Delivering Predictable Quality in Saturated Networks

Predictable Network Solutions

Neil.Davies@pnsol.com

September 2003

1 Introduction

To assure that a network application behaves correctly, there is a need to transport its data packets with at least a minimum quality. Although delivering such quality is not usually an issue for a single application instance on a closed network, adding more instances, applications and users soon raises the potential for the demand on the network to substantially outstrip the supply. Therefore there is the potential for saturation to occur. In this paper we are explicitly considering saturation of the network transmission resources associated with network elements.

We do not use the term "saturation" solely to refer to the condition where the average offered load is at (or above) 100% of the capacity. Saturation occurs for other reasons: oscillatory effects of elastic traffic sources[1] and variation of demand from statistically multiplexed sources as well as from external correlations, random or otherwise. The observed load at such a saturated resource may appear low when averaged over large time intervals[2], and would consist typically of periods of heavy offered load interspersed with idle periods.

For network quality to be effective it has to be delivered during periods of saturation, periods of low load, and the periods of transition between them. We consider quality to be more than just delivering transported data with assured throughput, loss and delay characteristics. That viewpoint would be sufficient if the interest was in delivering quality to a single, well behaved, packet flow treated as a single entity throughout the network. Our interest lies in delivering quality to multiple flows during a wide range of network operating conditions. This means considering the interactions between the packet flows, how packet flows can be aggregated (and de-aggregated), and how they should be treated when they "misbehave" (their demand exceeds their allocation). This knowledge should not be from a post-mortem analysis of the (failed) system, but derived from a consistent framework that permits planning, monitoring and management , i.e. it should be predictable.

[1] Elastic sources are ones which increase their demand until they infer, typically through loss, that they have saturated a resource on their end-to-end path. They then reduce their demand for some period before repeating the cycle. TCP is the prime example.

[2] A typical interval would be between 30 and 300 seconds.

1.1 Prerequisite Properties for Delivering Predictable Quality

To achieve this predictability, over the wide range of network operating conditions it is necessary that every packet flow has some measure of:

ISOLATION: The independence of operation of a packet flow from the effects of variation in the offered load of other flows. This property is needed to permit meaningful capacity planning. Without some measure of isolation any quality allocation could not be assured during actual operation.

FAIRNESS: The equitable treatment of any constituent sub-flows within a packet flow. It is not possible to manage all the packet flows within a network as individuals; there will, out of necessity, be aggregation (and de-aggregation) points where multiple flows will be (at least logically) considered as a single flow. Any quality treatment of that aggregated flow needs to be fair, in that all the constituent sub-flows experience the same quality.

DIFFERENTIAL TREATMENT: That packet flows can be delivered differing amounts of throughput, loss and delay; this is a prerequisite to delivering differential quality.

SATURATION BEHAVIOR: The control over the quality experienced by a packet flow, as its offered load approaches (and even exceeds) saturation. Similarly, there is a need to manage the system as it saturates.

EFFICIENCY: This is not necessarily a per packet flow property, but it is a constraint. Whatever approach is taken to achieve the above, it cannot be done by wasting scarce resources such as communication link capacity.

All of this needs to be achieved in addition to the control and management of throughput, loss and delay.

1.2 Our Viewpoint

The method that we have adopted in this paper is to approach the applications' requirements from the point of view of the quality that they require, combined with the quantity of that requirement. This *quality centric* view underpins our approach to reasoning about and delivering predictable quality of service.

In traversing the network, each packet experiences delay, and possibly loss. Every packet handling operation, including transmission, decreases the packet's experienced quality; put simply, every operation increases quality degradation. Quality of Service (QoS) is often seen as a process that gives a packet flow something special. In reality it is about controlling the degradation that the packet flow suffers during its transportation through the network.

Initially, we are going to examine the ways in which degradation occurs, isolating the sources of degradation over which the network elements could exercise control. We will then be in a position to extract some invariants and relationships as they relate to quality degradation. It will then be possible to apply these findings to examine existing approaches to QoS within network elements. Finally, we will be in a position to examine how the above prerequisite quality properties can be met both within a network element and the end-to-end network.

2 Introducing ΔQ

In traversing any network, the quality degradation experienced by a packet flow will consist of a fixed (immutable) part and a variable part. The fixed proportion arises from such actions as packet serialization/de-serialization and propagation delay/loss[3]. The variable portion of the experienced end-to-end quality occurs because a packet arriving at a node will find a variable number of packets already in the system. The fate of individual packets is too fine-grained a view from which to derive useful measures. We will focus on packet flows and their quality degradation using aggregate measures over some suitable time interval. We will use ΔQ as shorthand for these properties, their composite effects and their mutual interactions.

2.1 Properties of ΔQ

Quality degradation (ΔQ) only ever increases. This holds true whether ΔQ is for an individual flow as it traverses the network, or for the collective effect of many packet flows at a single node. Also, in any system *quality degradation is conserved*, at both a system and individual packet level. At the packet level, once a packet has been delayed that delay cannot be "undone", as is clearly the case when packet loss occurs[4]. At a system level although one packet flow may be given a lower delay (or loss) than others, there will be a balancing effect on other packet flows. This property is more clearly illustrated by examining the effects of different scheduling policies in the network element.

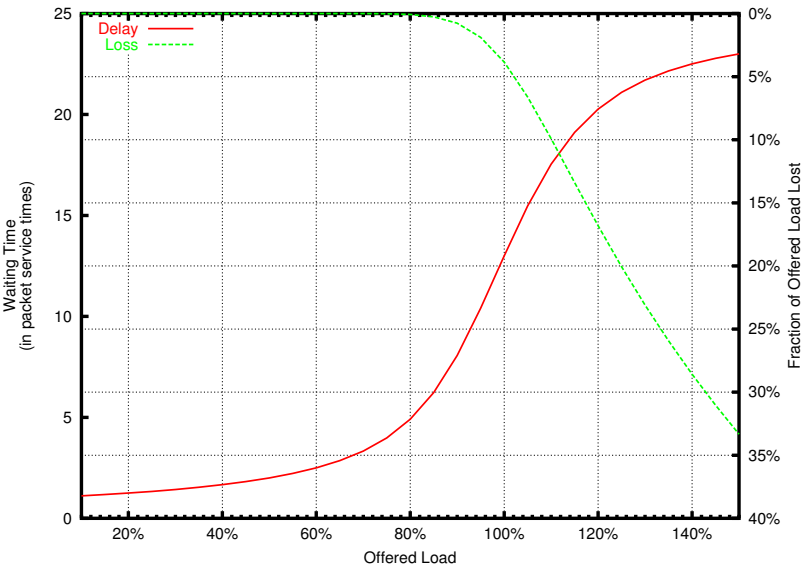


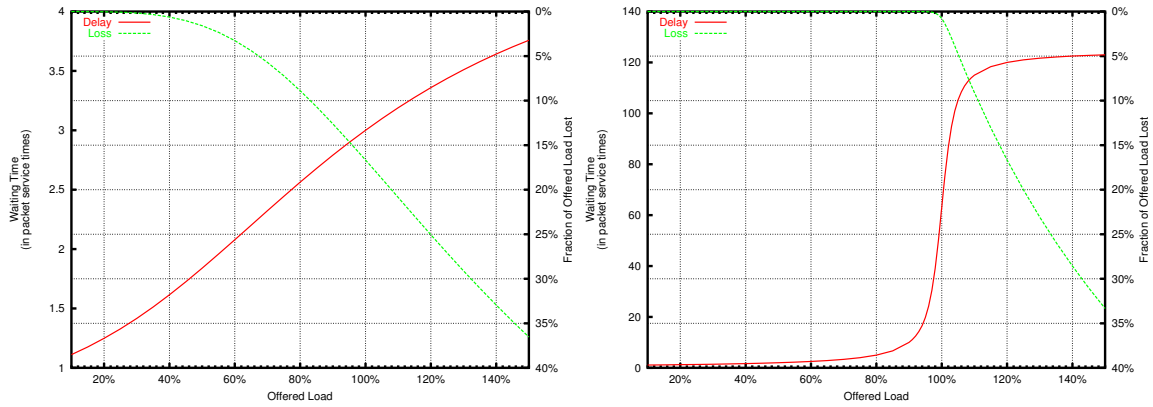
Figure 1: Changes in Quality by Offered Load (25 buffers)

[3] Although losses occurred during transmission are negligible across fixed network connections, this is not necessarily the case where wireless transmission is involved.

[4] It can be retransmitted, but that also has a cost and will, inevitably, decrease the available resources for use by other packet flows in the system.

2.2 Change in Delivered Quality at a Single Node

Figure 1 describes the change in delay and loss (in a finite FIFO) as the offered load[5] approaches saturation. In choosing the number of buffers to allocate (25 in Figure 1), a choice has been made



(a) 5 Buffers

(b) 125 Buffers

Figure 2: Resulting ΔQ with varying number of buffers

as to the ΔQ relationship with load. In Figure 2, the effect of varying the number of buffers can be seen. Increasing the amount of buffering reduces the load at which the onset of loss occurs. This

	Offered Load					
	80%		90%		95%	
# buffers	loss rate	delay (PST)	loss rate	delay (PST)	loss rate	delay (PST)
5	1.6×10^{-2}	2.6	1.2×10^{-1}	2.8	1.5×10^{-1}	2.9
25	7.6×10^{-4}	4.9	7.7×10^{-3}	8.1	1.9×10^{-2}	10.4
125	$\ll 10^{-10}$	5.0	1.1×10^{-6}	10.0	8.2×10^{-5}	19.8

Table 1: ΔQ at selected offered loads

is not at zero cost as the "penalty" for this is that the overall delay has increased, as illustrated in Table 1. Although we have used here a finite FIFO queue as our example, the general relationship that is brought out here exists irrespective of the scheduling strategy.

2.3 Differential Allocation of ΔQ

Given that quality degradation cannot be avoided, how can it be differentially apportioned? In Figure 3 there are two classes of arriving traffic—urgent and non-urgent[6]. The total load is kept constant

[5]The negative-exponentially distribution has been used here for both the arrival and service processes. This choice has been made purely because it gives the simplest formulation for the purposes of illustration. All other arrival and service patterns would yield the same general trend.

[6]Urgent packets being serviced before any non-urgent one.

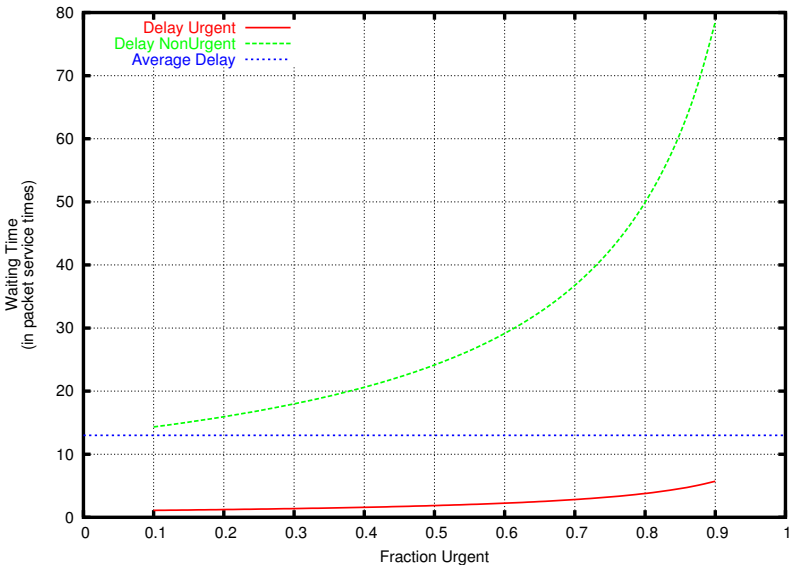


Figure 3: Differential Allocation of ΔQ (offered load at 100%, 25 buffers)

(100% in this case), and the ratio of urgent to non-urgent traffic is altered. Throughout, the loss rate remains constant for each packet flow at 3.8% of their offered load. The delay for the urgent traffic varies from 1.1PST[7] (at 10% urgent traffic) to 5.7PST (at 90% urgent traffic). This illustrates that it is possible to give very good delay treatment to urgent traffic, even though the system is operating at saturation. At what cost? The system's average delay is the same at 13.0PST; the non-urgent traffic is now suffering very large delays corresponding to several times the total amount of buffering (varying from 14.3PST with 10% urgent traffic to 78.5PST at 90% urgent traffic). This apparently dramatic increase in the delay suffered by the non-urgent traffic is a direct consequence of the conservation of ΔQ; the loss for both streams is remaining constant, and as the ratio changes, the balancing amount of the delay has to be shared over a decreasing number of non-urgent packets. Consequently, their delay increases.

2.4 Two Degrees of Freedom

In any queueing system there is a relationship between loss rate, throughput and delay[3]. Truly unlimited buffer capacity would allow for the possibility of a zero loss rate, and in that case throughput would determine delay. All practically realisable cases come with finite buffer resources. Managing loss is an important step towards managing the total delay in the system. In some cases, delay targets for applications may not be achievable without some loss.

- For a fixed loss rate, reducing delay means that throughput must fall.
- For a fixed throughput, reducing delay implies an increase in loss rate.
- For a fixed delay, reducing loss rate will require a reduction in throughput.

[7] Packet Service Time

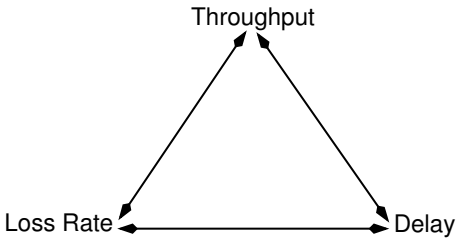


Figure 4: Illustration of the Two Degrees of Freedom

3 Assessing Existing Approaches

Having illustrated the underlying principles, in this section we will examine the basic properties of:

- Finite-FIFO
- Weighted Fair Queuing Family
- Cherish/Urgency Multiplexer

In performing this comparison we will make the usual assumptions common in this type of analysis: the system is in steady-state and the arrival and departure are Poisson point processes. Such approximations have been criticised as not representing a sufficiently accurate model of "reality". However, they do permit an analytical comparison of some of the practical network issues we review in Section 6.

It is important to note the central role that the quantity "Packet Service Time" takes in the analysis of any scheduling mechanism. In presenting these results we have chosen a normalized form; representing load as percentage of capacity, loss as a percentage of the offered traffic lost, and delay as the number of packet service times (PST). This approach encapsulates the two other common factors, packet size and service/transmission rate, into a single quantity. Every packet must experience at least a packet service time of delay for every network element that it passes through. At low transmission rates this can be the dominating quality degradation factor. In data networking, such packet service times can vary from a fraction of a microsecond (for a small packet on a $2.5\text{Gbit}\,\text{s}^{-1}$ backbone) to over $1/4$ of second (for a maximum size packet over dial-up modem at $33.6\text{kbit}\,\text{s}^{-1}$). It is this emergent property that fundamentally limits the usefulness of bandwidth allocation mechanisms.

3.1 Finite FIFO

The simplest and most often used discipline is a finite FIFO queue which discards incoming packets when full, as illustrated in Figure 1. A FIFO does not offer any isolation between packet flows or any differential quality treatment. Although it is often referred to as being "fair", such fairness only exists at low loads. The saturation behavior is dependent on the properties of the constituent packet flows, giving large variations in such things as burst-loss and jitter, and the delivered quality being dependent on the relative phase of arrivals and departures.

Practically, this means there is little predictability during saturation, hence the trend to vastly overallocate network resources. Of particular interest is the effect of long-lived, rate-adaptive connections,

such as bulk-data TCP transfers. Their feedback mechanisms continuously seek the maximum available throughput by pushing the system into saturation before backing off. This on/off saturation of the FIFO causes loss to be observed in other packet flows which has, traditionally, caused people to allocate more buffering. As we have illustrated above, from the point of view of managing and predicting delay, this is not necessarily an optimal strategy.

Although the finite FIFO offers some degree of fairness, it does not offer any isolation or differential treatment. Its saturation behavior is fixed and only operates at a "system" level. It can make effective use of the outgoing transmission resource if there is sufficient buffering and the applications can accept the consequential quality effects.

3.2 Weighted Fair Queuing Family

The lack of isolation and differentiation inherent in the FIFO scheduling approach has led to the development of the Weighted Fair Queues (WFQ) family of scheduling algorithms. In weighted fair queuing the incoming traffic is allocated to one of several queues. The underlying rationale being that, as a worst case, each of these queues will receive a fixed fraction of the outgoing link capacity. The underlying theoretical basis is that of generalized processor sharing, where the capacity is distributed in small quanta to each queue. It is this common basis on which we will base our critique. As packets are discrete, generalized processor sharing cannot be realised in practice. This has lead to several variations being developed; each differing in how the necessary accounting is realized and how the sharing of spare capacity is managed.

Each of the queues within a weighted fair queueing system can be viewed as a finite FIFO in which the service rate varies between some minimum and maximum, the minimum service rate being set as a relative weight and the instantaneous service being dependant on the current demand placed on the rest of the queues. The maximum service rate would typically be 100% of the outgoing link capacity.

Typically, in designing the network, a particular application would be allocated a particular bandwidth; this being its worst case service rate. From our quality-centric view, this raises the question: "What is the delivered quality that such a packet flow will experience during operation?". We will examine this question from two points of view: firstly, the system operating with some spare capacity and secondly, the system operating in saturation. For the moment we will assume that the packet flow originates from a well-behaved, non-elastic network source.

In the case where the rest of the arriving traffic is not pushing the system into saturation (i.e. there is "spare" service capacity), the packet flow experiences more service rate than its assigned minima. In this case, when calculating the ΔQ that this flow experiences, not only is the offered load a smaller fraction of a finite FIFOs capacity, but the associated packet service time is also smaller. For example, consider a packet flow that is operating at 95% of its configured WFQ bandwidth. It arrives at a node which has sufficient excess capacity to serve that queue at twice its configured rate. The combined effect dramatically reduces the experienced ΔQ. To continue the example not only is virtual FIFO operating at 47.5% load, the underlying packet service time is also twice as fast.

Now consider the saturation case where the application is offering the same absolute load, 95% of the configured rate, with all other streams taking up their allocations. Using the 25 buffer finite FIFO model presented above this would lead to a loss rate of about 1.9% and a delay of about 10.4 PST. This seemingly small difference in the behavior of other streams has lead to the loss rate going from effectively zero to 1.9% and delay increasing by a factor of 13.

	Effective Offered Load	Loss Rate	Delay PST	Effective PST	Experienced Delay (100 byte packets)
"spare" service (40kbits^{-1}effective service rate)	47.5%	$\ll 10^{-8}$	1.9	20ms	38ms
saturation service (20kbits^{-1}effective service rate)	95%	1.9%	10.4	40ms	416ms

Table 2: Summary of WFQ Example

This is best illustrated with a more concrete example. Consider a stream of 100-byte packets with a configured, worst case rate of 20kbits^{-1}. The above scenario (summarized in Table 2) would give an average delay of just over 400ms (1.9% loss) at a node in saturation, compared with about 38ms ($\ll 10^{-8}$ loss) at a node in the "usual" state.

Weighted fair queuing does deliver a measure of isolation, but the saturation ΔQ behavior is entirely dictated by the allocated bandwidth, each queue being its own finite FIFO. This offers an explanation as to why, when configuring network elements to carry real-time traffic such as VoIP, manufacturers generally recommend substantial over-allocation of bandwidth. To achieve, in saturation, a per-node delay less than 5ms for a bursty G.729 (16kbits^{-1} codec, 76-byte packets), an allocation of about 128kbits^{-1} is needed, an over provisioning by a factor of 8.

Rapid changes in delivered quality can interact extremely badly with elastic traffic sources. As other allocated queues start to use up their bandwidth allocation, the effective load on the finite FIFO can easily exceed its capacity, resulting in heavy burst loss.

In summary, weighted fair queuing manages bandwidth and, in its own terms, can deliver a measure of isolation and differential treatment. The saturation behavior and efficiency issues are broadly the same as the finite FIFO, with the consequential effects on loss rate and delay. Where the bandwidth allocation to a queue is small (and hence the worst case packet service time is large) those effects can be extremely detrimental. Although, at first thought, WFQ would appear to have the similar fairness properties to finite FIFO, the reality is slightly different. In practice the outgoing link capacity cannot be allocated in infinitesimal quanta and individual queues tend to receive service, and hence empty, in "bursts". The loss that a particular flow (or sub-flow) experiences can critically depend on the relative phase of packet arrivals with respect to this burst emptying process. Managing that loss pattern is crucial to TCP's effective use[11].

3.3 Cherish/Urgency Multiplexing

This approach[3, 9] was motivated by the observation of two degrees of freedom. In the queuing/scheduling approaches illustrated above (as in all other known approaches) only one of the three interconnected properties (throughput, loss and delay) is managed, leaving an implicit relationship between the other two. The cherish/urgency multiplexing approach defines two explicit orderings, loss and delay, combining them to create an overall quality partial order.

A1	A2
B1	B2

Figure 5: Cherish/Urgency 2×2 grid

In this relationship, as illustrated in Figure 5, the "A" row receives less loss than the "B" row, while the "1" column receives less delay than the "2" column. This provides an explicit mechanism to represent the quality ordering required for a packet flow. In this subsection we will illustrate the effects this can have on ΔQ for individual packet flows by investigating the four possible combinations of two streams within this 2×2 grid.

3.3.1 Common Loss—Differential Delay

A1	A2
—	—

The arrangement corresponds to strict priority queueing with the A1 traffic being serviced in strict preference to that of A2. In Figure 6 you can see the minimal effect that the presence of 60% capacity A2 traffic has on the experienced delay (left-hand y-axis) of the A1 traffic (offered A1 load on the bottom x-axis, total system load on the top x-axis).

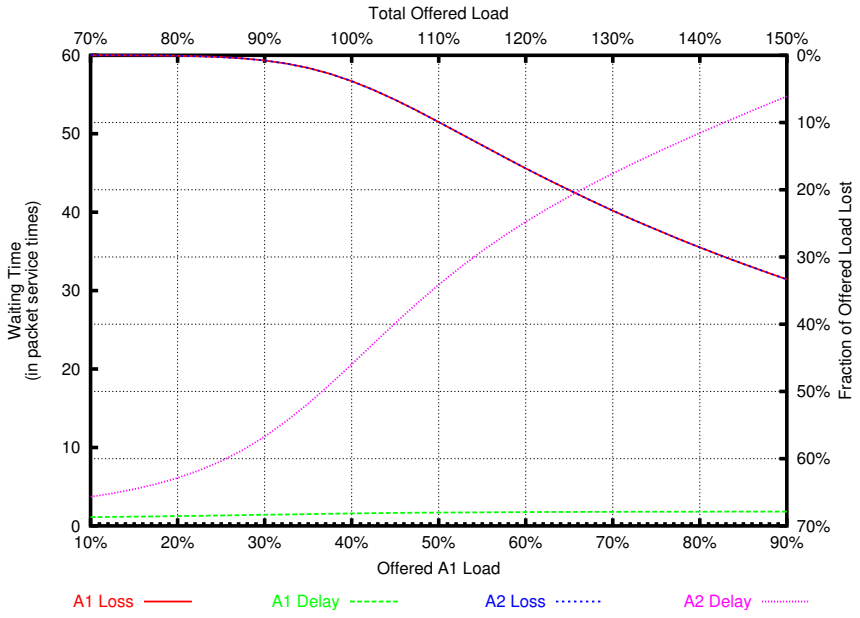


Figure 6: ΔQ with constant offered A2 load (Common Loss—Differential Delay)

We have already seen an example of this arrangement, in the discussion of a single FIFO (Figure 3). Here, as in that example, the loss rates (the right-hand y-axis) experienced by the A1 and A2 traffic are identical, differentially distributing delay while delivering the same effective loss rate to each packet flow.

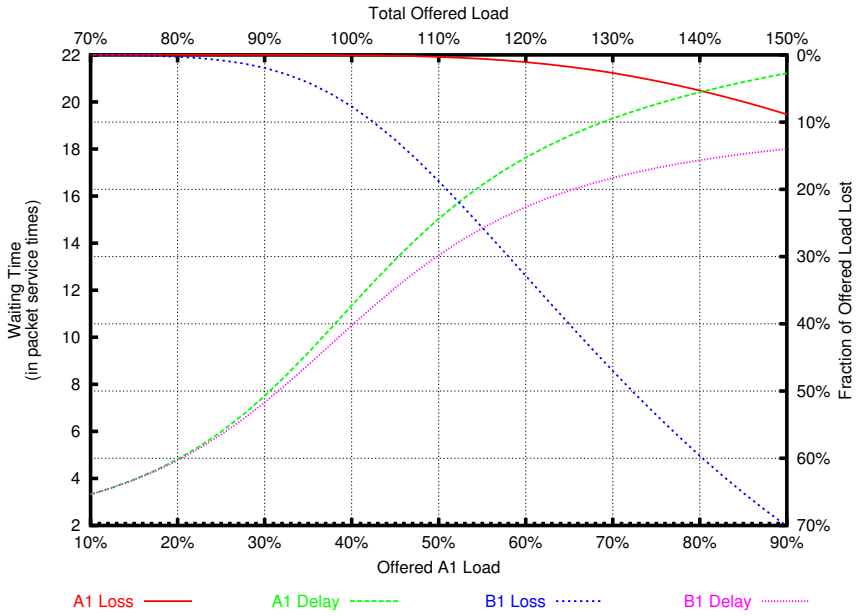


Figure 7: ΔQ with constant offered B1 load (Differential Loss—Common Delay)

3.3.2 Differential Loss—Common Delay

A1	—
B1	—

The arrangement corresponds to a partial buffer sharing approach. Here the loss rates for A1 can be made arbitrarily low by reserving buffers purely for A1 traffic. As such buffering is only used when the shared buffering is full, it is only dealing with a relatively rare case (under normal loadings); consequently small numbers of buffers deliver dramatic reductions in loss probability. However, there is a cost; the A1 stream now always suffers worse delay characteristics mainly because there is more of its traffic present. This is illustrated in Figure 7, in which the B1 traffic is held constant (at 60% of the system capacity) and the A1 traffic is varied.

It is interesting to note that in this configuration (25 total buffers, 5 being reserved for the sole use of "A" row traffic) how well the A1 traffic is treated. It is not until the system is offered 120% load does the A1 stream start to experience a significant loss rate (i.e. 1% or above). Even with just 5 reserved buffers, there is a significant level of loss isolation for the A1 packet flow.

3.3.3 Strict Quality Ordering

A1	—
—	B2

In the arrangement the delivered quality to the A1 traffic totally dominates that delivered to the B2 traffic; it receives both better loss and better delay characteristics. The graph in Figure 8 contains the resulting ΔQ on each packet flow with the buffering configuration being the same as for Figure 7 (25 total buffers, 5 reserved for "A" row traffic).

The loss isolation is just as good as for the previous configuration. The delay experienced by the A1 packet flow now lies in the range 1.1PST (at 10% offered load) to 3.6PST (90% offered load). It is interesting to compare this with the common loss/differentiated delay case (see Section 3.3.1 and

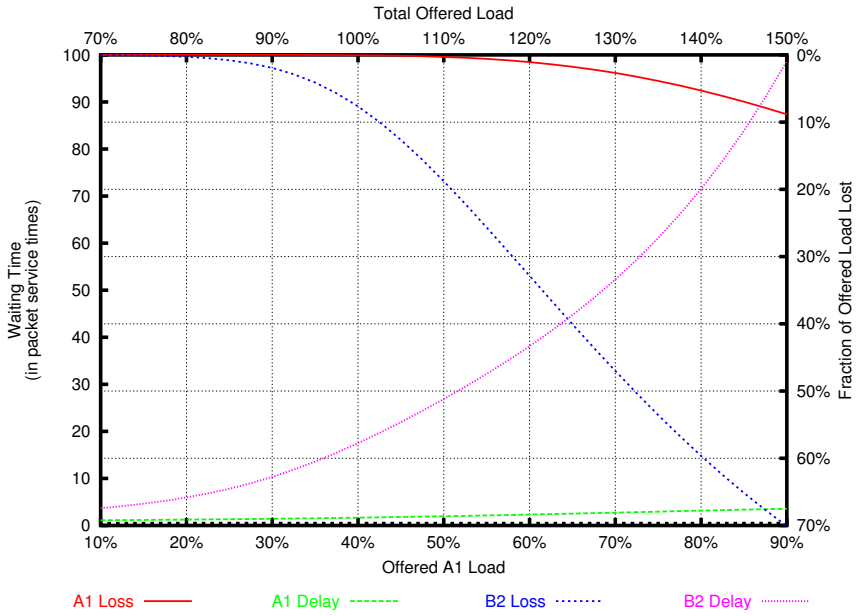


Figure 8: ΔQ for varying A1 traffic with B2 traffic held constant (Strict Quality Ordering)

figures 3 and 6). The delay experienced by the A1 traffic is slightly lower at the 100% total offered load point (1.6PST compared with 1.7PST), while the loss is significantly better (4.7×10^{-4} compared with 3.9×10^{-2}). The A1 traffic delay does not increase as much as the relative load increases reaching 3.6PST (compared with 5.7PST).

The A1 traffic is not only receiving preferential service, it is also "preventing" some B2 traffic from entering, hence excluding that traffic's contribution to the overall delay. The consequential cost is borne by the B2 traffic: Its loss is similar to the differential loss/common delay case, but its delay has grown substantially The worst case delay in Figure 7 was around 18PST and in Figure 8 this rises to 98PST.

The A1 traffic, if left uncontrolled, can deny the applications in B2 service, illustrating the need for the policing/shaping components of the complete network element architecture (see Section 6.1). In certain scenarios such quality domination is desirable. Where the A1 traffic is extremely rare but requires utmost quality in its handling (and as such the consequential effects on B2 are acceptable when A1 traffic is present) or where the system has to cope with varying link capacities and the A1 traffic still must receive high quality treatment.

3.3.4 Differential Loss—Differential Delay

—	A2
B1	—

The last combination represents a case that cannot be replicated by other QoS mechanisms. The A2 traffic here will potentially experience a lot of delay, but very low loss. In Figure 9 the B1 traffic has been kept constant at 60% of the capacity with the A2 load varying, with 5 buffers dedicated to the A-row traffic.

Again the loss characteristics are similar, but the delay distribution is such that the B1 traffic experiences an almost constant delay . The slight decrease in delay that the B1 experiences (dropping from

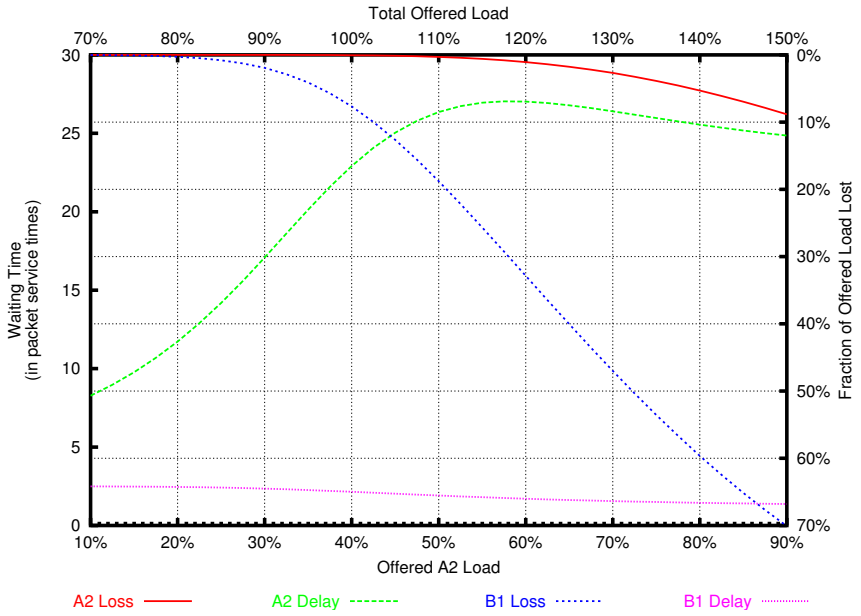


Figure 9: ΔQ for B1, A2 off-diagonal case (Differential Loss—Differential Delay)

2.5PST to 1.4PST) is due to there being less B1 traffic admitted as the A2 load increases.

Often there are keepalives in protocols — packets whose purpose is to probe for continued connectivity. In point-to-point connection testing the permitted delay may be 10 or more seconds, but loss will cause some, usually undesired, recovery process. This combination achieves that low loss, delay insensitive delivery with a minimum impact on the quality of the other traffic.

3.3.5 Generalization to N×M systems

The cherish/urgency multiplexing approach is not just a 2x2 system. It generalizes to an NxM approach and in existing applications is typically arranged as:

A1	A2	A3	
B1	B2	B3	
C1	C2	C3	
			BE

which can be implemented as

A1	A2	A3	—
B1	B2	B3	—
C1	C2	C3	—
—	—	—	BE

This arrangement is chosen as the quality traffic (A1-C3) dominates the best effort (BE) traffic while permitting a combination of quality relationships between the traffic.

The system has a closed formulation[9] and hence it is possible to calculate the ΔQ that all the traffic flows will experience before deploying the configuration. It is also possible to use the formulation to derive a configuration that will fulfill a given set of requirements (loss rate, delay and traffic volume), if such a configuration exists.

3.3.6 Summary of Cherish/Urgency Quality Properties

The cherish/urgency multiplexer can be configured to deliver a variety of differential treatments to packet flows, its saturation behavior is predictable as is its isolation behavior. Its fairness properties, within a individual class, are similar to that of finite FIFO. Creating this differential behavior has left open the possibility of misbehaving packet flows allocated to "high" quality classes to adversely effect the rest, we address these issues in Section 6. The differential approach permits a true "best effort" class to be formed, and such traffic can be used to make very efficient use of the transmission resources[12].

4 Outline of the Mathematical Basis for Cherish/Urgency Multiplexing

The original motivation to understand quality in queuing systems arose from the practical need to configure network elements to deliver a known quality of service. During this study, it became clear that without a consistent approach to loss management, delivering control of delay when under higher offered load was not possible[3, 5].

In the literature, especially when reasoning about bandwidth or delay, continuous system approximations such as "fluid flow" are often made. This is an inappropriate framework for reasoning about loss which is not a continuous phenomenon; on the contrary, it is very binary. Loss can only occur when the system is in a particular, discrete state (i.e. buffer full).

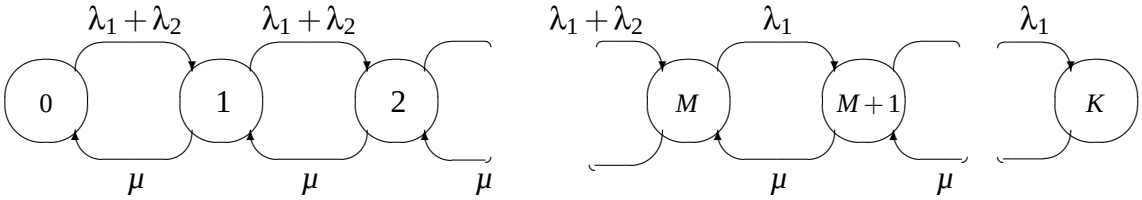


Figure 10: Birth Death Representation of Partial Buffer Sharing

The loss management processes can be illustrated by a simple variation on the birth-death key, as illustrated in Figure 10.

In this formulation the system has two arrival streams λ_1 and λ_2 (of rates λ_1 and λ_2). Packets from both streams are admitted when the system is in states 0 to $M-1$, in states M to K, only traffic from the $\lambda_1$stream is admitted. Traffic is served (when the queue is non-empty) at rate μ. Steady state probabilities can be derived by solving the resulting system of equations[3].

The approach has been developed, as described in [3, 9], to apply to multiple streams of multiple different delivered qualities. The cherish/urgency scheduling is an approach in which:

1. Packets are selectively discarded when the offered load creates too large an instantaneous backlog. This assures a finite overall system delay.
2. Delay can be distributed differentially to the packet flows, the simplest method, priority queueing, being chosen.
3. There is an algebraic formulation that captures the relationship between the delay and loss for all the constituent streams. This permits:

a) the calculation of loss and delay given the input loads and the buffering configuration; and

b) the derivation of a configuration for the system that simultaneously satisfies a given set of requirements for loss, delay and volume.

5 Managing End-to-End Quality

The concept of ΔQ that we have introduced is applicable on both a per-element and an end-to-end basis. In the preceding sections we have illustrated how the differential quality can be delivered at a single point in the end-to-end path. What happens to the composite behavior as the network scales? The results in this section are taken from [14].

5.1 Idealized End-to-End Network

Figure 11 describes an idealized network configuration to permit the assessment of QoS. To model the

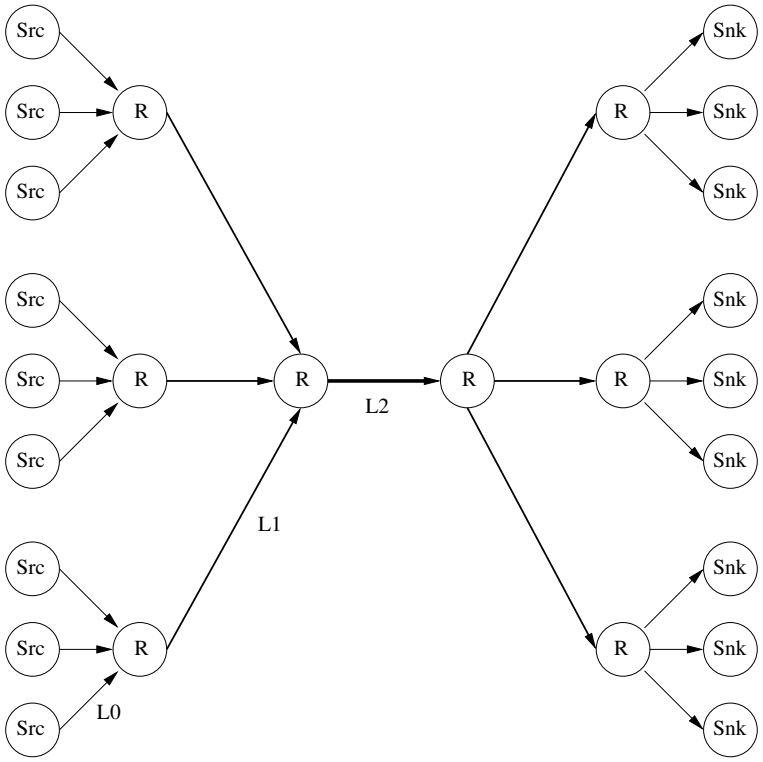


Figure 11: Idealized Network Topology

effects of network traffic on a particular packet flow it is only necessary to incorporate those streams that can influence the packet flow through use of common resources. This work is about investigating the QoS effects of queuing algorithms and, as such, has a number of idealizations: (1) there are no fixed delays associated with the transmission links[8]; (2) the traffic, while generated by several sources

[8]Their length is taken to be zero.

and comprising of several different applications, is assumed to be fixed in composition and load; (3) the MTU size is fixed at 512 bytes.

As to the configuration, the link speeds in this network are: L0 – 512kbits^{-1}, L1 – 1.544Mbits^{-1} and L2 – 6.312Mbits^{-1}. Each node has 200 buffers and the application loads at each source are chosen so that the "L1" links are loaded to 99% and the "L2" link at 73% — elements of the system are in saturation. The traffic mix consists of:

VoIP: Two calls are made from each source to the sink directly opposite, making a 18 total of concurrent calls. The VoIP model chosen was for a G.711 codec which generates 160-byte packets at 50Hz; when additional protocol layers are incorporated[9] this gives a packet size of 218 bytes. Each VoIP call is therefore offering a load of 87.2kbits^{-1}. The quality assessment criteria for the traffic is low delay and jitter while keeping the loss rate acceptable.

RIP: Simulated RIP packets are generated from each source to all the other sinks in the system. This results in 81 concurrent RIP sessions with an overall load of less than 1%. Each RIP source is modelled as generating a 128-byte packet every 30 seconds, giving an average data rate of 34bits^{-1}. The key quality factor here is delivery of at least one packet every 180 seconds. Very low delay is not an issue, but consecutive loss is.

NTP: Again each source is sending NTP packets to every other sink in the system, 81 in all. NTP is modelled as sending a 64-byte packet every 10 seconds, an overall load of less that 1%[10], an overall data rate of about 51bits^{-1}. To accurately and quickly synchronize local clocks, a low jitter is required, loss is not too critical.

HTTP: This traffic is used to simulate a best-effort rate adaptable source. An approximation to the TCP rate-adaptive, and greedy, behaviour was used[11]. The quality assessment used here is data volume. No attempt is made to relate the network seen throughput to the end user perception.

The results presented below are the aggregated results of 10 independent simulations. The application results are the average over all the source/sink combinations; this is possible due to the symmetry in the system.

5.2 Network of FIFO Queues

As a calibration the system was first run with all the queues being finite FIFOs each with 200 buffers, the aggregated results of 10 runs is summarized in Table 3 . As can be seen the applications each received broadly the same ΔQ with the variations being due to effects of the variation packet sizes used by each application and the correlation of the original constant packet rate streams. The loss, while broadly "fair" effectively means that none of the applications are receiving sufficiently high quality to fulfill their requirements. It is interesting to note that service discipline has little effect on the resulting ΔQ, this appears to be an example of the Heavy Traffic Approximation[12].

[9] That is the LAN, IP, UDP and RTP headers of 18, 20, 2, and 12 bytes respectively.

[10] This is a much higher data rate than NTP uses in practice, this was chosen so as to have sufficient events in the simulation from which to draw statistically valid conclusions.

[11] The approach taken was less aggressive than most TCP implementations; after eight receptions of packets without loss the source increases its offered rate, after two losses in a row the source decreases its offered rate.

[12] This approximation states that for a GI/G/1 queue as $\rho \rightarrow 1$the waiting time tends to an exponential distribution characterised by properties of the arrival rate and service rate. See [1] §5.5.1.

Service Discipline	Application	Delay	Delay Stdev	Loss Prob.	Effective Packet Rate	Throughput $bits^{-1}$
Markovian	VoIP	1.552s	0.391s	5.6%	4.76×10^{1}	8.30×10^{4}
	NTP	1.536s	0.450s	7.0%	9.25×10^{-2}	4.74×10^{1}
	RIP	1.538s	0.451s	6.9%	3.08×10^{-2}	3.15×10^{1}
	HTTP	1.570s	0.450s	6.3%	9.17	3.76×10^{4}
Deterministic	VoIP	1.622s	0.369s	5.3%	4.76×10^{1}	8.30×10^{4}
	NTP	1.609s	0.402s	9.7%	8.98×10^{-2}	4.60×10^{1}
	RIP	1.610s	0.402s	9.5%	3.00×10^{-2}	3.07×10^{1}
	HTTP	1.641s	0.407s	7.0%	9.26	3.79×10^{4}

Table 3: Network of FIFO Queues

5.3 Applying Differential Treatment

Two approaches to differential treatment were investigated, Deficit Round Robin (DRR), a variation of WFQ and Cherish/Urgency Multiplexing. In both cases each of the applications was assigned to a separate class. For WFQ/DRR each class was configured with 100 buffers. The bandwidth allocation was performed so as to give sufficient bandwidth capacity for the VoIP, some nominal capacity for the RIP and NTP traffic (which is not bandwidth intense) and the remaining capacity to the HTTP traffic, namely:

- VoIP 35%
- RIP 1%
- NTP 1%
- HTTP 63%

For Cherish/Urgency Multiplexing a total of 200 buffers was allocated with 100 buffers reserved for the “A” traffic. The applications were assigned to the Cherish/Urgency Multiplexer as follows:

VoIP	RIP
NTP	HTTP

The results are summarized in Table 4.

In the WFQ/DRR, as each of the classes is saturated, the resulting quality degradation is high. The zero loss rates for RIP and NTP are not surprising given that there were 100 buffers assigned to each of those classes. The main way in which this approach has fallen short is the delay treatment for the VoIP and NTP, which are both far too high for the applications to work effectively. However WFQ/DRR did perform its assigned function, the delivered bandwidths per class were as configured.

There were two sets of simulations with the cherish/urgency multiplexer, one with Markovian servicing (mirroring the mathematical treatment) and one with deterministic treatment (for comparison with the WFQ/DRR approach). In both cases the quality ordering was as expected. The “A” row traffic (VoIP & RIP) experienced no loss. Where as the “1” column traffic (VoIP & NTP) experienced substantially less delay than the “2” column traffic (RIP & HTTP). The delay for VoIP and

Service Discipline	Application	Delay	Delay Stdev	Loss Prob.	Effective Packet Rate	Throughput bits^{-1}
WFQ/DRR	VoIP	0.538s	0.241s	1.7%	5.00×10^{1}	8.72×10^{4}
	NTP	0.818s	0.413s	0.0%	9.95×10^{-2}	5.09×10^{1}
	RIP	0.820s	0.412s	0.0%	3.31×10^{-2}	3.39×10^{1}
	HTTP	0.977s	0.453s	7.1%	8.93	3.66×10^{4}
Cherish/Urgency	VoIP	0.024s	0.012s	0.0%	5.00×10^{1}	8.72×10^{4}
Markovian	NTP	0.025s	0.013s	6.0%	9.34×10^{-2}	4.78×10^{1}
Service	RIP	1.335s	0.536s	0.0%	3.31×10^{-2}	3.31×10^{-2}
	HTTP	1.353s	0.521s	6.7%	8.70	3.56×10^{4}
Cherish/Urgency	VoIP	0.011s	0.003s	0.0%	5.00×10^{1}	8.72×10^{4}
Deterministic	NTP	0.012s	0.004s	8.6%	9.08×10^{-2}	4.65×10^{1}
Service	RIP	1.543s	0.482s	0.0%	3.31×10^{-2}	3.31×10^{-2}
	HTTP	1.543s	0.480s	7.0%	9.01	3.69×10^{4}

Table 4: Results with Differential Treatment

NTP is substantially less, bringing it well within the acceptable ranges of operation. No bandwidth management is performed in this approach yet the delivered effective bandwidth to the HTTP traffic is practically identical to the WFQ/DRR approach, the elastic HTTP source is seeing just as much long term bandwidth capacity. It is also interesting to note, that not only does the delay decrease, the standard deviation of the delay also decreases with cherish/urgency multiplexer.

The effect of Markovian as against deterministic service was to reduce the overall delay, but at the cost of increased overall loss.

5.4 Remarks on End-to-End Quality Management

The work in [14] demonstrates that the use of the quality centric scheduling mechanism of the cherish/urgency multiplexing approach does appear to scale to larger networks. It does highlight the beneficial effects of the scheduling approach in delivering end-to-end QoS that would fulfill some of the strictest quality constraints required by today's applications, those of VoIP while concurrently servicing other critical applications — without having to sacrifice the efficient use of the communications resource. Although the cherish/urgency multiplexer does not explicitly manage throughput, through managing loss and delay the throughput takes care of itself. This is a direct consequence of the two degrees of freedom in finite queueing systems.

It should be noted that the results from the WFQ/DRR presented here are for an ideal implementation, in practice implementations of WFQ do not have the predictability ascribed to them here. The implementation of bandwidth control is often done with respect to fixed packet sizes and, under saturation the bandwidth allocations may not be respected[2]. However, implementations of the cherish/urgency multiplexer are predictable and operate consistently in saturation[10, 12].

6 Other Elements of the GoS[13] Architecture

As has already been mentioned, there are more elements to the GoS packet handling architecture. Figure 12 (based on Figure 4 in [16] and Figure 1 in [7]) illustrates the general way in which the architectural elements can be combined. The traffic is classified, policed ("P") and shaped ("S") before being multiplexed and transmitted onwards.

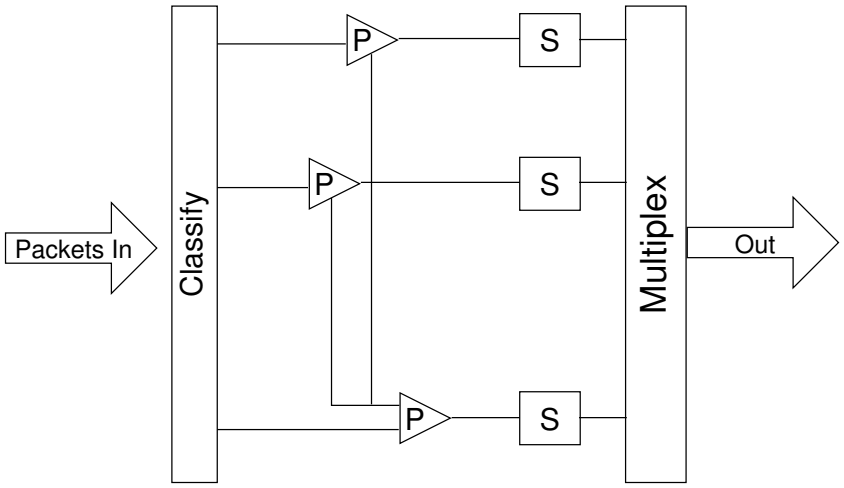


Figure 12: Outline GoS Architectural Elements

6.1 Policing and Shaping

After being classified[14], the "demand" of the arriving packet flow is distributed by the differential policers (see Section 6.2), before being shaped[4, 7] and presented to the cherish/urgency multiplexer. These actions assure that the pre-requisites for the correct operation of the cherish/urgency multiplexer are always met during operation. This ensures its fairness and system saturation behavior. The policer can "downgrade" traffic or discard it. This is a level of in-built protection against rogue packet flows, which might arise from deliberate actions or from mis-behaving applications (part of the isolation and saturation behavior management). Note that these operations also contribute to the packet flows' ΔQ. These effects, being under the control of the network element designer and network provisioner, can be calculated and "tuned" for the particular overall system requirements. As they only involve a single packet flow competing for a resource, the resultant ΔQ is much easier to assess.

By policing the packet flows the offered load placed on the cherish/urgency multiplexing element is bounded. This assures that the quality domination effects (as described in Section 3.3.3) cannot be abused, strengthening the delivered quality isolation. Both the policing and shaping components perform their operation with a controlled level of randomness[6]. This increases the fairness by as-

[13] GoS is a trademark of U4EA Technologies Ltd and is derived from the concept that this approach gives "Guarantee of Service".

[14] The classification requirements vary dramatically depending on the role of the network element. The classifier identifies packet flows, the classification may be as simple as the value of the TOS field, may be based on multiple fields in the packet header or some remembered state from previous packets in this flow.

suring that there cannot be any long-lived correspondence in phase between the packet arrival and any policing or shaping operation.

6.2 Instantaneous Load Dependant Quality Treatment

Even in the best planned and run networks, the demand of some flows will exceed their configured values at some time. This may be because of rate adaptive sources seeking the available network

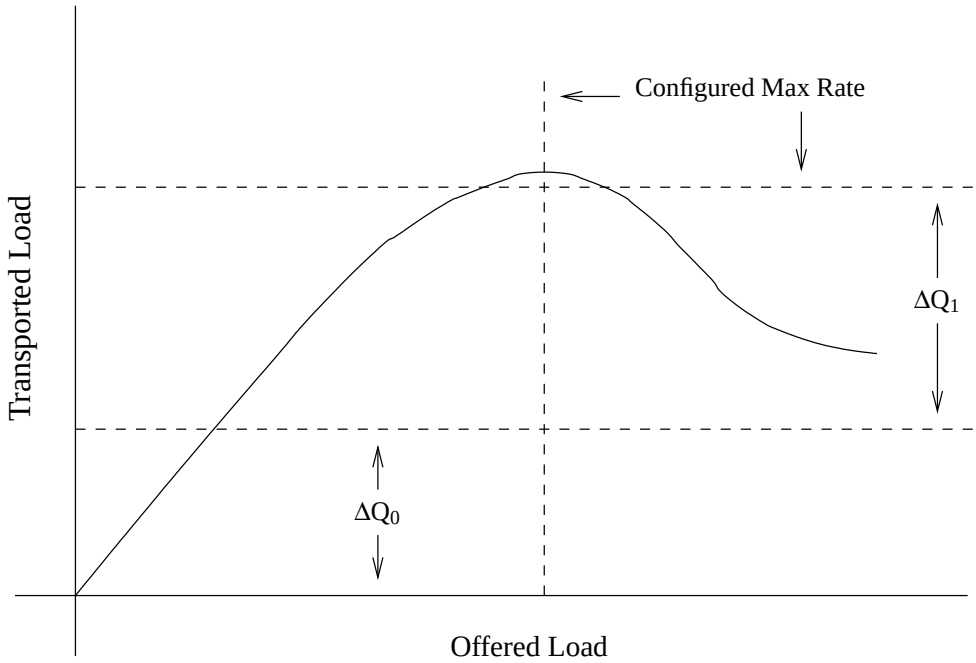


Figure 13: Variation of Transported Load and ΔQ with Offered Load

capacity because the expected statistical multiplexing gain did not occur (even for a short period of time), unexpected behavior of an application (e.g. aggressive retransmissions during an error condition), deliberate denial of service attack or just fall back operation due to equipment or transmission failure. When the demand exceeds the supply, what is the correct action that the network element should take? Unfortunately, there is not a "one size fits all" answer.

Elastic or other rate adaptive sources need feedback from the network (either directly or indirectly) to exercise the appropriate constraint over their demands[15, 8], this is inferred in two ways:

1. Packet loss (by receiver indication, usually implicit—multiple duplicate acknowledgements)
2. Round Trip Time (reception timing of acknowledgements)

Figure 13 (based on [4]) illustrates how, through the combined use of policing, shaping and multiplexing, both the experienced ΔQ and the transported load can be varied with the offered load. For example, this would permit feedback to an elastic application by increasing its experienced delay (thus reducing its effective throughput as bandwidth-delay product approaches the configured congestion window size) before discarding packets. For other applications it may be desirable to discard in preference to delay.

Tests[12] have found that this can interact well with mixtures of application traffic, even raising the general efficiency of use of communication links while still delivering predictable worst case quality, even when the system is in saturation[10].

6.3 Computational Complexity of Implementation

Ignoring the computation cost of classification[15], the GoS approach has a constant cost per packet, irrespective to the number of individual flows that are being managed. This compares well with WFQ where such computation costs increase typically as n or $\log(n)$ where n is the number of queues[13]. There is a complexity cost with the cherish/urgency multiplexer related to the number of urgency queues[16], which is constant for all practical considerations.

7 Conclusions

In this paper we have outlined the properties that we considered necessary to achieve predictability in the delivery of quality transport in saturated networks. We introduced ΔQ, a quantity that captures the inevitable quality degradation that data packets will suffer during their transportation through any network. This quantity can be used to capture the end-to-end degradation experienced by a packet-flow; the collective sum of degradation introduced by contention for a common resource; as well as the effects of transmission over a communication link. We have illustrated how ΔQ is "conserved". Demonstrating that any trading must occur within the the two degrees of freedom that inherently constrain any finite queueing system. We have analyzed three different queueing approaches, assessing them from their ability to deliver:

- differential quality treatment
- isolation
- fairness

as well as evaluating the efficiency and behavior of each queueing system as it becomes saturated. We have demonstrated that bandwidth management, while controlling the throughput delivered to packet flows, does not deliver assurances on delay and loss rates. The effective control of loss and delay is essential to successful operation of many applications.

We have presented an alternative multiplexing approach that is predictable. It can be configured to deliver a bounded ΔQ to a packet flow, even if the rest of the offered load to the system substantially exceeds the capacity of the outgoing link. We introduced the mathematical basis of the cherish/urgency multiplexer operation and outlined how, combined with other packet handling architectural elements, the cherish/urgency multiplexer can be engineered into a network element that can deliver all of the above-mentioned properties. We have drawn on other work[14], which has investigated the use of the cherish/urgency approach and its likely effectiveness in delivering end-to-end quality of service.

[15] The classification costs is highly dependant on the complexity of required classification task. This cost is independent of the QoS mechanism.

[16] The issue is the selection of the most urgent non-empty queue. The number of these queues is very limited, current products only use four such levels. In some processor architectures this selection process need only be one operation.

Unique amongst the systems studied the GoS paradigm delivers predictable quality management. Its mathematical underpinning permits not only the prediction of the likely ΔQ that a packet flow will experience during operation, but also permits the pre-configuration of GoS-based network elements to deliver a known ΔQ , including a worst case bound. As such we see the packet handling elements in the GoS architecture as the correct set of fundamental building blocks in constructing predictable network elements and hence a predictable end-to-end network.

References

[1] ALLEN, A. O. *Probability, Statistics, and Queueing Theory: with Computer Science Applications*, second ed. Academic Press Inc., 1990.

[2] BEURAN, R., IVONOVICI, M., AND DOBINSON, B. Evaluation of the delivery QoS characteristics of high performance switch/routers. In Preperation.

[3] DAVIES, N., HOLYER, J., AND THOMPSON, P. An operational model to control loss and delay of traffic at a network switch. In *Third IFIP Workshop on Traffic Management and Design of ATM networks* (Apr 1999).

[4] DAVIES, N. J., HOLYER, J. Y., LAFAVE, L. A., AND VOWDEN, C. J. Policing data based on data load profile. PCT Patent: WO 02/30063, Apr 2002.

[5] DAVIES, N. J., HOLYER, J. Y., THOMPSON, P. W., BRADLEY, J. T., AND FRANCIS-COBLEY, P. P. Routing device. PCT Patent: WO 00/65783, Nov 2000.

[6] DAVIES, N. J., THOMPSON, P. W., HOLYER, J. Y., LAFAVE, L. A., VOWDEN, C. J., AND WILLMOTT, G. Data flow control. PCT Patent: WO 02/30060, Apr 2002.

[7] DAVIES, N. J., THOMPSON, P. W., HOLYER, J. Y., LAFAVE, L. A., VOWDEN, C. J., AND WILLMOTT, G. Information flow control in a packet network based on variable conceptual packet lengths. PCT Patent: WO 02/30064, Apr 2002.

[8] HANDLEY, M., FLOYD, S., PADHYE, J., AND WIDMER, J. TCP Friendly Rate Control (TFRC): Protocol specification. RFC 3448, Jan 2003.

[9] HOLYER, J. A queueing theory model for real data networks. In *Sixteenth Annual UK Performance Engineering Workshop, University of Durham, UK* (Jul 2000).

[10] KEYLABS. Report on FlowFusion^{TM} QoS device. Available from `http://www.u4eatech.com/`, Aug 2002.

[11] KUZMANOVIC, A., AND KNIGHTLY, E. W. Low-rate tcp-targeted denial of service attacks (the shrew vs. the mice and elephants). In *ACM SIGCOMM* (Aug 2003).

[12] MIERCOM INC. Lab test report 070202 on FlowFusion^{TM} 2M. Available from `http://www.u4eatech.com/`, Feb 2002.

[13] RAMABHADRAN, S., AND PASQUALE, J. Stratified round robin: A low complexity packet scheduler with bandwidth fairness and bounded delay. In *ACM SIGCOMM* (Aug 2003).

[14] REEVE, D. C. *A New Blueprint for Network QoS*. PhD thesis, University of Kent at Cantebury, UK, 2003. In Preperation.

[15] STEVENS, W. R. TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. RFC 2001, Jan 1997.

[16] U4EA TECHNLOGIES. GoS technical description. Available from `http://www.u4eatech.com/`.