Performance Analysis of a VoIP Access Architecture

E. Noel
AT&T Laboratories
Middletown NJ
eric.noel@att.com

K. W. Tang
Dept. of Electrical Engineering
SUNY at Stony Brook NY
wtang@ece.sunysb.edu

Abstract

Using a simulation model for a benchmark VoIP access architecture, we investigate the performance issues associated with mixing real-time voice and congestion-sensitive data traffic. Arbitration of shared facility is accomplished via First Come First Serve (FCFS), Strictly Priority (SP), and Weighted Fair Queuing (WFQ) disciplines. The performance metrics used are facility utilization, transmission delay, queuing delay and packet loss. Engineering rules for sizing the network are provided. More specifically, our results indicate that proper engineering of the queue size under the SP discipline can prevent any packet loss of voice traffic; proper setting the weights of the WFQ scheduler can control the delay for voice traffic; and increasing the queuing delay of the WFQ scheduler can improve the packet loss rate of data traffic.

1. Introduction

Even though the internet community long-term view is that real-time voice and video services can be multiplexed with existing data traffic, Quality of Service (QoS) has not been considered with the same intensity as by the telecommunication community with real-time services on ATM [2]. Currently in the internet, the dominant standard for transmitting multimedia in packet-switched networks is the ITU Recommendation H.323, which does not provide any QoS guarantees [31]. The IETF has proposed several service models and mechanisms to meet the demand for QoS. Notably among them are the integrated services Resource Reservation Protocol (RSVP) [6], the Differentiated Services (DS) model [5], the Multi-protocol Label Switching (MPLS) protocol [30], traffic engineering and constraint-based routing [3, 9].

Mixing real-time traffic like voice (UDP) with bursty, congestion sensitive traffic (TCP) has potential for creating performance problems. Real-time traffic not only performs poorly because of delay variations and packet drops, but also hurts congestion-sensitive traffic when they compete for scarce bandwidth [32, 16]. Traffic self-similarity has been observed over a wide range of networking contexts [20, 10, 29, 34]. From a queueing theory viewpoint, long-range dependency salient point is that the queue length distribution decays much more slowly (i.e. polynomially) when compared to short range dependent traffic sources (i.e. Poisson with exponential decay). A number of performance studies have shown that self-similarity has detrimental effect on network performance, leading to increased packet loss rate, delay, and a degraded delay-throughput trade-off relation [1, 15].

We investigate self-similarity impacts on data network performance, and focus on performance issues associated with mixing voice and data traffic within the internet. To do so, we ran simulations with a modified version of *ns-2*, a network simulator widely adopted in the network research community [7]. Within the scope of our analysis, we consider scenarios where carrier grade telecommunication networks (no silence suppression) use the internet to transport portion of their voice traffic. And we investigate the impacts of congestion-sensitive traffic on real-time traffic. To account for the emergence of QoS standards, we allow for packet classification and bandwidth access arbitration. Typically, classification and arbitration is most useful at the network edges where access capacity is a sparse commodity. Whereas within the internet backbone, two opposite philosophies have been suggested: constrained routes with resource access arbitration [3, 9], and over engineered best effort with shortest path based routing [17].

This chapter is composed of the following sections: Section 2 where we list properties of self-similar processes and present the traffic source models. Section 3 where we describe the queueing discipline used in our simulation model. And Section 4 where we present our simulation model, engineer our test network, and provide simulation results.

Proceedings of the 2004 International Conference on Parallel Processing Workshops (ICPPW'04)
1530-2016/04 $20.00 © 2004 **IEEE**

Model	Phase	Mean	Comment
Web traffic generator [14]	1.2 1.2, 1.5 1.5 2	12 (kB) 4, 3 0.5 (sec) 50, 10 (sec)	"object size" "objects per page" "inter-object" "inter-page"
HTTP reply traces [21]	1.04-1.14	8-10 (kB)	
FTP traffic [33]	1.18	80 (kB)	Exponential session and burst inter-arrival time
ns-2 Pareto ON/OFF	1.5	64 (packets)	Set in *ns-default.tcl*
Heavy-tailed traffic [27]	1.05-1.95	4.1 (kB)	

Table 1. Published Pareto distribution parameters used in modeling internet data traffic.

2. Traffic models

2.1. Data Source

We model data sources as a packet train generators with idle time exponentially distributed, and a Pareto distributed train size (for its self-similar properties[8, 20, 34]).

Pareto distribution is a two parameter distribution (scale and phase) where the scale specifies the minimum value that the Pareto random variable can take, and the phase determines mean and variance. When the phase is less than 2, the distribution has infinite variance, and if the phase is less than 1, it has infinite mean and variance.

We chose 1.2 for the phase and 15kB for the mean train size based on the Pareto distribution parameters listed in Table 1. The key parameter being the phase, we made no attempts to distinguish between FTP and HTTP traffic. Trains are encapsulated in TCP, and inter-train departure time is the sum of the time needed for TCP to transmit the previous train plus the exponential idle time. We created a new *ns-2* class which allows for transmission of Pareto distributed packet trains over a TCP agent.

We use the Reno flavor of TCP to transport our data source packets, as it is one of the most popular implementations in the Internet today [28]. We used *ns-2.1b7* default values (window size of 20 packets and initial window size of 1 packet), and set packet size to 1kB.

Simulation studies for traffic sources with Pareto distributed packet trains of phase parameter ranging from 1.05 to 1.95 showed that the reliable transmission and flow control mechanisms of TCP maintain the long-range dependency structure induced by heavy-tailed packet train size distributions [27, 14]. Hence, we expect these findings to be equally applicable within our simulation environment.

In order to validate simulation runs, and engineer our simulated network, we derived expressions for U, the average facility utilization (of bandwidth C_{link}) consumed by a TCP session.

When TCP parameter RTT (Round Trip Time) is larger than the time required to insert a full window worth of packets, and no delayed ACK is used (one acknowledgment packet per successfully transmitted packet), the number of packets sent per cycle increases in powers of 2 until the window size W has been reached (assuming the file being transmitted is larger than the window size). So the number of packets sent per cycle progresses as follows:

Cycle	Packets sent
$c_{0<i<P}$	2^{i-1}
$c_{i\geq P}$	W

where $P = \lceil \log W / \log 2 \rceil$. When no packet retransmission are required (due to packet loss or other transmission impairment), our expression for the average facility utilization becomes:

$$U \approx \frac{|\text{file size}|}{n\text{RTT} + |\text{idle time}|} \times \frac{1}{C_{\text{link}}} \quad (1)$$

where n is the number of cycles needed for TCP to transmit the whole file (i.e. sum of packets sent starting with cycle c_1 to cycle c_n equals the number of packets needed to transmit the requested file), and C_{link} is the facility bandwidth. For models which account for packet loss, consult [24].

When RTT is smaller than the time required to transmit a full window worth of packets, and when no retransmission are required, our expression for the average utilization becomes (ignoring the initial cycles where RTT is larger than the time required to transmit maximum allowed packets):

$$U \approx \frac{|\text{file size}|}{|\text{file size}|/C_{\text{bottleneck}} + |\text{idle time}|} \times \frac{1}{C_{\text{link}}}$$

where $C_{\text{bottleneck}}$ is the bandwidth of the slowest facility in the path.

2.2. Voice Source

In circuit switch networks, voice call arrivals are Poisson distributed and their holding time is exponentially distributed [12]. There, traffic engineering of circuit switch

networks relies on Erlang-B distribution [13]. Therefore we model voice sources by an $M/M/N/N$ queueing system (Poisson arrivals, exponential service time, no waiting room) where N represents the number of voice channels (for the purpose of our study, assumed to be 64kbps or DS0). And we use the following parameters: λ the arrival rate, N the number of voice channels, and μ the mean holding time (fixed to 3 minutes).

ITU standard	Data rate
G.711	64kbps
G.721	32kbps
G.728	16kbps
G.729E	11.8kbps
G.729A	8kbps

Table 2. Most popular ITU speech compression standards together with their data rate.

The media gateway is responsible for conditioning the audio stream into IP packets prior traversing the internet, and reconstructing the stream upon exiting the internet. On the transmit side, the media gateway compresses the audio stream (64kbps PCM to one of the compressed audio standards listed in Table 2, with or without silence suppression) into fixed duration compressed speech frames (typical lengths are 10-50msec), appends to the frames an UDP header, and transmits them over internet facilities. On the receiver side, the media gateway reconstructs the compressed audio stream. This activity includes delay jitter removal via a playout buffer [23, 22], decoding compressed audio, possibly applying a packet loss concealment algorithm, and removing echo.

We model the media gateway as an element which sends bursts of audio packets at periodic intervals (period equals to that of the speech frame size). To achieve carrier grade voice quality, silence suppression is disabled. So within each burst, the number of transmitted packets equals the number of active calls. The media gateway part of our voice source uses the following parameters: speech frame size, IP payload size, and IP header size (fixed to 40B).

We created a new *ns-2* class which allows for transmission of $M/M/N/N$ distributed call arrivals processed by a media gateway over a UDP agent.

As we did for the data source, to validate simulation runs and engineer our simulation network, we derived an expression for U, the average facility utilization (of bandwidth C_{link}) consumed by our traffic source:

$$U \approx \lambda\mu(1-p_b) \times \frac{\text{packet size}}{\text{packetization delay}} \times \frac{1}{C_{\text{link}}}, \qquad (2)$$

where p_b is the blocking probability given by the Erlang-B formula. Note that when $N = 1$, our traffic source reduces to an ON/OFF traffic source with exponential ON and OFF periods.

3. Queueing Disciplines

3.1. First Come First Serve (FCFS) and Strict Priority (SP)

With First Come First Serve (FCFS), incoming packets are served in their order of arrival [18]. Such queueing discipline requires no distinction among packets. Hence no performance guarantees (delay, loss, bandwidth) can be established. Internet networks that use FCFS are often referred as best effort networks.

When the incoming packets are distinguishable according to groups, then priority queueing disciplines are possible. With Strict Priority (SP), incoming packets within the strict priority group will always be served before packets within the low priority group. Hence the longest time a strict priority packet would be queued equals to the maximum time needed for serving a packet within the low priority group, plus the time needed for serving all packets within the priority group that may have queued while waiting for the low priority packet to clear. Strict priority traffic must be properly engineered in order to avoid starving low priority traffic.

3.2. Weighted Fair Queueing (WFQ)

The problem of fair network resource allocation has led to the development of a class of algorithms that provide tight end-to-end delay bounds and efficient resource utilization. These algorithms try to approximate the ideal behavior of the Generalized Processor Sharing (GPS) algorithm [11].

The GPS algorithm assumes that input traffic is infinitely divisible and all sessions can be served simultaneously. Each connection (i) can be associated with a service weight (w_i), so that it receives service in proportion to the weight whenever there is data in its queue. A GPS server guarantees each session receive a service rate of at least: $C \times w_i / \sum_j w_j$, where C is the connection rate and the sum is over the weights of all active connections.

Because a GPS server is unimplementable, approximations to GPS servers are used. Weighted Fair Queueing (WFQ) is such an approximation. WFQ does not make GPS's infinitesimal packet size assumption [25, 26], and serves packets in order of the time a packet would complete service had it been served with a GPS server.

Even-though the delay of WFQ buffers subject to leaky bucket traffic shaped source has been shown to be bounded [35], in a network with hierarchical link sharing service the inaccuracy introduced by WFQ in approximating GPS

is detrimental to both real-time and best effort traffic [4]. Moreover, in [4], an illustration is provided to show how the maximum delay through a WFQ buffer increases with the number of sessions.

4. Simulation Model

In order to investigate the impacts of congestion-sensitive traffic on real-time traffic, we consider scenarios where carrier grade telecommunication networks use the internet to transport portion of their voice traffic.

Using *ns-2*, we constructed Figure 1 reference network. It consists of media gateways (GW) fed by $24n$ voice channels, and of data routers (R) fed by N_{ds} data sources. Both the gateway and data traffic is forwarded to router A.

With our hardware configuration (dual 400Mhz CPU 2Gb RAM Sun Enterprise 250 and 450Mhz CPU 250Mb RAM Sun Blade 100), *ns-2* runtime becomes prohibitive beyond T3 rate (≈44Mbps), and simulated networks with more than 1,000 TCP sessions.

To account for the effects of QoS standards, packets are classified into two classes: real-time (voice over UDP) and congestion-sensitive (data over TCP). Router A arbitrates bandwidth access according to Section 3 algorithms: FCFS, WFQ, and PS.

	Voice	Data
Duration or size	$\mu_{\text{voice}} = 3$min	$\mu_{\text{train size}} = 15$kB $\alpha = 1.2$
Volume	Offered load = 80% × # voice channels	# data router = 17 Idle time = 30msec
Packet size	120B (G.711 10msec frame size)	1kB

Table 3. Voice and data sources parameters used for simulation.

In Table 3, we list all the parameters for voice and data sources used in our simulation.

All facilities are 0 delay but the facilities between router A and B on access and B and A on egress, both are 10msec long. As an estimate for RTT, we ignored queueing delays, used the emission delays for 1kB packet and 40B ACK, plus the 4×10msec transmission delay. This resulted in a RTT of ≈ 42msec.

As mentioned previously, the mean train size is 15 data packets. Under the assumption that RTT is larger than the time needed to transmit a full window of packets (insertion time of 20 data packets over a 10Mbps facility is 16msec), the number of cycles (n) needed to send the whole train is 4 ($= \lceil \log_2(15) \rceil$).

Based on our estimate for RTT, our traffic sources parameter values (Table 3), together with the expressions for their corresponding average facility utilization (Sections 2.1 and 2.2) we engineered the number of traffic sources to drive our reference network (Figure 1). In Table 4 we summarize the various configurations. The estimates for the bottleneck facility utilization (facility between router A and B in Figure 1) ignore impacts on data sources from queueing and packet loss.

For each combination of T1 and data traffic quantities listed in Table 4, we ran simulation within the *ns-2* environment for FCFS, WFQ, and SP cases. For WFQ, we tried the following weights for voice: 0.2, 0.4, 0.6, 0.8 (sum of voice and data weights must equal 1).

Each simulation ran for a simulated time of 2.8 hours. Statistics associated with the first 1,000sec were disregarded. For each simulation run we created 10 independent replications from which we extracted a 95% confidence interval (interval around the sample mean that captures 95% of the samples) assuming a t-distributed normalized error [19]. That number turned out to be approximately $\pm$ 5% of the reported values for means and $\pm$ 10% of the reported values for maximums.

However, we concede that, because our data sources have infinite variance, assuming the normalized error to be t-distributed when constructing confidence intervals is questionable.

5. Performance Metrics

Our simulation model reports the following measurements:

- Facility utilization: Ratio of the bits carried during the measurement interval to the product of the facility bandwidth and the measurement period. This is done in the link delay object.
- Transmission delay: Packet emission time is stored in the common packet header, and compared to the arrival time in the link delay object. We capture the mean, maximum, and histogram (1msec sampling period) of transmission delay over the measurement period.
- Queueing delay: Packet arrival time to a queue is also stored in the common packet header, and compared to the departure time from the queue in the queue monitor object. We capture the mean, maximum, and histogram (sampled over 1msec periods) of the queueing delay over the measurement period.
- Packet loss (due to queue overflow): Number of packets dropped due to queue overflow over the transmission period. This is measured in the queue monitor object.

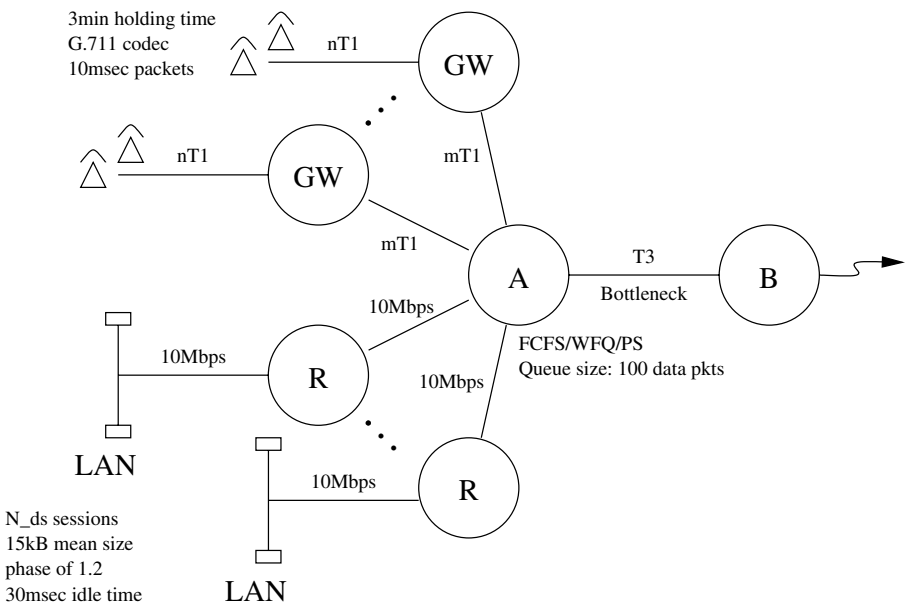

Figure 1. Access reference network. (Egress side is a mirror image of the access connection.)

- Delay jitter: Within our network architecture, delay jitter is entirely due to queueing.

6. Results

When possible, we will highlights issues and tradeoffs associated with designing IP networks that combines voice and data traffic. Our metrics of interests are: average facility utilization, queueing delay, and packet loss.

6.1. Utilization

In Figure 2, we compare average facility utilization at the bottleneck facility measured from simulation against the one derived from our expression for data and voice traffic facility utilization (Equations 1 and 2), as a function of number of data sources, for varying number of voice T1s and under the single or multiple gateway scenarios.

We find that our expressions for mean facility utilization provide a good upper bound, and account for the measured utilization up to a level of $\approx 80\%$.

6.2. Delay

Under the SP queueing discipline, and within the scope of our reference network, the maximum delay a voice packet would incur is $mt_{\text{voice}} + t_{\text{data}}$ where t_{voice} and t_{data} are the voice and data packets emission delays (length of packet over facility throughput), and m is the maximum number of voice packets queued during the emission of a data packet.

	Single Gateway	Multiple Gateway
17 voice T1s 3 data routers	0.362msec $(0.333 \pm 2e^{-3})$	0.550msec $(0.514 \pm 1e^{-3})$
14 voice T1s 4 data routers	0.362msec $(0.181 \pm 5e^{-3})$	0.485msec $(0.414 \pm 3e^{-3})$
9 voice T1s 6 data routers	0.362msec $(0.181 \pm 3e^{-3})$	0.377msec $(0.220 \pm 2e^{-3})$
5 voice T1s 8 data routers	0.362msec $(0.181 \pm 7e^{-3})$	0.290msec $(0.202 \pm 4e^{-3})$

Table 5. Comparison between measured (parenthesis) and estimated maximum queueing delays for voice packets under the SP queueing discipline. (Each voice T1 corresponds to a M/M/24/24 queueing system, and each data router corresponds to 17 data sources.)

Based on our simulation results, the following expression for m yields a good approximation:

$$m = \begin{cases} t_{\text{data}}/t_{\text{voice}} & \text{Single gateway case,} \\ n & \text{Multiple gateway case.} \end{cases}$$

In Table 5, we compare the measured maximum queueing delay values to our empirical upper bound estimates. With decreasing number of voice sources we found agreement between simulation and upper bound estimate to degrade, the rarity of such events should account for the dif-

		Single gateway		Multiple gateways	
Voice T1s	Data routers	Erlang-B	U_{bn} (%)	Erlang-B	U_{bn} (%)
0	1	NA	9.8	NA	9.8
	2		19.7		19.7
	3		29.5		29.5
	4		39.4		39.4
	5		49.2		49.2
	6		59.1		59.1
	7		68.9		68.9
	8		78.8		78.8
5	0	$2.3e^{-3}$	20.8	$5.3e^{-2}$	19.8
	4		60.2		59.2
	6		79.9		78.9
	8		99.0		98.6
9	0	$1.8e^{-4}$	37.5	$5.3e^{-2}$	35.5
	2		57.2		55.2
	4		76.9		74.9
	6		96.6		94.6
14	0	$9.1e^{-6}$	58.4	$5.3e^{-2}$	55.3
	1		68.2		65.1
	3		87.9		84.8
	4		97.8		94.7
17	0	$1.6e^{-6}$	71.0	$5.3e^{-2}$	67.2
	1		80.8		77.0
	2		90.6		86.9
	3		100.4		96.7

Table 4. Voice ($\rho = 0.8$, $\mu = 3$min, 10msec frame size, 160B packets) and data (17 data sources per router, $k = 2.5$, $\alpha = 1.2$, RTT=42msec, $n = 4$, 1kB packets) sources configuration for both the single and multiple gateway cases together with voice call blocking probability (Erlang-B), and estimates for the maximum average facility utilization (U_{bn}) in the bottleneck facility. (NA: Not applicable.)

	FCFS	SP	WFQ $w = 0.2$	WFQ $w = 0.8$
Packet Loss Probability	$6e^{-3}$	$8e^{-3}$	0	0
Queueing Delay	49msec	49msec	87msec	185msec

Table 7. Maximum packet loss probability and queueing delay incurred by TCP traffic across all simulation runs.

ference.

As mentioned, WFQ queueing discipline attempts to provide each session a fraction of facility bandwidth proportional to its WFQ weight. In Figure 3, we found that selection of the WFQ weights has direct impacts on queueing delays.

With FCFS, as expected, queueing delay increases with increasing bottleneck facility utilization. At at fixed facility utilization, queueing delay increases with increasing number of data sources, which can be accounted for by the increasing variability in traffic (see Table 6). For same number of voice T1s and routers, we found negligible differences in queueing delay between the single and multiple gateway scenarios.

With data traffic, the queue discipline impacts whether TCP is throttled down by the self-clocking property or the slow-start algorithm. In Table 7, we list maximum values for packet loss probability and queueing delay over all our simulation runs. Based on these values, we deduce that TCP will be throttled down by the self-clocking property under the WFQ queueing discipline, and by the slow-start algorithm under FCFS and SP queueing disciplines.

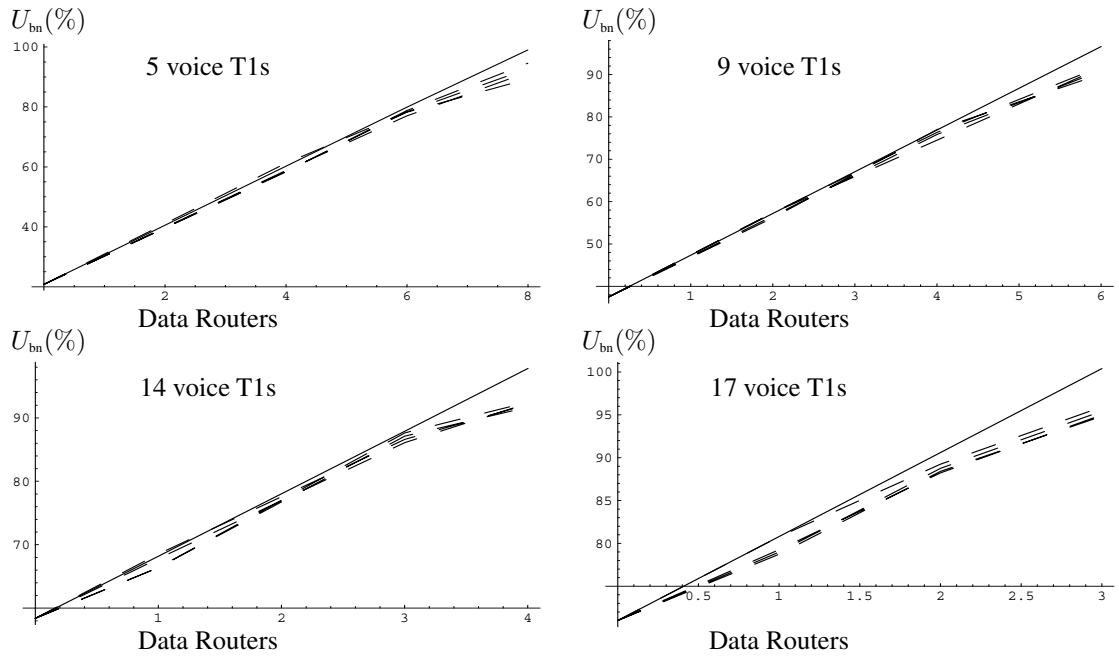

Figure 2. Comparison between simulation (dashed lines) and model (continuous line) for the average Bottleneck facility utilization under the single gateway scenario. (Dashed lines starting from top most correspond to: FCFS, WFQ with weight 0.2, WFQ with weight 0.8, and SP.)

6.3. Packet Loss

Packet loss of voice traffic under the SP queueing discipline can be controlled by setting its maximum queue size to the maximum number of voice packets which can arrive during the emission time of a data packet. However, we cannot provide any such conditions for the other two queueing disciplines. Whereas, data traffic packet loss can be avoided at the cost of increasing queuing delay using the WFQ queueing discipline.

7. Conclusion

To investigate performance issues associated with mixing voice and self-similar data traffic within the internet, we used the *ns-2* simulation tool. We modified *ns-2* data collection procedures so as to improve run time. And we constructed objects to simulate our voice (M/M/N/N queueing system) and self-similar data sources. Our reference scenario consisted of several voice and data traffic sources contending for a shared facility of limited bandwidth. Contention resolution consisted of the following queueing disciplines: FCFS, WFQ, and SP.

With our hardware platform (dual 400Mhz CPU 2Gb RAM Sun Enterprise 250 and 450Mhz CPU 250Mb RAM Sun Blade 100) we found run time to be prohibitive beyond T3 (44.184Mbps) for the simulation scenarios we investigated.

We derived simple expressions for average facility utilization which we used to engineer our reference network. When compared to simulation, we found that it accounted for most simulation results up to $\approx 80\%$ utilization. We confirmed that under SP queueing discipline, the maximum delay a voice packet would incur is the emission delay of a data packet plus the emission delay of all voice packets which arrived during the data packet emission delay. We observed that proper setting of WFQ weights allowed us to control delay for voice traffic. With FCFS, we found that for fixed average bottleneck facility utilization, queueing delay would increase with increasing number of self-similar data sources. From our simulations, we deduced that data sources transported by TCP are throttled down by the self-clocking property under the WFQ queueing discipline, and by the slow-start algorithm under the FCFS and SP queueing disciplines.

For packet loss of voice traffic, we noted that proper engineering of the queue size under the SP queueing discipline should prevent any packet loss. Whereas packet loss of data traffic could be avoided at the cost of increasing queuing delay using the WFQ queueing discipline.

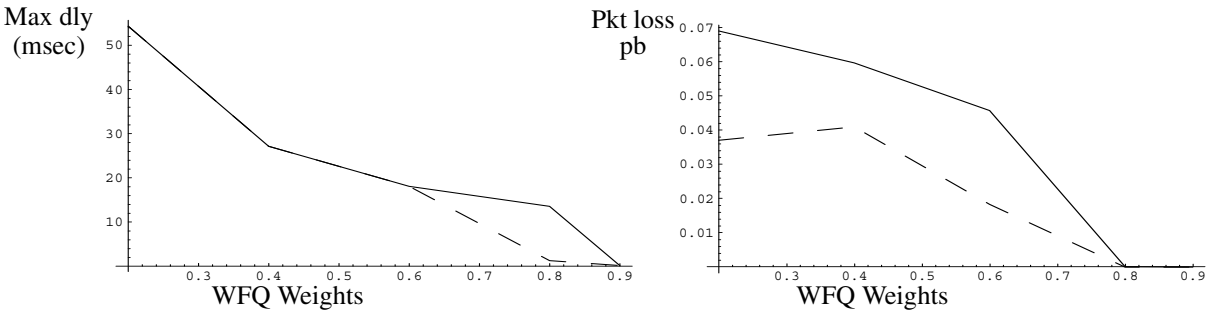

Figure 3. Maximum voice packet queueing delay and packet loss probability as function of WFQ weights for the case 17 voice sources and 3 routers. (Continuous line: single gateway, Dashed line: multiple gateway).

5 voice T1s	10.541msec @58.3% 39.230msec @78.8% 49.029msec @94.5% 75.778msec @99.9%	14 voice T1s	0.385msec @66.4% 21.478msec @87.1% 22.959msec @91.7% 43.753msec @99.8%
9 voice T1s	0.362msec @57.3% 26.533msec @74.5% 32.948msec @91.6% 52.407msec @99.7%	17 voice T1s	13.377msec @78.7% 16.725msec @88.4% 19.617msec @94.8% 30.827msec @99.8%

Table 6. Maximum FCFS queueing delay function of average bottleneck facility utilization for varying number of voice T1s.

8. Acknowledgment

The authors would like to thank the anonymous referees for providing insightful comments.

References

[1] R. Agrawal and F. Baccelli. "Dominating tails in a tandem of queues with long range dependent arrival and service processes" In *IEEE International symposium on Circuits and systems*, Geneva Switzerland, pages 369–371, May 2000.

[2] ATM forum. "Traffic Management Specification Version 4.0". af-tm-0056-00, April 1996.

[3] D. Awduche. "Requirements for Traffic Engineering over MPLS". Internet draft, draft-ietf-mpls-traffic-eng.00.txt, October 1998.

[4] J. C. R. Bennett and H Zhang. "WF2Q: Worst-case Fair Weighted Fair Queueing". In *Proceedings of INFOCOM '96*, San Francisco, CA, March 1996.

[5] S. Blake. "An Architecture for Differentiated Services". Internet RFC 2475, December 1998.

[6] R. Braden. "Resource ReServation Protocol (RSVP) Version 1". Internet RFC 2205, September 1997.

[7] L. Breslau and D. Estrin and K. Fall and S. Floyd and J. Heidemann and P. Hunag and S. McCanne and K. Varadhan and Y. Xu and H. Yu. "Advances in Network Simulation". *IEEE Computer*, pages 59–67, May 2000.

[8] D. R. Cox. "Long-range dependence: A review". *Statistics: An appraisal*, pages 55–74, The Iowa S tate University Press, Ames, IA, USA, 1984.

[9] E. Crawley. "A Framework for QoS-based Routing in the Internet". Internet RFC 2386, August 1998.

[10] M. Crovella and A. Bestravos. "Self-similarity in world wide web traffic: Evidence and possible causes". In *Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*.

[11] A. Demers, S. Keshav, and S. Shenker. "Design and Analysis of a Fair Queuing Algorithm". In *Proceedings of ACM SIGCOMM'89*, Austin, September 1989.

[12] A. K. Erlang. "The Theory of Probabilities and Telephone Conversations". Nyt Tidsskrift for Matematik B, vol 20, 1909.

[13] A. K. Erlang. "Solution of some Problems in the Theory of Probabilities of Significance in Automatic Telephone Exchanges". Elektrotkeknikeren, vol 13, 1917.

[14] A. Feldmann and P. Huang and A. C. Gilbert and W. Willinger. "Dynamics of IP traffic: A study of the role of variability and the impact of control". ACM/SIGCOMM 1999.

[15] M. Grossglauser and J. C. Bolot. "On the relevance of long-range dependence in network traffic". In *Proceedings of the ACM SIGCOMM'96*, pages 15–24.

[16] D. P. Hong and C. Albuquerque and C. Oliveira and T. Suda. "Evaluating the Impact of Emerging Streaming Media Applications on TCP/IP Performance". *IEEE communications Magazine*, pages 76–82, April 2001.

[17] D. S. Isenberg. "The Rise of the Stupid Network". *Computer Telephony*, pages 16–26, August 1997.

[18] L. Kleinrock. *"Queueing Systems Volume 1: Theory"*. Wiley Interscience, 1975.

[19] S. S. Lavenberg. *"Computer Performance Modeling Handbook"*. Academic Press, 1983.

[20] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. "On the Self-Similar Nature of Ethernet Traffic (Extended Version)". *IEEE/ACM Transaction on Networking*, 2(1):1–15, 1994.

[21] B. A. Mah. "An Empirical Model for HTTP Network Traffic". *IEEE* 1997.

[22] W. A. Montgomery. "Techniques for Packet Voice Synchronization" *IEEE Journal on Selected Areas in Communications*, vol sac-1, no 6, pages 1022–1028, December 1983.

[23] S. B. Moon and J. Kurose and D. Towsley. "Packet audio playout delay adjustment: performance bounds and algorithms". *Multimedia Systems*, 6:17–28, 1998.

[24] J. Padhye and V. Firoiu and D. F. Towsley. "Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation". *IEEE/ACM Transactions on Networking* vol. 8, no. 2, pages 133–145, April 2000.

[25] A. K. Parekh, R. G. Gallagher. "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case". *IEEE/ACM Transactions on Networking*, Vol. 1, No. 3, pages 344-357, June 1993.

[26] A. K. Parekh, R. G. Gallagher. "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple-Node Case". *IEEE/ACM Transactions on Networking*, Vol. 2, No. 2, pages 137-150, April 1994.

[27] K. Park and G. Kim and M. Crovella. "On the relationship between file sizes, transport protocols, and self-similar network traffic". In *Proceedings of the International Conference on Network Protocols*, pages 171–180, October 1996.

[28] V. Paxson. "Automated packet trace analysis of TCP implementations". In *Proceedings SIGCOMM'97*, 1997.

[29] V. Paxson and S. Floyd. "Wide-area traffic: the failure of Poisson modeling". In *Proceedings ACM SIGCOMM'94*, pages257–268.

[30] E. Rosen and A. Viswanathan and R. Callon. "Multiprotocol Label Switching Architecture". Internet draft, draft-ietf-mpls-arch-01.txt, March 1998.

[31] H. Schulzrinne and S. Casner and R. Frederick and V. Jacobson. "RTP: a transport protocol for real-time applications". Internet RFC 1889, January 1996.

[32] S. Shenker. "Fundamental Design Issues for the Future Internet". *IEEE Journal on selected areas in communications*, vol. 13, no 7, pages 1176–1188, September 1995.

[33] K. Thompson and G. Miller and R. Wilde. "Wide-area Internet traffic patterns and characteristics". *IEEE Network*, 11(6):10–23, November 1997.

[34] W. Willinger, V. Paxson, and M. S. Taqqu. "Self-Similarity And Heavy Tails: Structural Modeling of Network Traffic at the Source Level". In *A Practical Guide To Heavy Tails: Statistical Techniques and Applications*, pages 27–53, R. Adler, R. Feldman, and M.S. Taqqu, editors, Birkhauser, Boston, 1998.

[35] H. Zhang. "Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks". *Proceedings of the IEEE*, 83(10):1374-1399, October 1995.