prove that the approximation involved in this adjustment is compensated by the high iterative rate of the feedback loop. The parameter $T_{\text{min}}$ is a floor value preventing $T_n$ from drifting too low if the incoming traffic is below what is necessary to achieve $p_T$, but is still threatening to network efficiency. In this case the formula brings $T_n$ to $T_{\text{min}}$ and keeps it at this value, so that it is ready to respond if the traffic increases again. Since $T_{\text{min}}$ is calculated as the minimal gap value to maintain the target under extreme circumstances, then $T_{\text{min}} = T_0$ can be considered a good floor value.

In the case of feedback on overflow measurement, the update formula becomes:

$$ T_n = \text{max} \left\{ \frac{1}{(\Omega_n + \Omega_{\text{max}})T_U + 1/T_{n-1}}, T_{\text{min}} \right\} $$

where $\Omega_n$ is the target overflow, $\Omega_{\text{max}}$ is the overflow measurement at cycle $n$, $T_U$ is the feedback cycle length (10 seconds), and $\lambda$ is the number of switches participating in the congestion. This formula translates the measured overflow into a per-switch call rate, where the inter-arrival time is directly adjusted from the difference between the target rate and the observed rate. The value of $\lambda$ is a priori unknown, it may vary with time, and it has direct effects on the speed of convergence to the target. If an estimate of $\lambda$ is too large, then the overflow rate converges only slowly to the target value; if it is too small, then the system may be unstable. Knowing $T_U$, $\lambda$, and $\Omega_{\text{max}}$, it is possible, however, using linear regression or linear systems parameter identification theory, to obtain a proper estimate of $\lambda$ and to adjust it dynamically as the congestion evolves.

The call gap control stops being applied when $p_0$ or $\Omega_n$ drops below a predefined threshold. This threshold is set lower than the activation threshold to provide hysteresis in the control process.

After the control parameter $T_n$ has been calculated, some additional treatment must be applied at the switch level to implement it. The implementation must take into account the various release times already calculated at the moment the switches receive $T_n$. As an example, consider a call that originates at time $t$ in a switch where the gap since the last launched call is supposed to end at time $t_i$. If $t < t_i$, the call is blocked, but otherwise a new release time, $t_{i+1}$, must be calculated after the call is launched: $t_{i+1} = t_i + T_n$. Once this switch receives the next call gap, $T_{n+1}$, it must update $t_{i+1}$ to properly reflect the new prescribed call flow. This is done through the formula:

$$ t_{i+1} = t_{i+1} + T_{n+1} - T_n $$

However, since at any moment there is a latent call flow imbedded within the various release times of the network's switches, modulation of the time axis provides immediate impact of the call gap control on the mean time between calls, which is not the case with the simpler formula. An experimental comparison is shown in Section 3.4. A way equivalent to (4) for updating the release times is through the formula:

$$ t_{i+1} = t_i + T_n + \text{rand}[0,1] $$

where $\text{rand}[0,1]$ is a random number between 0 and 1 which is independently selected by each switch (for its local $t_{i+1}$). The release times are thus randomly spread throughout the network.

### 3.4 Experimental Results

The system described in the preceding section was tested through software simulation. The test case is a 47-node network where a particular subscriber is the target of a focused congestion. The target subscriber has 48 lines to answer its customers and the MHT is 36 seconds, thus giving a service rate of 1.3 calls/sec. Arrivals are Poisson and holding times are exponential. The congestion starts at time 0 and takes the form of a volume of calls rising at the rate of 0.0048 calls/sec/sec towards the target subscriber. At this rate, the target subscriber becomes saturated in less than five minutes. The focused traffic ceases to rise when they reach 13.9 calls/sec (50000 calls/hour). The traffic is normal in the remaining of the network.
Figure 4 shows the progression and the control of congestion with the overflow on the subscriber's lines as the source of feedback. The lower curve, referring to the right vertical axis, is the number of calls admitted in each cycle. The upper curve (left axis) is the line occupancy observed during the cycle, expressed as a proportion of the total number of lines. The time (horizontal axis) is expressed in number of 10-second feedback cycles. The target overflow rate has been fixed at \( \alpha_i = 10 \) calls/cycle. The overflow threshold above which the controls are applied is fixed at 7 calls/cycle. In the first cycles of Figure 4, the occupancy rises sharply as the volume of incoming calls increases. When the observed overflow reaches 7 calls/cycle, the system starts applying time gaps between calls. When it passes above 10, the system restricts the flow of calls, preventing further increase and maintaining it around the level corresponding to the target overflow rate, that is, 23 calls/cycle (the service rate being 13 calls/cycle). The graph shows a very stable steady state (from cycle 75 onwards), in spite of the highly changing traffic level in the background and the noisy nature of call attempts (Poisson arrivals). Allowing a modest overflow gives an occupancy level near saturation for the target subscriber. The graph shows a very stable steady state (from cycle 75 onwards), in spite of the highly changing traffic level in the background and the noisy nature of call attempts (Poisson arrivals). Allowing a modest overflow gives an occupancy level near saturation for the target subscriber.

![System Response / Feedback on Overflow](image1.png)

**Figure 4** System under nominal conditions. Feedback source is line overflow.

Figure 5 shows the case where the CPU load of a target switch is monitored. The CPU load corresponding to the left axis in the figure is the CPU load attributable to call processing, as explained in Section 3.2. Considering the overhead needed for background tasks, the target CPU occupancy for call processing has been fixed at 0.85. Although the call processing time can vary greatly according to the nature of the calls involved, the simulation uses averages of 0.035 sec and 0.014 sec for call set-up and call release, respectively. The behavior of the system is very similar as for feedback on overflow. The system locks onto the target occupancy as soon as the arrival rate can sustain it, and then throttles the arrival rate to maintain a stable steady-state around it.

![System Response / Feedback on CPU](image2.png)

**Figure 5** System under nominal conditions, when source of feedback is the CPU of a switch. The Accepted Calls/Cycle do not include the calls originating from the switch.

It has been observed in Section 2.2 that relatively little information is needed to handle congestion as long as the feedback loop is fast enough to rapidly correct the applied controls. At this point, the sensitivity to the update cycle length could be questioned: will performance still be adequate if 10-second update cycles cannot be provided? This situation has been simulated by forcing the transmission and calculation delay to nearly one complete cycle (9.9 seconds) and leaving the network without feedback every other update cycle (Figure 6). The result is shown in Figure 7. Obviously, for this case, the system can still maintain the target overflow rate without stability problems (compare with Figure 4).

![Alternate timing diagram, to test resilience towards update cycle length.](image3.png)

**Figure 6** Alternate timing diagram, to test resilience towards update cycle length.

In the assessment of a target overflow rate (Section 3.2) there is an underlying assumption on the MHT of the calls towards the congested subscriber. This causes a problem because the MHT is unknown and can vary significantly from one focused congestion to another, since the time that customers spend on the telephone is dependent on the reason why they call. Now, an MHT lower than expected would bring greater variance in the observed overflow. Referring to Figure 3, this means that \( \alpha_i \) would be in effect closer than expected to the elbow of the \( \alpha \) curve, which is outside the planned operating region. Thus, the system must prove itself to be resilient to differences between assumed and actual MHT. Figure 8 shows the performance of the system when the actual MHT is 3 times smaller than the assumed MHT (note that this is an extreme case, as the MHT is then 12 seconds, which is unrealistically small). All the other parameters are the same as those of Figure 4. When the observed overflow reaches the threshold of 7 calls/cycle, the system starts applying time gaps. As the traffic demand increases, it progressively restricts the call flow until it stabilizes around the target of 10 calls/cycle (the service rate being now 40 calls/cycle). The system is thus considered sufficiently resilient to the uncertainty of the MHT.

![Mean Holding Time three times smaller than expected.](image4.png)

**Figure 8** Mean Holding Time three times smaller than expected.

Figure 9 shows the case where the release times are updated with Formula (5) instead of Formula (4). The burstiness of the call
admission curve is caused by the delay with which the new gap value takes effect. During this delay, feedback is virtually eliminated. The system is shortening or lengthening the gap as if the values previously recommended were insufficient. By the time the delay is over, the recommended control has become completely inadequate. Formula (5) does not cause this effect (see Fig. 8, for instance) because the impact of the new gap is immediate.

![Figure 9 System's behavior when release times are updated through Formula (5).]

4. DCR as a congestion control platform

Dynamically Controlled Routing (DCR) is a system (architecture and algorithm) designed by Bell-Northern Research for the efficient routing of calls in a telephone network [1]. A network controlled by DCR is characterized by a high connectivity between its switches, the absence of hierarchy between trunk groups and between switches, and by a central Network Processor (NP) connected by data links to all the switches. Through these data links takes place a cyclic exchange of measurement data and control recommendations. During a fixed time frame of 10 seconds, called an update cycle, the NP first collects routing-relevant data from the switches, consisting of the load state of the switch CPUs and the idleness and overflow of every trunk group. Second, given this knowledge of the network state, the NP picks the best route for the overflowing calls of each origin-destination node pair. Every possible 2-link alternate route is considered in the choice. Finally, the NP sends the preferred routes as recommendations to the switches for routing their overflowing calls. A “Block” recommendation can also be sent if no alternate route is available, and the recommendation computation process in the NP embodies the concept of protective allowance on links to protect the direct traffic.

The architecture of DCR exhibits important features for efficient network management. First, it is centralized. Every update cycle, the NP receives a snapshot of the network state in terms of resource occupancy (trunks, switches' CPU consumption, etc.). All the information is thus concentrated in the NP, making it available for traffic management decisions. Second, DCR is practically real-time. A 10-second update cycle is short enough to react to network's traffic fluctuations, despite the technology trends towards shorter time constants [13]. Third, DCR can effectively re-route traffic around local bottlenecks like trunk failures, small spurious congestions, etc. In fact, DCR already embodies the principles of expansive controls: every possible 2-link route is taken into account to avoid a congested area, routes are limited to two links to prevent snowballing, and the protective allowance gives priority to direct traffic to increase the network throughput. Fourth, DCR provides an end-to-end view of the network, data which is much more suitable than raw trunk group occupancies for congestion analysis.

The congestion control system presented here therefore appears simply as a natural extension of DCR. Using the architecture already in place, it complements the traffic management capabilities of DCR by providing automatic control over network access. The practical implementation of the congestion control system within the DCR architecture is a simple process of providing additional routines which interwork with those already in place. No structural changes are required.

5. Conclusion

A feedback congestion control system has been presented which fits naturally into the current architecture of Dynamically Controlled Routing. The system extends the network management capabilities of DCR from its actual expansive controls to restrictive controls, taking advantage of its centralized and real-time architecture.

The system is an efficient tool for preventing the overload of a network with useless signalling. Whether the feedback is based on direct occupancy measurement or on overflow of the bottlenecks resource, the system reacts promptly to congestion by immediately blocking at source the excess signalling and then keeping the resource occupancy close to its assigned target level. It is found that promptitude of reaction and precision of controls are best achieved with simple measurements and short update cycle. Resilience to variations of update cycle length and mean holding time are satisfactory within a practical and reasonable range.

Call gapping is the means through which the system's controls are implemented. This is a well-known feature, standard in many SPC switching equipments. Therefore, interface to switching equipments should not pose excessive difficulties.

Finally, the system can be presented as a natural extension of the DCR, which has already proven to be an effective expansive control system. Using its elements and its architecture, the Congestion Control System completes DCR by providing restrictive control capabilities, and thereby enables full traffic management within a unified system.

References