Emergent Properties of Queuing Mechanisms in an IP Ad-Hoc Network

17 Nov 2005 -- Document Version 1.0.1

By

Minh Lee, PR Rai, Fred Yip, Tim Eagle, Steve McCormick and Bob Francis – Cisco Systems

&

Dr. Dave Reeve, Dr. Neil Davies and Fred Hammond -- Predictable Network Solutions

A research investigation into the emergent properties of queuing mechanisms (at a single instance) in saturated IP ad-hoc networks sponsored under a Joint Research Agreement between The Boeing Company and Cisco Systems with technical assistance provided by Predictable Network Solutions.

Acknowledgements

The following companies and individuals made this study possible:

Cisco Systems, Inc.:

Cisco Systems, Inc. is the worldwide leader in networking for the Internet. Cisco engineers have been leaders in the development of Internet Protocol (IP)-based networking technologies. This tradition of IP innovation continues with industry-leading products in the core areas of routing and switching, as well as advanced technologies in areas such as: Home Networking, IP Telephony, Optical, Network Security, Storage Networking and Wireless LAN.

Contributors from Cisco to this report were: Minh Lee, PR Rai, Fred Yip, Tim Eagle, Steve McCormick and Bob Francis. Further information on Cisco Systems, Inc. can be found at www.cisco.com. Questions and comments can be emailed to Bob Francis at bfrancis@cisco.com.

Predictable Network Solutions Inc:

Research consultants that have created a quantitative mathematical understanding of end-to-end network quality of service (QoS). Predictable Network Solutions uses that understanding to educate and advise developers, builders and managers in the construction of networks and distributed systems so that predictable application outcomes can be delivered to end-users.

Contributors from Predictable Network Solutions to this report were: Dr. Dave Reeve, Dr. Neil Davies and Fred Hammond. Further information on Predictable Network Solutions can be found at www.pnsol.com. Questions and comments can be emailed to Fred Hammond at fred.hammond@pnsol.com.

Boeing – Phantom Works:

Phantom Works is the R&D unit of The Boeing Company. Phantom Works is working with the U.S. Coast Guard on its Integrated Deepwater program and with the U.S. Army on its Future Combat System program. Both require the development of an integrated set of systems - communication satellites, aircraft, sea-or ground-based systems and more - that will maximize mission capabilities while minimizing overall operations and support costs.

Contributors from Phantom Works to this report include Dr. Jae Kim. Further information on Phantom Works can be found at www.boeing.com/phantom. Questions and comments can be emailed to Jae Kim at jae.h.kim@boeing.com.

Revision History

Revision 1.0 – October 17, 2005

Orignally published report.

Revision 1.0.1 – November 17, 2005

Page 19 – fixed missing figure reference ("**Error! Reference source not found"**), which now reads as:

> Figure 8: Sample QoS Summary Graph for Strict Priority Queuing at 1Mbps illustrates the composite results for all the test runs using Strict Priority Queuing with a link speed of one megabit per second.

Page 20 – fixed missing figure reference ("**Error! Reference source not found"**) and heading reference (**"0"**), which now reads as:

> The "Consumed DQ Scatter Diagrams" is representing, for a given queuing discipline at a given speed, the amount of end-to-end budget (in terms of both loss and delay) that an application has consumed. The vertical red bar represents the end-to-end budget. So being to the left of that mark is goodness. Figure 8 is an example of a "Consumed DQ Scatter Diagram" chart. The complete set of graphs can be found in Appendix A.

Table of Contents

List of Figures

1 Overview

Cisco Systems, Inc. in conjunction with Boeing's Phantom Works and Predictable Network Solutions have conducted an industry leading investigation into the emergent properties of queuing mechanisms with regards to IP Quality of Service (QoS) in Ad-Hoc wireless networks. This study was commissioned specifically to provide empirical data on how these queuing mechanisms deliver and maintain QoS across multiple IP traffic flows within a single network node. This data was observed at fixed link speeds as well as their transitions with the intent to understand how physical link speed changes affect the ability of queuing mechanisms to maintain QoS during those transitions.

To meet the goal of the study a synthetic traffic model to simulate a set of real world applications and their traffic loads was derived through industry research. Multiple fixed (steady state) link speeds and their transitions were identified that would best simulate the current state of Ad-Hoc wireless networking environments. To generate a comprehensive set of observations for the study, a test bed was constructed that consisted of network equipment (routers and switches), workstations, applications to simulate/generate IP traffic, and a set of software tools were used to capture selected data and report the observed results.

The goal of the study was to:

- Develop an understanding of the interactions of a defined traffic mix on a single network node in the context of a set of desired application outcomes;
- Study and document the impact that selected industry queuing mechanisms have on achieving those outcomes;
- Investigate the impact that selected industry best practices have on improving those outcomes; and
- Provide a foundation for future research in the area of IP QoS for Ad-Hoc wireless networks.

2 Problem Statement

Converging to a packet-based network, with variable length packets, requires some effort to get the predictability necessary to support all service types and applications. In Ad-Hoc networks WAN bandwidth is expensive and potentially unpredictable, especially in the area of bandwidth availability and route stability.

A high bandwidth circuit can become a low bandwidth circuit and it must continue to be suitably shared among diverse applications with diverse needs -- from videoconferencing to Voice-over-IP (VoIP) to database lookups. For jitter-sensitive traffic, like voice or video, adding more bandwidth alone will not guarantee the desired results as it is not possible to tell when another packet will get in the way. Other traffic can be very sensitive to delay, but not as sensitive to packet loss (e.g., voice). Traffic can also be very bandwidth hungry, not overly delay sensitive, but still sensitive to packet loss (e.g., bulk-TCP traffic).

QoS endeavors to ensure that selected applications can coexist and function at acceptable levels of performance across a shared network infrastructure. The network QoS function is a set of technologies that work together to provide appropriate end-to-end treatment of various traffic types. The scope of this paper is to focus on the emergent effects of queuing mechanisms on a collection of application traffic flows as they traverse a single network element. Understanding how this building block of the QoS-puzzle reacts to the changing dynamics of an Ad-Hoc network is key to determining application predictability, as well as being able to engineer a reliable network.

Wireless Ad-Hoc networks present new challenges in delivering predictable QoS. When working in the wireless environment there can be rapid changes in the available capacity, as a link's speed can change as the Signal to Noise Ratio (SNR) varies. This, combined with a changing demand for service, can result in rapid, and possibly unpredictable, changes in the delivered QoS.

Wireless links present a variable amount of capacity to the queues that feed them. This changing capacity is a new variable not found in wired networks, and its impact needs careful consideration from a QoS perspective. As the "service capacity" changes it will affect both the loss and delay that packets experience as they traverse a network element. Assuming that the offered load remains constant, decreasing the service capacity will cause both increased delay -- as it now takes longer for a packet to reach the front of the queue and increased loss -- as the queue is more likely to become full. This variability in service capacity presents a new challenge for today's queuing mechanisms.

The applications used in this study were identified as having properties similar to those that might be used in a safety critical/tactical mobile environment. Such an environment presents new challenges for QOS in that an application's safety critical requirements must be delivered even as the network degrades. To highlight this point the study identified eight different types of traffic along with their order of failure and studied their emergent behavior under a combination of queuing mechanisms and link speeds.

3 Conceptual Overview

Networks have to operate within their underlying constraints – "Network Physics". In the wireless environment the effects of these constraints are more apparent. Obtaining a better grasp of the fundamental constraints on network element performance will enable a new range of solutions to be developed.

3.1 Quality Degradation

3.1.1 Quality Degradation over an End-to-End Path

From the point of view of an application or any other network-service user, a network is just a black box; accepting traffic at one edge, and perhaps, transporting it to another edge. From the point of view of an external observer, packets just enter and (perhaps) some time later leave the network "cloud". Within that cloud there are only two things that can happen to a packet. It can:

Experience Delay There must always be some delay (at a minimum due to propagation), and, more often than not, delay due to queuing. Once delay has been experienced it cannot be "undone".

Experience Loss As a packet crosses the network its chances of corruption increase, as does the chance of arriving at a full queue.[1]

It is clear that rather than providing a packet with "quality", a network's affect is to introduce *Quality Degradation*; that is an increase in a packet's observed loss and delay.

3.1.2 Quality Degradation at a Node

Consider for a moment a single queue with a given number of buffers and a fixed service rate. Every packet in the arriving traffic takes part in two "competitions". The first is to be admitted to the queue (failure to do so is loss). The second is to get to the front of the queue and be serviced (failure to do so is delay). It is these two competitions that are responsible for introducing Quality Degradation. As we increase the load on the queue (by increasing the rate of arriving traffic or decreasing the outbound link rate) the chances of the queue being full increase, as does the length of the queue. The net effect is an increase in the total Quality Degradation (across all flows).

The quantity of Quality Degradation, for a given queue/network element under a given load, is both immutable and conserved. Therefore, all that can be done with that quantity is to differentially share it out amongst the set of competing flows.

3.2 Trading Quality Degradation

In order to provide network QoS we must have the ability to give differing amounts of loss and/or delay to a set of flows at the points where their packets queue for service. The ability to do so, and the extent to which it can be done, is determined by the *emergent properties* of the *queuing mechanism*.

As we have already stated, **the Quality Degradation introduced by a single queue is an immutable, conserved quantity, irrespective of the queuing mechanism being used.** This means that all a mechanism can do is to differentially share the degradation out, in essence engaging in a trading game. Given this, it follows that to provide a lower degradation to one flow means that (by conservation) another flow must receive a higher degradation. Providing "better quality" to one flow implies that other flow(s) will have to receive a "lower quality".

Trading Quality Degradation is a constrained problem, one that has precisely two degrees of freedom, in offered load, loss and delay. This is analogous to Boyle's Law governing the

---

[1] We are assuming that if a packet is lost then it can never be recovered within the network. Retransmitting the packet, from the edge, is considered to be a new packet.

relationship between pressure, temperature and volume (although that is a linear relationship, in networks this is not the case). This restriction is inherent in all finite queues, irrespective of the queuing mechanism.

To illustrate the point, listed below are some examples of trades:

- *Trade a lower load (the ratio of the arrival rate to service rate) for a lower Quality Degradation*. This works by ensuring that the queue is lightly loaded and therefore unlikely to introduce loss. The result is a system that is under-utilized and has less quality degradation to be distributed.
- Where the load is fixed we can:
  - *Decrease the delay by introducing loss*. By discarding packets the queue length can be decreased, and as a result the delay is also decreased.
  - *Decrease the loss by introducing delay*. By increasing the number of buffers the loss is reduced, but the delay is now increased as there is more queuing.

Although not always directly apparent, all approaches to QoS are governed by the same rules. Interestingly most commercial wired networks opt for the first trade; namely keeping the load below some selected threshold. This approach causes the system to remain in a stable condition (i.e. under loaded). This has the effect of minimizing loss, resulting in a system that trades load for delay. Such approaches are not applicable to Ad-Hoc wireless networks where the supply is constantly changing (and in many cases infeasible/expensive to increase).

3.3 Emergent Properties

Given that there is a finite amount of quality degradation that can be traded, what restrictions do current queuing mechanisms place upon the trades? In order to answer this question we must understand what properties emerge from the operation those mechanisms.

The emergent properties of a queuing mechanism affect its ability to deliver a predictable QoS to the end-user/application. The following properties are those that are of interest, some of which we examine in this report:

3.3.1 Isolation

At a network element there are always a number of arriving flows. These flows are grouped together into classes, where each class aspires to some pre-defined level of service. Each of these classes places a demand on the common resources within the network element and as such may have an effect on the emergent properties that other classes experience.

There is a range of potential isolation:

Complete Isolation The emergent properties of a given class are only affected by the traffic offered by that class (intra-flow), and are not affected by traffic in any other class (inter-flow).

Partial Isolation	In most network elements that provide differential service, classes affect other classes, and are affected by other classes. This relationship needs to captured and understood.
No Isolation	Any arriving flow can have an arbitrary effect on other flows (e.g., FIFO queuing mechanism).

3.3.2 Differential Treatment

For a given class there can be quantitative differences in the delivered quality. This can be expressed in terms of the following attributes -- throughput, loss and delay (and their moments and derivatives, such as jitter). To compare different approaches we must capture the queuing mechanism's ability to differentially share out the total quality degradation amongst the classes. We can characterize the behavior as follows:

Managed	The queuing mechanism explicitly manages the parameter and the parameter can be configured as desired;
Resultant	The observed quality of a class is the result of its behavior (intra-class) and not on the behavior of other classes (inter-class); and
Emergent	The observed behavior of a class is the result of its behavior (intra-class) and on the behavior of (some or all) other classes in the system (inter-class).

3.3.3 Saturation Behavior

In all networks there are periods of time where the demand at a network element exceeds the capacity available; this is especially true of Ad-Hoc or low-speed networks. When a network element approaches (and even exceeds) saturation, the total quality degradation increases. This may cause some or all of the classes to exceed their contracted quality. During such periods we wish to know, a-priori, the "order of failure"; the order in which the classes fail to meet their quality constraints. In addition we need to understand how "stiff" that delivered quality constraint is, i.e. how susceptible it is to small changes in its input.

3.3.4 Fairness

When a number of flows are aggregated into the same class they compete for service. However, it is possible for some flows to get better service than others - which is clearly unfair! This property is undesirable in mission- or safety-critical environments, where it is accepted that some people will lose; it must be assured that the same person does not lose repeatedly.

4 Traffic Types

In order to evaluate the emergent properties of a number of queuing mechanisms it was necessary to generate some synthetic traffic. This traffic is generated by a number of synthetic applications. For each of the applications we have derived a per-hop and an end-to-end quality degradation budget. In addition to this the applications have "to fail", that is be delivered degradation in excess of their budget, in a given order.

The synthetic applications are modeled using a stochastic labeled transition system. The result is a set of applications with well known behaviors. When a number of instances of the applications are aggregated, as required, the resulting traffic pattern is again well known. Each of the applications is written to generate traffic in a pattern as close to a real application as is possible.

In this section we will describe each of the applications, and their quality constraints in terms of a per-hop and end-to-end quality degradation budget. We will also outline the number of instances of each application used and the load this generates.

4.1 Synthetic Application Descriptions

Below is a description of each of the applications. For a more detailed discussion of the application behavior and their associated constraints please refer to *Annex I of this report*.

4.1.1 Alarm

The alarm application is intended to convey some sort of emergency signal to a number of subscribers. This could be for example a fire alarm; when activated the klaxons will receive the signal and start sounding. Clearly this kind of application will be rarely used; however, we will ensure that it is used at least once in the tests. The alarm application must work over 10 hops; this is to ensure that an emergency signal can be propagated over a network with a potentially large diameter. To achieve acceptable levels of reliability, the alert will be repeated every 10th of a second, for one second.

4.1.2 OSPF

The routing protocol OSPF will be used between routers in this network. OSPF is a relatively complex protocol to model; for the purposes of this experiment, we will assume that the network is stable. This means that there will be no link state updates for the duration of the tests. As such we can assume that the only communication happening on the network will be the Hello packets between the routers. We are assuming a hello time of 10s and a dead time of 40s.

4.1.3 VoIP

The users of this wireless network use Push-To-Talk (PTT) Voice-over-IP (VoIP) to communicate. This type of VoIP traffic requires no control traffic; it simply passes the audio data between the parties. We are assuming that a user waits for a 20s negative exponentially distributed amount of time between transmissions, and then transmits for 10s (again exponentially distributed). The requirement is the deliver the users with a PESQ score greater than 3.0.

4.1.4 Photos

Video cameras placed around the network take still photos of their environment every 30s. The cameras take photos at QCIF resolution (144x176) at 24bps. Each of the frames is compressed at a ratio of 15:1, resulting in 5k per frame. These frames are then transmitted across the network using TCP. The photos have to reach their destination within an average of 3s.

4.1.5 Video

In addition to the still photo capture, presented above, it is possible to stream video over a single hop. This comprises of a maximum of 30fps of QCIF 24bps video (when there is sufficient bandwidth, if not the rate is reduced). A UDP based streaming protocol is used to transmit the data. The camera is in operation 90% of the time.

4.1.6 RPC

Users of the network have access to an HTTP-style web-based application. This application makes RPC style requests to a server located no more than 3 hops away. Each request is no bigger than 500 bytes in size, and the responses are broadly negatively exponentially distributed with a mean size of 10k bytes. Clients can request information in one of two modes, normal, or priority, the selection of which changes the time to complete. Users request a page approximately every 10s for normal priority.

4.2 Traffic Loading

The following table show the load that the applications place on the network when they are aggregated. We assume that there is a seven byte overhead per-packet due to link encoding. For each application we show the mean rate that traffic is generated when the application is being used (remember that instances have idle periods). Next is the number of instances used in the tests, and finally the average load (in bps at L2) offered to the network. Overall the total load placed on the network is approximately 1Mbps.

Application	Mean (on) Rate Per Instance (bps)	Number of Instances	Mean Offered Load (bps)
Alarm	21.0K	1	21.0K
OSPF	56	1	56
VoIP	17.7K	8	141.6K
Video	602.8K	1	602.8K
Photos	13.6K	5	68.0K
Normal RPC	16.9K	8	135.0K
Priority RPC	42.1K	1	42.1K

Figure 1: Application Traffic Loads

4.3 Quality Budgets

The following table shows the per-hop and end-to-end quality budgets for each of the applications. For a detailed description of the process used for deriving these constraints please refer to *Annex I* of this report.

Application	Per-hop Budget		End-to-end Budget	
	Delay (ms)	Loss	Delay (ms)	Loss
Alarm	65	6.00%	1000	50%
OSPF	1000	5.00%	1000	5%
VoIP	50	1.50%	150	5%
Video	1000	1.00%	1000	1%
Photos	100	8.00%	100	8%
Normal RPC	100	0.60%	300	2%
Priority RPC	65	0.30%	200	1%

Figure 2: Application Loss & Delay Quality Budgets

In addition to the quality constraints per application, an order of failure for meeting these quality constraints has been established. That is, when the system is unable to deliver all of the required quality constraints the applications *should* fail in a predefined order. In this case, Best Effort should fail first, then Video, then Normal RPC, etc. The last application to fail *should* be Alarm.

Order of Failure	Application
Last	Alarm
7	OSPF
6	Priority RPC
5	Photos
4	VoIP
3	Normal RPC
2	Video
First	Best Effort

Figure 3: Order of Failure

5 Test Overview

5.1 Test Setup

The test environment that was used is shown in Figure 4: QoS Test bed Topology. At the center is a Cisco 7204VXR router and is the “unit under test”. It is connected via a serial link to a Cisco 3725 router that will be used as a media converter. The combination of these two components is used to achieve low link speeds. On each side there are two switches. The switches are configured to mirror the port connected to the device under test, sending the data to a single measurement machine. Machines connected to the switches function as the traffic generators and hosts.

The test bed equipment details are:

- Cisco 7200VXR Series Router NPE-G1 with 512MB of memory running on Cisco IOS software version 12.3(14)T2;
- Cisco 3725 Multi-service Access Router with 256MB of Memory running Cisco IOS software version 12.3(14)T1;
- Two Cisco 3550 XL Series Switches running on Cisco IOS software versions 12.0(5.2)XU and 12.1(22)EA3;
- Cisco 3750 Series Switches running on Cisco IOS software version 12.2(20)SE3; and
- Five workstations running on Linux version 2.4.27-1-386 and Debian version 1:3.3.4-9

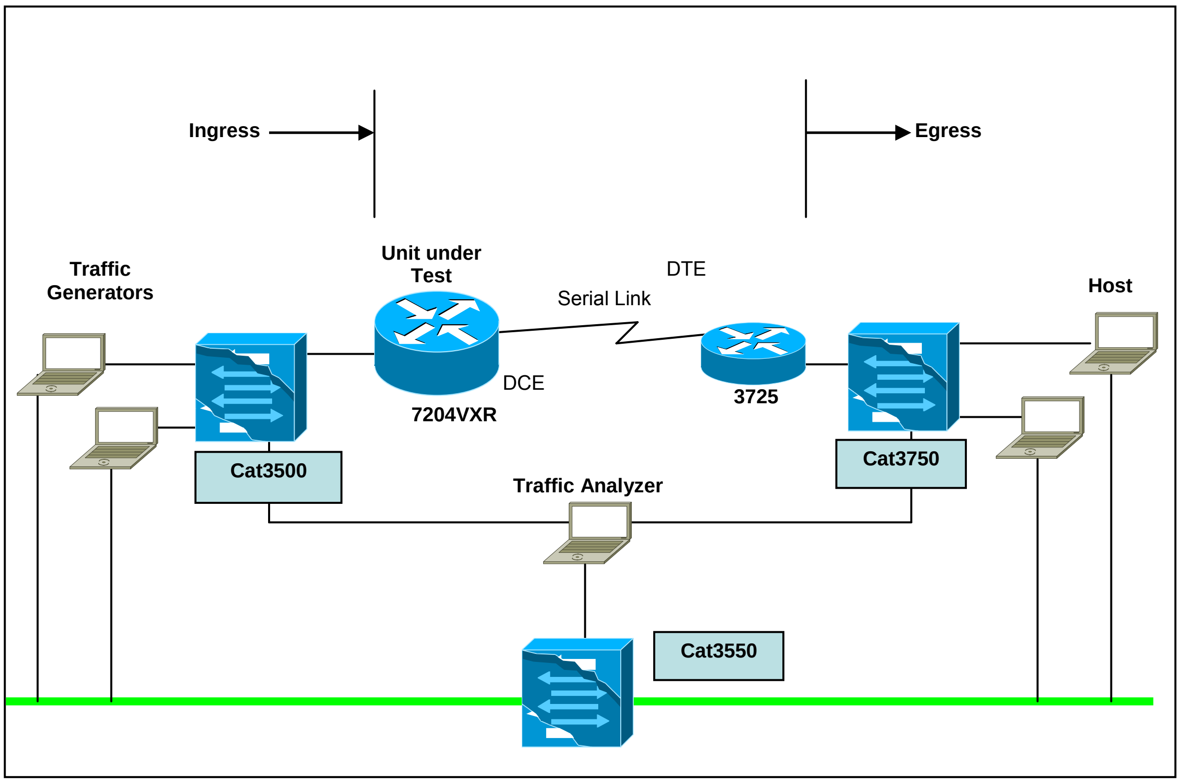


Figure 4: QoS Test bed Topology

The traffic generator workstations have been configured to create packet traffic with the required properties in both data volume and timing. Separate workstations are used to generate the background traffic so that the workstations are not overloaded to ensure that the timing is as accurate as possible. The workstations are unaware of the monitoring that is occurring, and essentially will see the network as they would under operational conditions.

Traffic flows from the left side of the figure to make sure that contention occurs in the device under test. A network tap is taken on either side of the device under test and fed into the measurement workstation. This workstation records both flows of incoming packets. As there is a single machine that is receiving both flows, the packets can be time stamped with relative precision.

To emulate the properties of multiple link speeds the serial link between the unit under test (7204VRX) and the 3725 was configured at the speed under test. These speeds are enumerated in sections 5.2 and 5.3 below.

By comparing the two traces we are able to deduce the distributions of:

- Loss
- Delay
- Throughput

These comparisons can be used to understand the emergent properties of the device under test.

5.2 Phase I – Fixed Speed Link Testing

Prior to sending the traffic, each application and instances within the application is given a random seed number. This random seed is used to ensure that the generated traffic is difference each time. Each test runs for approximately 5 minutes and is controlled by a script (running on the management workstation) that will:

- Populate all instances of all applications with the random seeds;
- Instruct the traffic generators to initial the TCP and UDP connections;
- Wait for approximately two minutes to let the connections establish;
- Start sending the applications traffic for approximately two minutes;
- When finish, close out all the connections; and
- FTP all the results back to the management workstation for analysis.

This process is repeated 20 times for each link speed and queuing mechanisms.

The fixed link speeds tested were:

- 2Mbps
- 1Mbps
- 512Kbps
- 256Kbps
- 128Kbps

5.3 Phase II – Multi-Speed Link Testing

In the second phase a scenario to test how the router (unit under test) and its queuing mechanisms work in a changing physical link environment was developed. The exact same test bed and test tools were utilized to derive the data. Two changes to the testing environment were made to simulate the multi-speed environment:

- the physical link speed was changed while traffic was sent into the test bed;
- the process was repeated 20 times for each scenario; and
- when the link speed was reduced the original (higher speed) configuration was kept for 30 seconds before the configuration associated with the lower link speed was applied – to expose issues associated with the transition.

The multi-speed links tested were:

- 2Mbps to 1Mbps;
- 1Mbps to 512Kbps;
- 512Kbps to 256Kbps; and
- 256Kbps to 128kbps.

5.4 Test Tools

These are the tools that are used during the tests:

Traffic Generator	Refer to Section 4 above for a detailed description of the method used to generate the application traffic.
NETCAT	A utility that is available in many flavors of UNIX Operating Systems (OSes), was used for generating background TCP traffic. NETCAT is a utility that allows data to be piped between two machines using a TCP connection. The background traffic was generated by sending a large file of random data using NETCAT.
TCPDUMP	A packet-capture utility that is available in many UNIX OSes.
Traffic Analyzer	Consists of a set of scripts that were used for parsing the captured packets and calculating the loss, delay and offered load and then comparing them to the requirements.

5.5 Queuing Mechanisms Under Test

In an attempt to effectively manage congestion on network interfaces, various queuing mechanisms are used. These mechanisms endeavor to ensure that applications are properly prioritized and that critical packets - from applications such as voice-over-IP, Telnet, and stock trading, where latency and jitter are unacceptable - get through with the highest priority. For the purpose of this investigation the team chose the following queuing mechanisms to test:

- First In, First Out (FIFO);
- Strict Priority Queuing; and
- Low Latency Queuing (LLQ).

In the industry there are other queuing mechanisms which were not tested, such as Weighted Fair Queuing (WFQ), Weighted Round Robin (WRR) and Modified Deficit Round Robin

(MDRR). Though important, these mechanisms were not chosen to limit the scope of the testing.

5.5.1 First In, First Out

First In, First Out (FIFO) queuing is the most basic queue scheduling discipline. In FIFO queuing, all packets are treated equally by placing them into a single queue, and then servicing them in the same order that they were placed into the queue. FIFO queuing is also referred to as first come, first served (FCFS) queuing. FIFO has many issues especially when deployed on a network providing converged services since it does not have any concept of priority.

5.5.2 Strict Priority Queuing

Strict Priority Queuing is a legacy queuing mechanism implemented in Cisco IOS software. As the name implies, this queuing mechanism provides strict priority in selecting which packets are sent first on an interface. There are four output queues with four levels of priority or classes of service in Strict Priority Queuing:

1. High-Priority;
2. Medium-Priority;
3. Normal-Priority; and
4. Low-Priority.

Users can assign packets to the difference priority queues based on TCP or UDP ports, protocols and Access Control List. The way Strict Priority services these four queues is packets in the High Priority queue will always get serviced before packets in the Medium-Priority queue, and so on. This may prevent the lower priority queues from being serviced at all and lead to discarding of packets if the queues become full. Strict Priority works well in environments where traffic has a hierarchy of importance.

5.5.3 Low Latency Queuing (LLQ)

Low Latency Queuing (LLQ) is the newest queuing mechanism in Cisco IOS software. It retains some of the characteristics of Strict Priority Queuing with the added ability to limit the service time given to the strict priority class. LLQ enables the use of a single priority queue within which individual classes of traffic can be placed. The Strict Priority Queuing scheme possible with LLQ allows delay-sensitive traffic such as voice to be taken out of the queue and sent first - before packets in other queues are sent. In other words, delay-sensitive data is given preferential treatment over other traffic.

LLQ enables the use of a single, strict priority queue within Class Based Weighted Fair Queuing (CBWFQ) at the class level. One or more classes may be given priority status. When multiple classes within a single policy map are configured as priority classes, all traffic from these classes is queued to the same, single, strict priority queue.

Class Based Weighted Fair Queuing (CBWFQ) uses packet classification features that allow traffic to be partitioned into multiple class levels, or classes of service. Packets can be classified in a variety of different ways, ranging from input interface, Layer 3 and 4 header information, to NBAR (Network Based Application Recognition) for difficult to classify applications, to arbitrary access-control lists. You can also mark packets in a variety of ways using the policy-framework component of the Modular QoS CLI - Layer2-802.1p/Q / ISL, ATM CLP bit, Frame-Relay DE-bit, MPLS EXP bits, Layer3 IP Precedence, and Differentiated Services Code Point (DSCP) bits.

Without LLQ, CBWFQ provides queuing based on defined classes with no strict priority queue available. CBWFQ allows for the definition of traffic classes and then allows characteristics (e.g., the minimum bandwidth to be delivered to the class during congestion) to be assigned to each class. For CBWFQ, the weight for a packet belonging to a specific class is derived from the bandwidth assigned to the class when it was configured. Thus, the bandwidth assigned to the packets of a class determines the order in which packets are sent. All packets are serviced fairly based on weight, and no class of packets may be granted strict priority. This poses problems for voice traffic that is largely intolerant of delay, especially variation in delay.

For this paper the setting of the DSCP bits were used to determine priority.

5.6 Configuration Strategies

5.6.1 Low Latency Queuing

The seven applications listed above were divided up into five classes with different combinations depending on the link speed. However, the three classes below are common across all link speeds because on the order of failure, service level agreements, underlining bandwidth and protocol type.

ALARM_OSPF	Alarm and OSPF application traffic
PRCpri	Priority RPC application traffic; and
VoIP	VoIP application traffic.

Since Alarm and OSPF traffic are the most critical applications, they are assigned to the high priority LLQ to ensure that these traffic types are delivered in a timely manner. All other applications are assigned to other classes with the appropriate bandwidth allocation.

> **Note**: This differs from Cisco's QoS best practice recommendations where the LLQ is designed for applications that are delay and loss sensitive, such as Voice over IP applications.

Given the strict application failure order, the QoS provisioning on the router will need to adapt to the changing link bandwidth. For instance, at 128Kbps, it is impossible to satisfy all applications. In such a case, the router needs to aggressively drop the least important traffic in order to meet the higher importance applications' requirements. As bandwidth improves, the router's QoS configuration will also need to change to allow more traffic through.

The initial amounts of bandwidth allocated are based on the average requirements as documented in Section 4.2 above. The allocation was tested and modified and tested again until a solution that satisfied the quality restraints was found. The bandwidth is allocated in percentages of the link to make the configuration more scalable and manageable.

Since Alarm and OSPF are absolutely critical applications and required approximately 17% of the 128 kbps link, it is assigned to the LLQ class with 20% of the link bandwidth. The remainder of the bandwidth is assigned to the next priority application, priority RPC. As the link speed increases, the amount of bandwidth assigned to the classes increase proportionally to the increased bandwidth. However, at the 512 kbps and 1 mbps link speed, the bandwidth assigned to the classes required some trials and errors in order to come to the right amount to accommodate as many applications as possible without degrading the quality of service.

In addition to allocating bandwidth to each class, Weighted Random Early Detect (WRED) was employed to help actively manage the queues. Figure 5: Sample LLQ Configuration below shows a sample QoS configuration.

```
class-map match-any ALARM_OSPF
match ip dscp af41
match ip dscp af21
class-map match-any VoIP
match ip dscp ef
match ip dscp cs5
class-map match-any RPCpri
match ip dscp af43
class-map match-any RPCnor_VIDEO
match ip dscp af33
match ip dscp af23
class-map match-any PHOTO
match ip dscp af31
!
policy-map FCS
class ALARM_OSPF
priority percent 5
class RPCpri
bandwidth percent 75
random-detect dscp-based
random-detect exponential-weighting-constant 1
random-detect dscp 38   5     8     20
class PHOTO
bandwidth percent 10
random-detect dscp-based
random-detect exponential-weighting-constant 1
random-detect dscp 26   5     10    8
class VoIP
bandwidth percent 10
class RPCnor_VIDEO
class class-default
```

Figure 5: Sample LLQ Configuration

Below is the list of Cisco IOS commands used to capture the router configuration, QoS policy, queues and interface information:

- show running;
- show policy-map interface <interface name>;
- show queue <interface name>; and
- show interface <interface name>.

5.6.2 Strict Priority

Strict Priority only consists of four queues; high, medium, normal and low. The seven applications are assigned to the queues based on their failure order as shown below:

High Priority Alarm and OSPF applications;

Medium Priority Photo and Priority RPC applications;

Normal Priority VoIP, Normal RPC and Video applications; and

Low Priority All other applications.

The queue sizes (in packet) for all five link speeds are shown in the table below. These numbers were arrived at through a series of trial and error configurations.

Link Speed	High	Medium	Normal	Low
128Kbps	1	40	60	80
256Kbps – 2Mbps	1	20	60	80

Figure 6: Strict Priority Queue Sizes

Figure 7: Strict Priority Configuration below contains a sample configuration.

```
access-list 100 permit ip 10.10.10.0 0.0.0.255 10.10.30.0 0.0.0.255 dscp ef
access-list 110 permit ip any any dscp default
priority-list 1 protocol ip high udp 10001
priority-list 1 protocol ip high udp 11001
priority-list 1 protocol ip normal tcp 15000
priority-list 1 protocol ip normal udp 13001
priority-list 1 protocol ip low list 110
priority-list 1 protocol ip medium tcp 16000
priority-list 1 protocol ip normal list 100
priority-list 1 protocol ip medium tcp 14000
priority-list 1 queue-limit 1 20 60 80
```

Figure 7: Strict Priority Configuration

Below is the list of Cisco IOS commands used to capture the Strict Priority configuration and interface information:

- show queuing priority;
- show queuing <interface name>; and
- show interface <interface name>.

5.7 Addition of Fragmentation

In order to provide better latency to high priority traffic at low speed links, it was determined that it was necessary to turn on Cisco's Link Fragmentation and Interleaving (LFI) for Multilink PPP (MLP) feature. Without this feature turned on, interactive traffic such as OSPF and VoIP had no chance of meeting its required quality requirements due to potential head-of-line blocking on the low-speed links.

Unacceptable queuing delays for small real-time packets exist regardless of which queuing mechanism was used. Head-of-line blocking is caused by the servicing of large packets. Currently in Cisco IOS LFI for MLP is implemented only in conjunction with LLQ.

The LFI scheme is relatively simple: Large datagrams are multilink encapsulated and fragmented to packets of a size small enough to satisfy the delay requirements of the delay-sensitive traffic; small delay-sensitive packets are not multilink encapsulated, but are interleaved between fragments of the large datagram. This solves the issue of head-of-line blocking and provides an environment that allows real time traffic to better meet its quality requirements.

6 Test Cases

To provide a complete set of test data the following six test configurations were defined. The configurations where:

- FIFO
- Strict Priority
- Low Latency Queuing (LLQ) with no best effort traffic and LFI turned off
- LLQ with best effort traffic and LFI turned off
- LLQ with no best effort traffic and LFI turned on
- LLQ with best effort traffic and LFI turned on

Each configuration was then executed at all the prescribed link speeds as outlined earlier in this paper and then again in the multi-speed scenario.

6.1 Fixed Link Speed Scenarios

The execution of a test was as follows:

1. Configure the link speed to the rate under test;
2. Start the background traffic generators and test applications;
3. Allow all the connections/applications to come up to speed (120 seconds);
4. Run with link speed unchanged (60 seconds);
5. End test

6.2 Multi-Speed Link Scenarios

To produce the desired effect of a changing link speed, the testing scenario was designed to produce this. We used this function to simulate the effect that changing link capacity had on the performance of the queuing mechanism under test. In each test the following procedure was used:

1. Configure the link speed to the highest rate under test (2Mbit/sec)
2. Start the background traffic generators and test applications
3. Allow all the connections/applications to come up to speed (120 seconds)
4. Run with link speed unchanged (60 seconds)
5. Reduce the link speed to next lowest speed – leaving configuration alone, if already at lowest link speed go to step 9
6. Run in this arrangement (30 seconds)
7. Reconfigure elements as required
8. Go to step 4
9. End test

Throughout this entire sequence, data is being collected. This means that it is possible to analyze the effects before, during and after the change in link speed. It is important to note that the configuration of the device under test is not changed at the same time as the link speed is changed. In real world scenarios, the configuration may be automatically generated; therefore, we need to know the properties of the previous configuration as it is pushed into saturation. This is key, as it will highlight some of the emergent behavior of the queuing mechanisms that we are interested in.

6.3 Results

6.3.1 Analysis Approach

The traffic analyzer is capturing packets at the ingress and egress to the simulated “hop”; the packets are time-stamped as they are recorded in the trace.

The two timed traces (ingress and egress) are compared to generate a single trace in which each ingress packet is annotated with either a) lost marker or b) the delay that packet experienced in being transported across the simulated hop. This is the quality degradation annotated trace (the ΔQ trace).

This ΔQ trace is divided into individual traces for each of the application classes.

The perceived quality of experience of an application is dependent on the treatment of that application’s traffic over some period of time. To assess the “quality” of the stream (as experienced by each application) the individual application classes’ ΔQ trace is processed in the following way:

1. The delivered loss and delay is averaged over a suitable interval (0.25sec in this case).

    - This gives, for that time interval, the quality degradation experienced.
    - All intervals in which there was no ingress packets are discarded.

2. The fraction of the end-to-end quality degradation budget is calculated.

- Initially this is done separately for the end-to-end loss budget and the end-to-end delay budget.
- To reduce these two numbers to a single value we choose the larger figure from either the loss budget consumed or the delay budget consumed[2].

3. This gives a population of samples that approximate the instantaneous quality degradation for each application class.

As part of the evaluation, approximately twenty runs where made at each line speed. Each run used different random seeds to prime the application simulators.

Figure 8: Sample QoS Summary Graph for Strict Priority Queuing at 1Mbps illustrates the composite results for all the test runs using Strict Priority Queuing with a link speed of one megabit per second.

The individual points on horizontal lines are the average quality degradation experienced for a particular application and a particular run. These are plotted as a percentage of the total end-to-end degradation budget for that application (as defined in Figure 2: Application Loss & Delay Quality Budgets). The vertical marks represent the per-hop budget as a fraction of the end-to-end budget.

If all of the samples, for a particular application, are to the left of the vertical mark then the configuration "works" – all of the test runs produced average results that were within the allocated budget.

[2] This is an approximation for the purposes of graphical representation; a better choice may be to incorporate into this calculation the "distance" to the contour line in loss/delay space that represents equal quality of outcome at the application level.

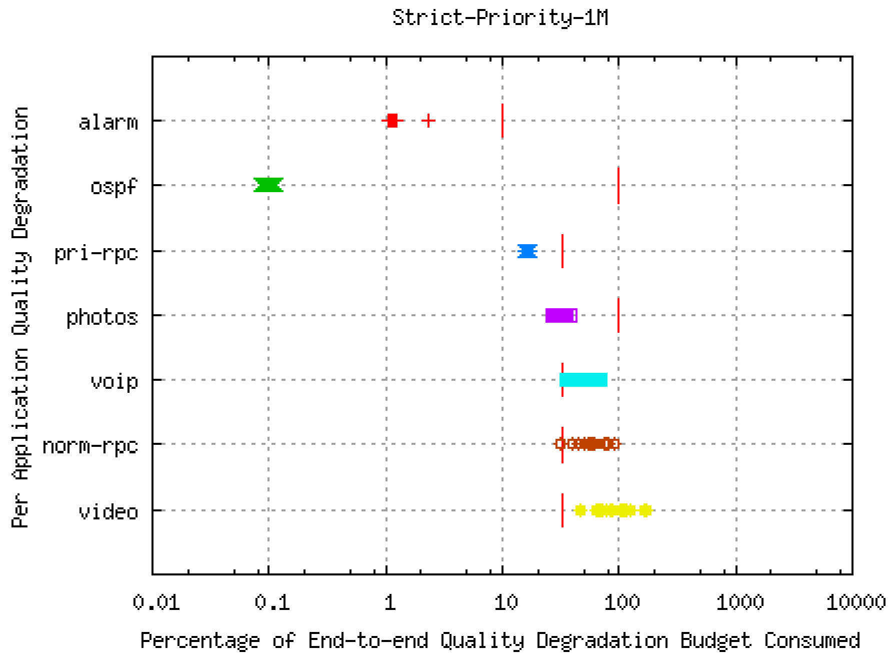


Figure 8: Sample QoS Summary Graph for Strict Priority Queuing at 1Mbps

6.3.2 Charting Results

6.3.2.1 Consumed DQ Scatter Diagrams

The "Consumed DQ Scatter Diagrams" is representing, for a given queuing discipline at a given speed, the amount of end-to-end budget (in terms of both loss and delay) that an application has consumed. The vertical red bar represents the end-to-end budget. So being to the left of that mark is goodness. Figure 8 is an example of a “Consumed DQ Scatter Diagram” chart. The complete set of graphs can be found in Appendix A.

6.3.2.2 Link Utilization Percentages

The "Link Utilization Percentages" are split into three graphs; offered load, transported load and load difference. There are multiples of these for each of the link speeds. All three types of graphs have the application of interest on the x-axis, and two y-axes. The left y-axis is the absolute volume of traffic in bits per second, the right is the fraction of the outgoing link that that volume represents at that speed.

The "Offered Load" graph (Figure 9) shows you the amount of traffic that was offered to the device under test. The two bars per queuing discipline represent the total amount of offered traffic, and the total amount of traffic offered that we are required to provide quality guarantees ("quality" traffic).

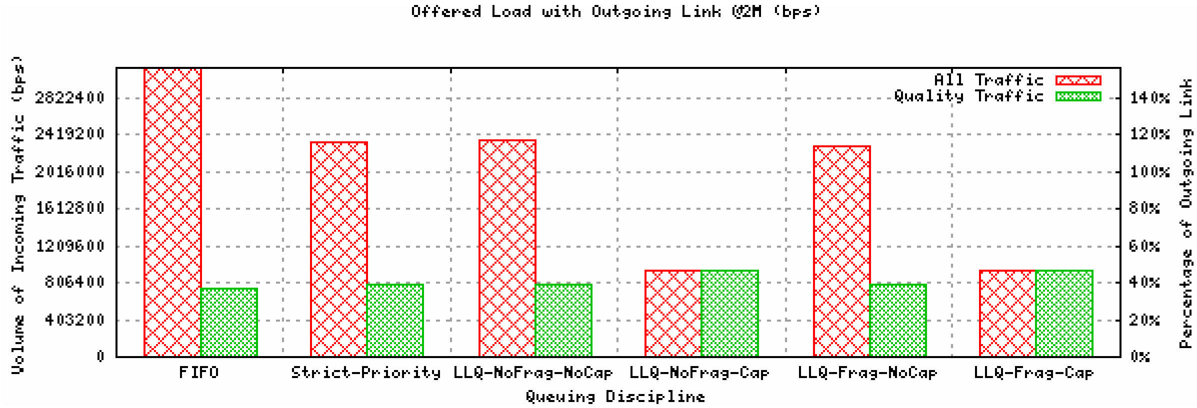


Figure 9: Offered Load

The "Transported Load" graph (Figure 10) has three bars per queuing discipline. The first bar "All Traffic" shows the total amount of traffic that was transported. The second bar shows that amount of "quality" traffic transported. The third bar represents the amount of "quality" traffic that was transported in-contract; that is satisfied its quality constraints. Think of the first bar as the total amount of work, the second as the amount work done for quality traffic, and the third as the amount of useful work done for the quality traffic.

It should be noted that, for the LLQ-NoFrag-Cap and LLQ-Frag-Cap scenarios the least important application's traffic was policed away within the network element. This assured that the transported load was within the available link capacity.

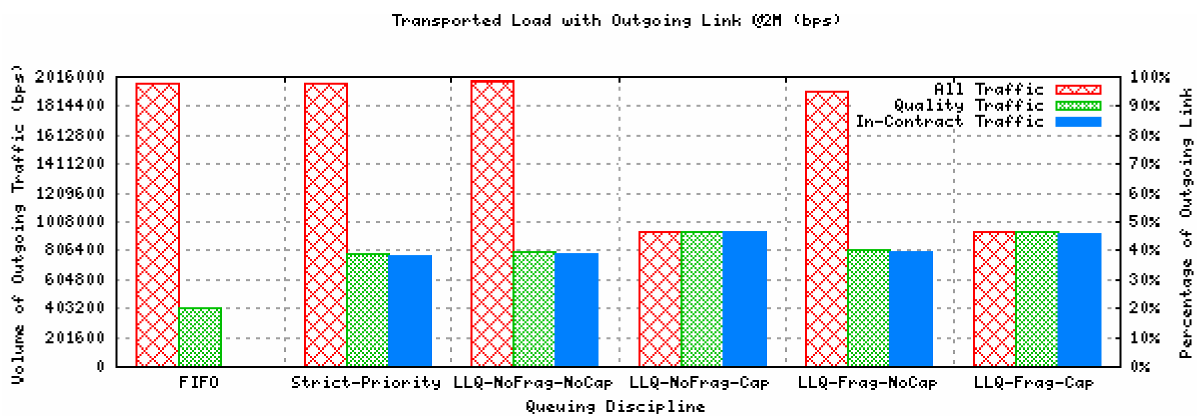


Figure 10: Transported Load

The final graph "Load Difference" (Figure 11) again has three bars per queuing discipline. The first bar "Discarded - All traffic" represents the difference between the total load offered and the total load transported. So if we offered 2M, and the link was 1M and fully loaded, then this bar will read 1M (because that's what we had to discard). The second bar "Discarded - Quality Traffic" is the difference between the offered and transported load for the "quality" traffic. The third bar is a little more interesting. It represents the amount of "quality" traffic that we transported that was out of contract - this is the difference between the "Quality Traffic" and "In-contract Traffic" bars as shown on the "Transported Load" graph.

In the “Capped” cases it should be noted that all best-effort traffic was policed away in the network element.

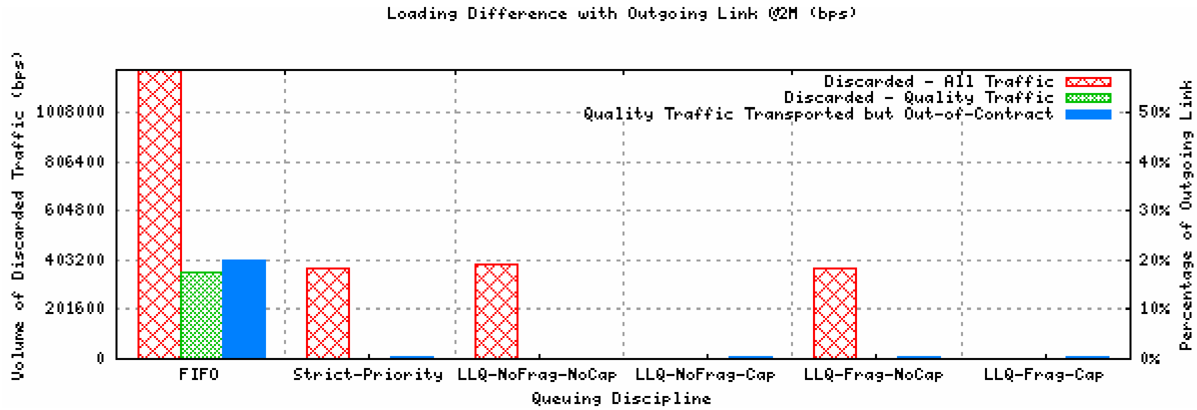


Figure 11: Loading Difference

6.3.2.3 Per-App Percentage in Contract

The final set of graphs "Per-App Percentage in Contract" is again split into three graphs. All of which are per queuing discipline, per speed. To produce these graphs we split the whole test run into a number of time intervals, all of which were 25ms long. For each one of these time intervals we calculated the observed loss and delay for a given application (under that queuing discipline, at that speed).

The first graph “In-Contract” (Figure 12) represents all of the percentage of the time intervals which were within their loss AND delay constraints. The second graph “In Delay Contract” (Figure 13) represents the percentage of intervals that were within the delay constraints. The third graph “In Loss Contract” (Figure 14) represents the percentage of intervals that were within the loss constraints.

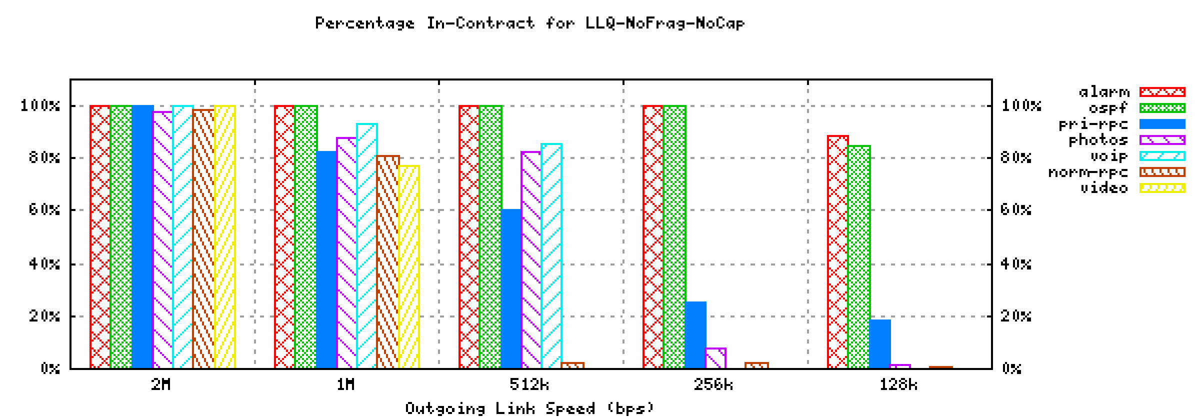


Figure 12: In-Contract

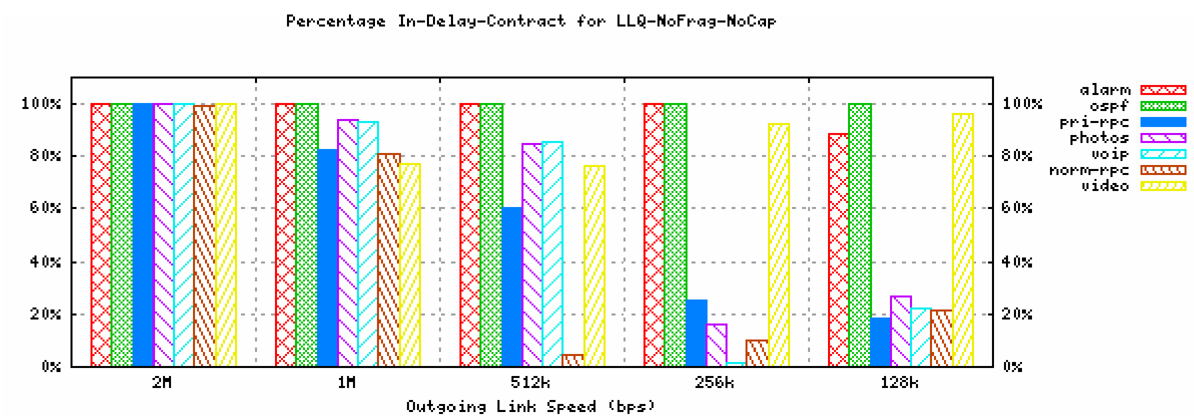


Figure 13: In-Delay Contract

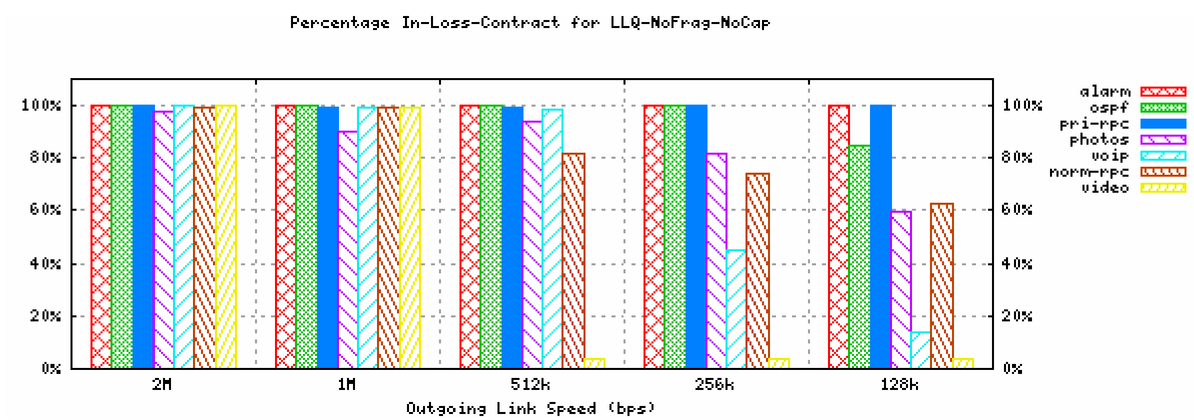


Figure 14: In-Loss Contract

6.3.3 Observations

6.3.3.1 Fixed Speed Link Test Cases

From the analysis of the results the following observations can be made:

- The availability of quality bandwidth is limited but queuing structures do provide a mechanism for distributing the available service to permit priority traffic to be delivered in-contract.
  - When the network link reaches saturation levels, the lower priority traffic is either dropped or managed efficiently thus allowing higher priority traffic to meet its necessary quality requirements when queuing mechanisms (e.g., Strict Priority and LLQ) are utilized.
  - To deliver the necessary QoS for high priority applications, queuing structures are necessary to protect higher priority traffic in the network device during time of resource contention.

- Queuing structures do provide the foundation to predict the nature of traffic and its ability deliver QoS across a network device.
    - Strict Priority and LLQ do deliver the ability for applications to fail in the prescribed order thus providing graceful degradation.
    - FIFO does not represent a solution if the outgoing link capacity becomes saturated. Though traffic is delivered, it is out of contract by two orders of magnitude.
- While queuing structures do protect and provide a service to higher priority traffic, it does negatively effect the utilization of a link at saturation because a larger fraction of the link's capacity needs to be allocated to the servicing of quality traffic.
    - The percentage of the traffic that is transmitted in-contract is a direct function of the size of the bandwidth speed of the outgoing link. The higher the bandwidth the better utilization of the capacity.
- Ensuring that the queues do not reach saturation through the policing of best effort and lower priority traffic allows for greater delivery of higher priority traffic.

6.3.3.2 Multi-Speed Link Test Cases

Some unique attributes that were observed during the test cases that involved changing the link speed are:

- The presence of best effort traffic does have a negative effect on the delivery of quality traffic especially during link speed changes. This shows that quality traffic is not totally isolated from best effort traffic.
- Changing of the link speed does affect the ability to deliver traffic in-contract.
    - Quality traffic is transported and continues to have priority but the percentage of out-of-contract traffic increases.
    - Since the priority structures are still intact graceful degradation is usually observed.
- In-contract link utilization is negatively affected when the link speed changes. The test interval we observed this over was 30 seconds.
- To optimize the link utilization and in-contract traffic router configurations do need to be optimized based on the change of speed while using the LLQ mechanism.
    - Continuous optimization of the router configuration may be necessary to make sure that no one application is receiving too much quality as link speed changes.
    - Bandwidth assignment in LLQ can be configured as a percentage of link speed or in absolute Kbps.
        - As bandwidth decreases percentage might not provide enough bandwidth for critical applications.

  - If configured as an absolute then application bandwidth is maintained but the policy will be shut down if configured bandwidth becomes greater then available bandwidth. Queuing will revert to FIFO then.
- This does add complexity and cost to the solution though does increase optimization.
- Configuration changes are not necessary for Strict Priority or FIFO, as they are static configurations.

7 Comparison of Outcomes

7.1 Positives

Throughout this study the basic question has been, "For network elements / connections running at saturation are QoS mechanisms needed and what are the emergent properties of a selection of them?"

The key property of Ad-Hoc networks is that they are not planned, managed and controlled like landline based ones. The overall properties are not ones that can be tested by "build and see", wide variation in traffic load and pattern is expected to be the norm. That, combined with the desire to carry traffic for applications with safety critical features is the reason for this study.

So the evaluation criteria is "do the QoS mechanisms support (via configuration) the creation of the emergent properties needed to support the application set under study?"

7.1.1 QoS Mechanism Value Add

From the results of the 2MB FIFO test run, it can be determined that for a network device to delivery quality traffic something needs to be done to manage the packets while in the device traffic. Utilizing the defined traffic model and the specified quality requirements necessary for traffic to be useful to the application you can see that no traffic was able to meet its quality requirements.

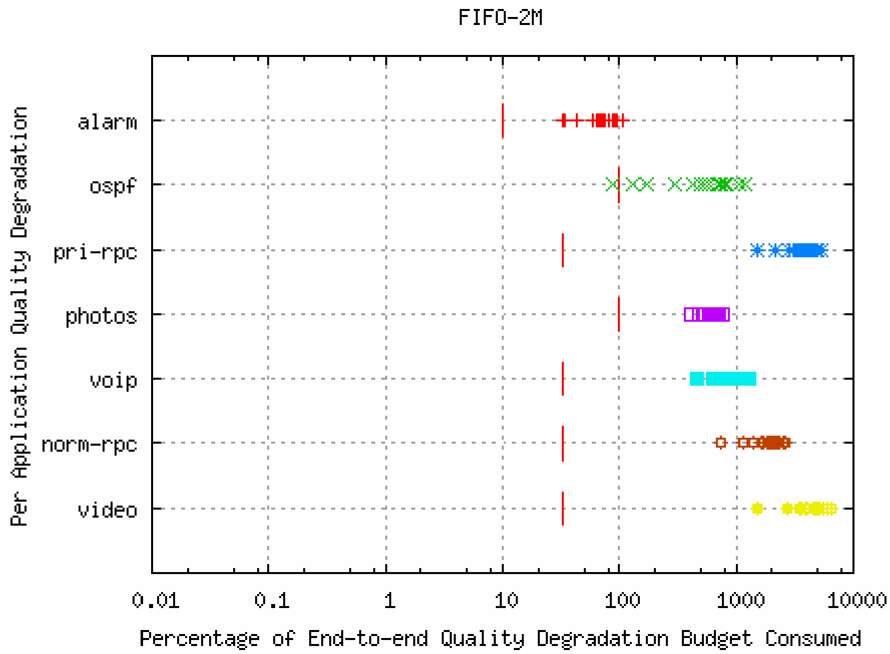


Figure 15: Delivery of Traffic using FIFO Queuing at 2Mbps

As shown in Figure 15, we can say that FIFO queuing is markedly insufficient. While utilizing FIFO queuing at 2MB, traffic is delivered by the router but none is useful to any of the applications. For the traffic to be delivered as quality traffic most if not all of the traffic should plot to the right of the hash.

As you look at FIFO at lower speeds (Figure 16) it can be seen that traffic is continued to be delivered and the traffic distribution remains the same. The only change is that the time to deliver the traffic continues to move to the right. From an application perspective this is dead traffic.

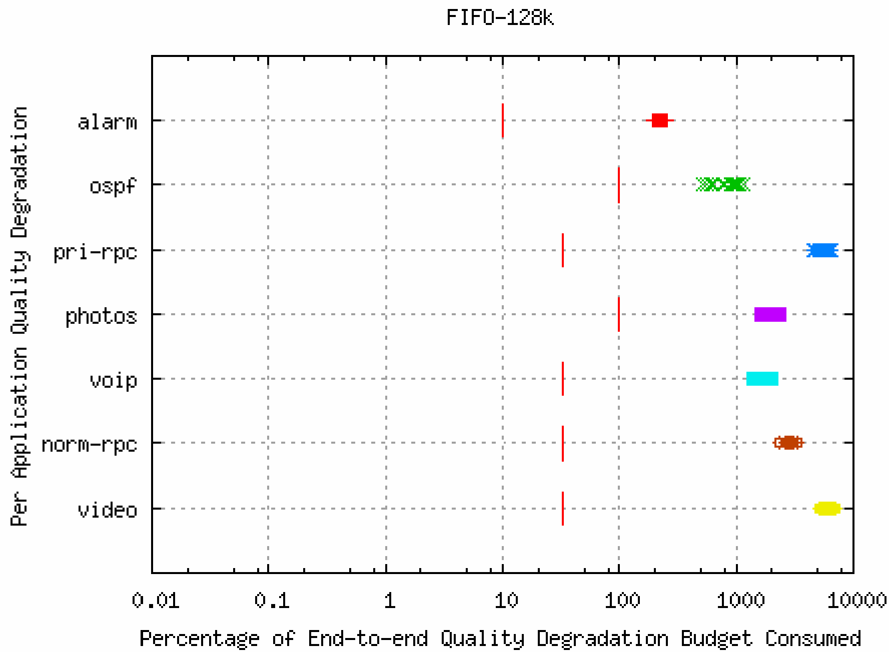


Figure 16: Delivery of Traffic Utilizing FIFO Queuing at 128Kbps

Just the opposite can be seen in the test runs for the 2MB test cases for Strict Priority Queuing (Figure 17) and all of the LLQ cases (Figure 18). In all cases all the traffic was delivered with quality, aka within its quality requirements for it to be useful data to the end application.

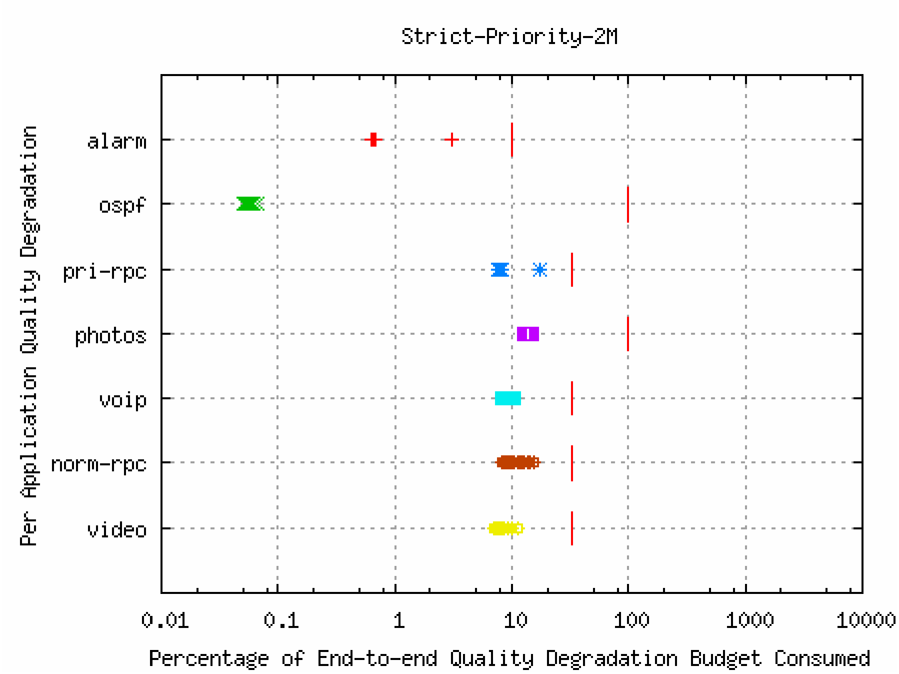


Figure 17: Delivery of Traffic Utilizing Strict Priority Queuing at 2Mbps

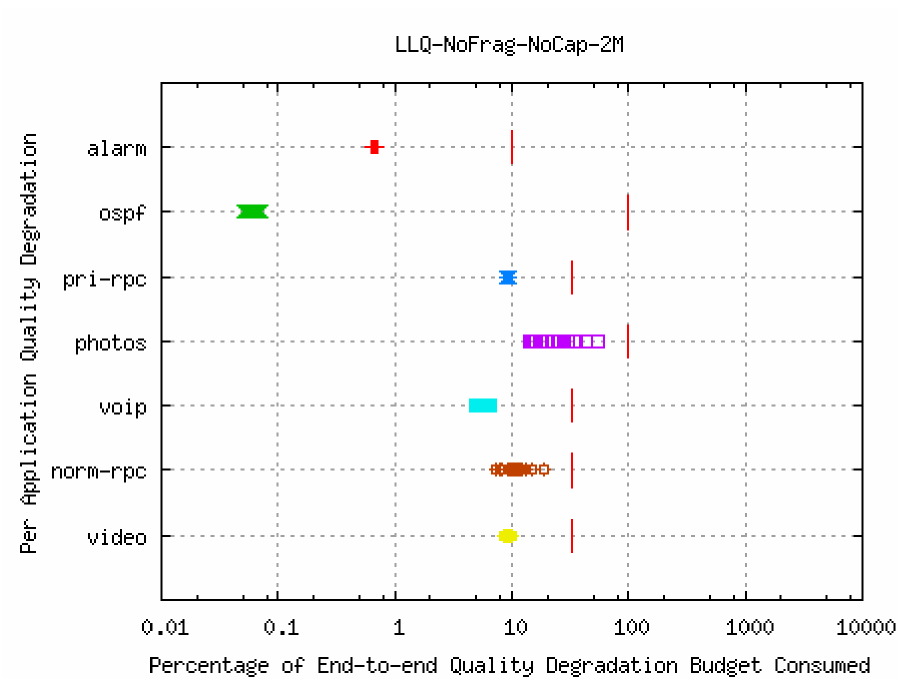


Figure 18: Delivery of Traffic Utilizing Low Latency Queuing at 2Mbps

This is not a surprise since FIFO does not provide any isolation between flows. This is the basic reason why other queuing mechanisms were developed.

At 2Mbps while the queuing does protect the priority traffic, which is a value add, it is fair say that 2Mbps we are not seeing link saturation thus allowing all the traffic to be delivered correctly. The first sign of saturation can be seen in the results for the 1MB test cases.

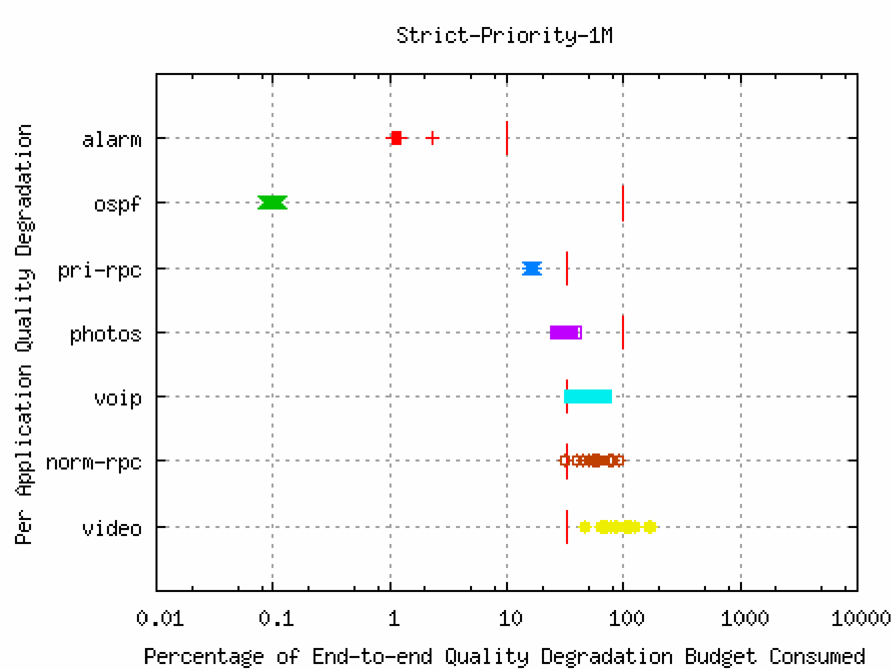


Figure 19: Delivery of Traffic Utilizing Strict Priority Queuing at 1Mbps

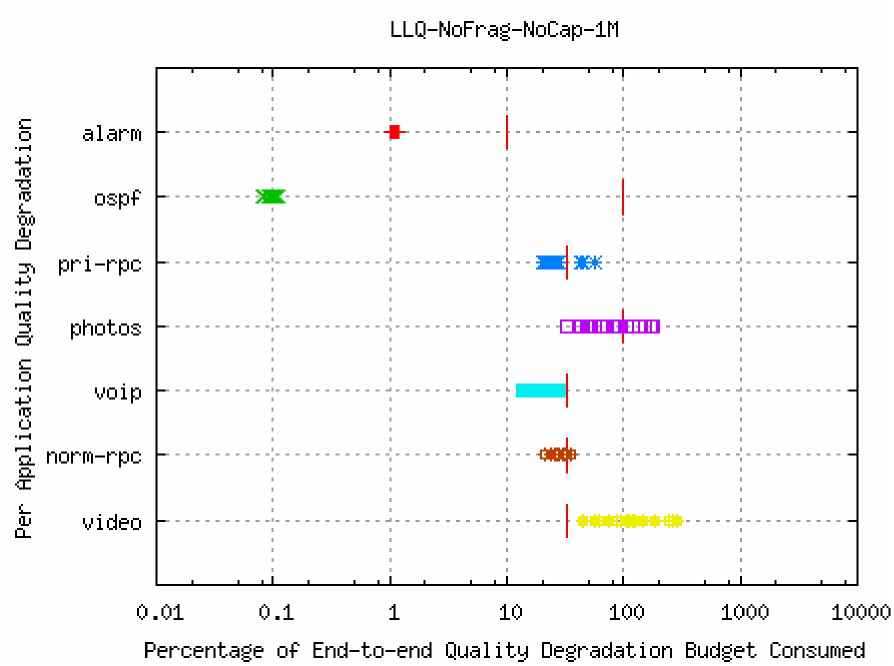


Figure 20: Delivery of Traffic Utilizing Low Latency Queuing at 1Mbps

Here for both Strict Priority (Figure 19) and LLQ (Figure 20) we see that video traffic (the lowest priority traffic) is delivered but outside of its link budget.

At 1MB other observations also become visible based on the properties of the queuing structures. Depending on the queuing structure used, traffic is handled differently. In the Strict Priority model the higher priority traffic is preferential service making just not video but also norm-rpc and VoIP also fall outside of its link budget. In the LLQ model while most of the traffic in the norm-rpc, VoIP, photos and pri-rpc is delivered within its budget a percentage is not. The challenge here is how does this property effect the end applications and can they tolerate some traffic being out of its defined boundary. In a real network, Strict Priority Queuing vs. the efficiency of LLQ would need to be traded to determine the best configuration.

These are just a couple examples of how industry defined queuing do provide value in the delivery of quality traffic across a network node. With the introduction of queuing mechanisms it can be seen that priority traffic is protected as it transverses the node. Another important observation is that queuing mechanisms can also increase the ability and efficiency of a network node to deliver multiple application flows. Priority vs. efficiency is something that needs to be traded by the end customer to determine what is best for them.

7.1.2 Traffic Policing

One of the mitigation strategies employed during test was to introduce traffic policing. This is a technique that allows the router to intelligently discard traffic queued for the egress port. Traffic policing was configured on the best effort traffic in some of the test cases. Where best effort traffic was policed the amount of traffic delivered in contract increased for all traffic speeds.

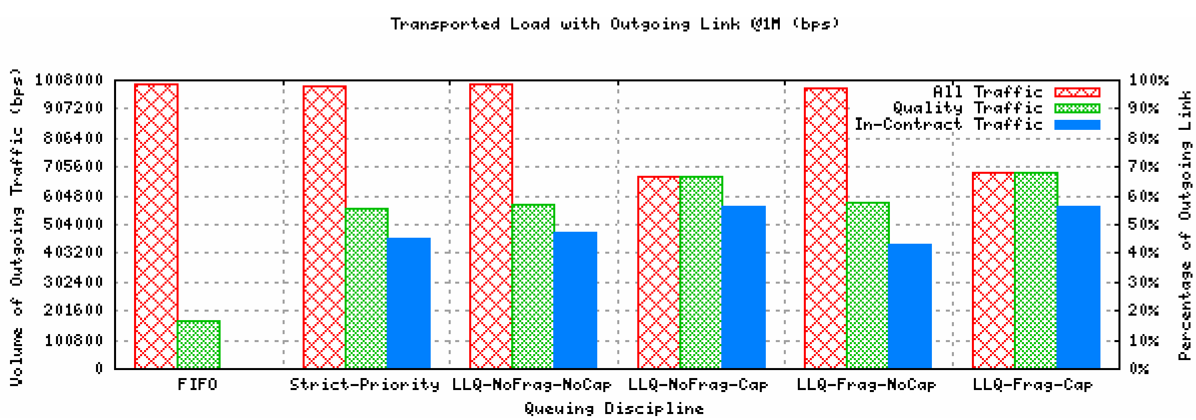


Figure 21: Transported Load at 1Mbps

Figure 21 shows that when best effort traffic is capped or limited the queuing structures are able to better handle the quality traffic and deliver more of it in contract. This is shown in the bars labeled LLQ-NoFrag-Cap and LLQ-Frag-Cap.

This is one example of how mitigation strategies can make a difference in the delivery of quality traffic. The deployment of these mitigation strategies is for future study.

7.2 Negatives

7.2.1 Complexity

LLQ brings a lot of ability to refine and optimize a configuration. Flexibility brings complexity and the increased probably of a mistake in the configuration. Also the issue of autonomy vs. central control comes into question. To optimize a router to deliver quality traffic the configuration may need to be refined every time a speed change is detected. This reconfiguration can bring its own set of issues and concerns.

Since the configuration of Strict Priority and FIFO is static, reconfiguration is not an issue as link speeds change. The pros and cons of a reconfiguration need to be traded to better understand when and if it should be executed.

7.2.2 Link Utilization

As link speeds decrease, the ability to deliver quality traffic is not just a physical issue but also a queuing issue. As shown in Figure 22 saturated links at 2Mbs are still able to deliver most if not all of its quality traffic.

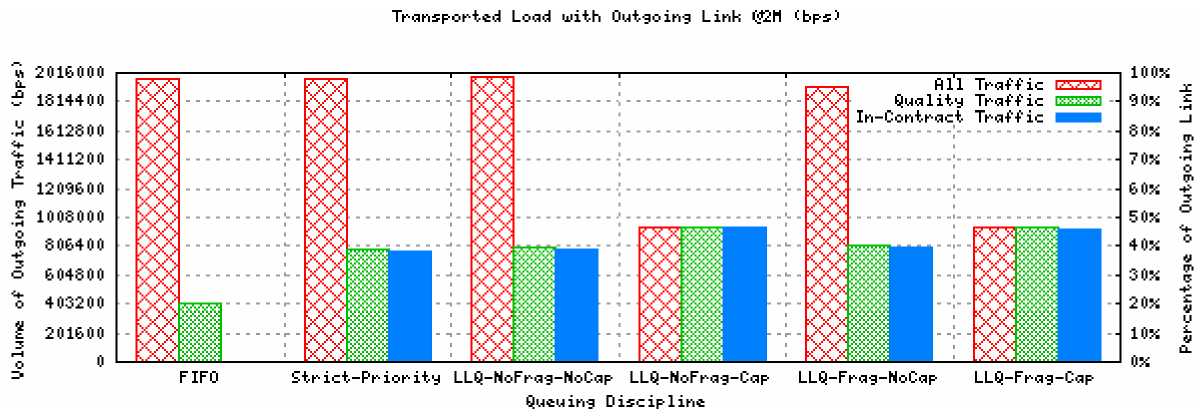


Figure 22: Transported Load at 2Mbps

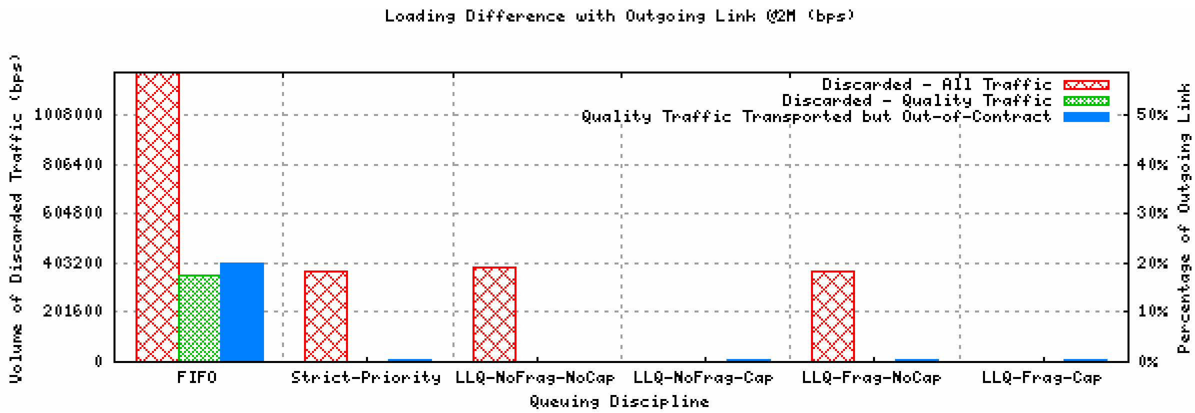


Figure 23: Loading Difference at 2Mbps

In Figure 23 the link is in saturation and all quality traffic is delivered with only a small percentage being delivered out of contract for both the Strict Priority and LLQ tests. But as the speed of the link decreases the percentage of traffic delivered in-contract decreases as shown in Figure 24.

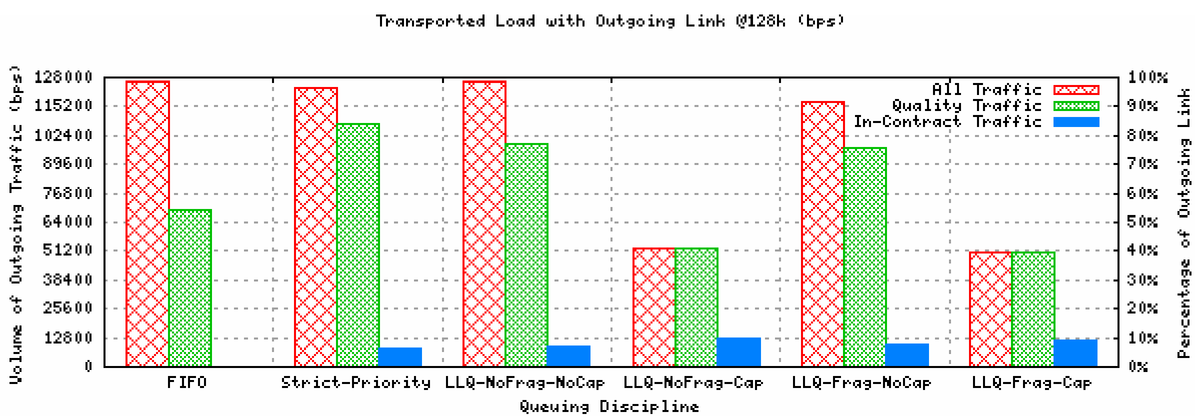


Figure 24: Transported Load at 128Kbps

The concern is that the actual in-contract link utilizations falls to the 10% range for the test case of 128K to less then 10% for the test case where the speed is changed from 256K to 128K as shown in Figure 25:

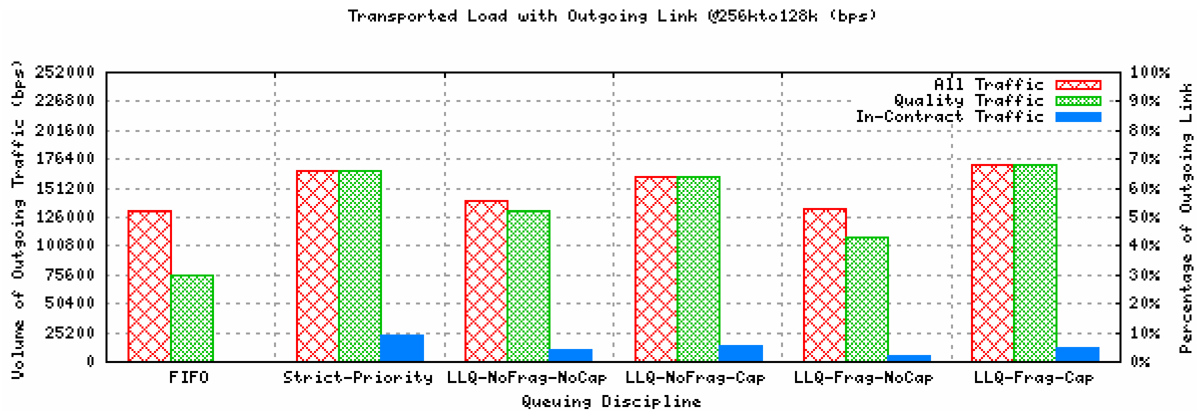


Figure 25: Transported Load at 256Kbps to 128Kbps

Even in these cases where link utilization is poor, the queuing mechanisms still protect priority and higher priority application traffic is maintained to allow for a graceful degradation as shown in Figure 26:

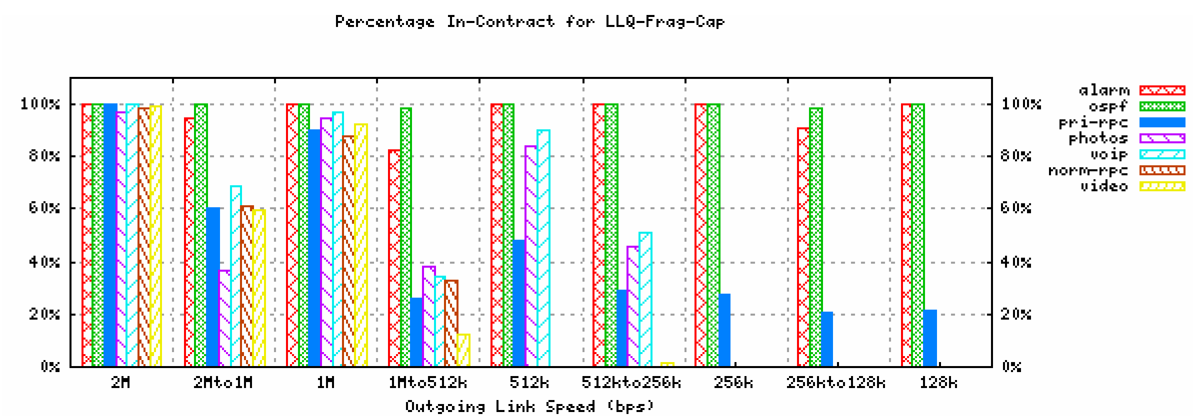


Figure 26: In-Contract for LLQ

7.2.3 Traffic Fragmentation

Though traffic fragmentation was not an explicit function to be studied in this whitepaper, during the testing it was added as a mitigation strategy to try to improve the utilization of the link and the increase of the delivery of quality traffic. Though fragmentation is a proven attribute in improving jitter especially at lower speeds, we did not see any noticeable improvement in the volume of in-contract quality traffic.

7.3 Metrics for Effectiveness

7.3.1 Predictability

Of the QoS mechanisms studied both Strict Priority and LLQ offered some degree of predictability in saturation. It is seen that the loss and delay that FIFO introduces is dependent on the overall traffic load, as expected.

There is a point, which is a function of both the offered load and the link capacity, at which delivering predictable quality degradation to a selection of flows becomes impossible, it is at this point that some applications have to experience a breach of their application quality requirements.

This can be done by policing. However, if the saturation is only short lived, permanently policing classes of traffic just in case saturation occurs does not seem an optimum strategy as utilization of network resources can be severely limited.

7.3.2 End-to-End Quality Budgets

As it can be seen from the graphs some flows were delivered well within their per-hop quality requirements for loss and delay. Is this optimum? As there is a finite capacity for delivering “quality”, there is a penalty for delivering traffic with quality that effects the other flows, delivering a quality higher than needed may not lead to the most optimal use of the link.

7.3.3 Sensitivity of QoS Queuing/Scheduling Mechanism

In performing this analysis we engaged in approximately 20 runs for each configuration. There was substantial individual variation between these runs including the loss and delay experienced, as can be seen in the spread of the data values in the various graphs. How significant this variation is with respect to the application outcome and how this variation is dependent on the particular mechanism is something for future study.

7.3.4 Order of Failure

There was not total isolation as the effect of a lower precedence application on the higher precedence applications, given that there is a desire to have an order of failure in applications is such total isolation possible? If not what are the bounds on the effect?

7.4 Observations

Without a QoS mechanism nothing can be delivered in-contract. The lack of traffic isolation means that all traffic affects all other traffic and no loss and delay requirements can be predictably met. Furthermore, there is an upper bound on the amount of offered quality load which can be successfully carried by a given mechanism which in and of itself is dependent upon the link speed. This was observed to be as low as 10% of in-contract traffic being delivered at low link speeds (i.e., 256Kbps, 128Kbps).

Even when a mechanism can provide some level of isolation between traffic classes, this was not always sufficient to assure that best effort traffic did not affect the quantity of in-contract traffic. In cases where best effort traffic was capped a higher quantity of in-contract traffic was delivered.

In most cases the applications that were in-contract were provided a better level of quality than was required. By providing better quality than was needed by some applications access to that quality was denied to other applications. It is likely that this led to more applications being out of contract than would be strictly necessary.

In highly constrained environments, such as Ad-Hoc wireless networks it is important to extract every bit of quality possible. It is clear that the emergent properties of the tested queuing mechanisms could be improved upon to maximize delivered quality and more effectively share out the Quality Degradation.

8 Follow-on Strategies

In this report we have investigated a selection of QoS queuing and scheduling mechanisms, they play one role in any network QoS management approach. Queuing and scheduling bear the main portion of the responsibility, operating at the highest frequency and the largest number of places – they operate on a packet-by-packet basis on every port of every network element. They are the means by which quality degradation is distributed amongst the current set of competing demands at that point in the network. There are other mechanisms required for an end-to-end QoS solution. Their role is to manage the offered load, either globally or at the network element, so as to keep the offered load at the network elements within the acceptable boundaries of the network elements queuing mechanisms behavior.

8.1 Additional QoS Strategies

In this report we have investigated a number of queuing mechanisms. As we have shown these mechanisms have limitations in terms of the quality that they can deliver. To deliver quality in an end-to-end network it is important to structure the system to avoid the known limitations, if they cannot be changed.

In the deployment of networks, Cisco Systems utilizes a set of tools to manage these issues, both local to a network device and continuous across the network: Classification and Marking, Congestion Avoidance, Congestion Management, Traffic Conditioning, Signaling, Link Efficiency Mechanisms, and QoS Management. In this study we focused on the queuing structure of a single network router and its effects on QoS. In the case of LLQ we also investigated how fragmentation and policing affects the results.

This provides us with an excellent baseline for investigate further the many attributes that are utilized to deliver QoS on an Ad-Hoc packet network. Some of the other attributes and tools that need to be further investigated are described in the following sections.

8.1.1 Classification & Marking

Packet classification features allow traffic to be partitioned into multiple priority levels, or classes of service. Packets can be classified in a variety of different ways, ranging from input interface, to NBAR for difficult to classify applications, to arbitrary access-control lists. You can also mark packets in a variety of ways using the policy-framework component of the Modular QoS CLI - Layer2-802.1p/Q / ISL, ATM CLP bit, Frame-Relay DE-bit, MPLS EXP bits, Layer3 IP Precedence, and DSCP bits.

8.1.2 Congestion Avoidance

The Weighted Random Early Detection (WRED) mechanism provides for congestion avoidance on network interfaces by enabling buffer management, and allowing TCP traffic to throttle back before buffers are exhausted. WRED helps avoid the loss of the last part of a packet (thus wasting the resources spent on the first part of the packet), and global synchronization issues, thereby maximizing network utilization and TCP-based application performance.

8.1.3 Congestion Management

Often a network interface is congested (even at high speeds, transient congestion is observed), and queuing techniques are necessary to ensure that critical applications get the forwarding treatment necessary. For example, real-time applications, such as VoIP and stock-trading, may need to be forwarded with the least latency and jitter (up to a provisioned limit). Cisco's LLQ provides for such a solution. Non-delay sensitive traffic, such as FTP and HTTP, can be assigned to user-defined classes within the CBWFQ queuing mechanism. The queuing techniques can be instantiated using the policy-framework of the MQC.

8.1.4 Traffic Conditioning

Traffic entering a network can be conditioned by using a policer or shaper. A policer simply enforces a rate limit, while a shaper smoothes the traffic flow to a specified rate by the use of buffers. Once again, mechanisms such as Class-based Policing, Class-based Traffic Shaping, and Class-based Frame Relay Traffic Shaping (CBFRTS) can be configured within the MQC framework.

8.1.5 Signaling

Cisco IOS Software, in addition to supporting provisioned QoS (including the IETF DiffServ Model with techniques such as Policing, Traffic Shaping and Layer 3 packet marking), also provides for the Integrated Services (IETF-IntServ) model, where resources are actually reserved ahead of time along the entire path of a flow. Resource Reservation Protocol (RSVP) is the primary mechanism to perform Admission Control for flows in a network. A perfect example is in the case of VoIP (Voice over IP), where a call is completed only if the resources are available for it, ensuring that a call coming into a network does not bump or affect the quality of existing calls. Another technique called QPPB (QoS Policy Propagation via BGP) allows for indirectly signaling (using the community-list attribute in BGP) the forwarding priority for packets destined toward an autonomous system, AS path, or IP prefix. This is a highly useful feature for service providers and large enterprises.

8.1.6 Link Efficiency Mechanisms

Streaming video and voice traffic uses the Real-Time Protocol (RTP). IP, UDP, and RTP packet headers can be compressed from approximately 40 bytes down to 5 to 8 bytes. This saves a tremendous amount of bandwidth in the case of low-speed links, and when supporting a large number of media streams. In addition, FRF.12 (Frame-Relay Forum specification for frame-fragmentation) and Cisco LFI (Link Fragmentation & Interleaving) allow for fragmenting large data packets, interleaving them with RTP packets, and maintaining low delay and jitter for media streams.

8.1.7 QoS Management

Includes the monitoring and learning functions required to monitor, understand and optimize current network behavior, as well as configuration and provisioning functions. AutoQoS, QoS Policy Manager (QPM), Class-based QoS Management Information Base (CBQ0SMIB) and NBAR Protocol-Discovery MIB are important examples of QoS management tools.

8.2 MANET

MANET (Mobile Ad-Hoc Networking) is a set of IP based features that allow routers to establish an IPv6 based mobile Ad-Hoc peer-to-peer network with no pre-existing or defined infrastructure. The solution consists of multiple phases including the first with OSPFv3 routing protocol updates, radio to router data link management for real time radio metrics, and IPv4 support over the IPv6 backbone. Subsequent phases of MANET will focus on scalability and performance.

Some advantages of MANET include:

- Minimizing both the size and number of OSPFv3 routing updates.
- Providing real time radio link information will allow the rapid establishment of the ‘best’ route in a MANET environment.

9 Conclusions

IP Ad-Hoc networks provide many unique challenges over today’s traditional wired networks. One such characteristic of a mobile Ad-Hoc network is the changing of speed on the physical link without notice. This possible change of link capacity brings new stresses on a network element to maintain QoS. Other key attribute is that in most Ad-Hoc networks today, the ability to add bandwidth is not as easy an option as in today’s wired networks. This makes it even more difficult to maintain QoS.

In this report we have focused on documenting emergent properties of queuing mechanisms found within a commercial network device. The queuing mechanism and its associated algorithms are the foundation to maintaining the QoS required during times of high demand the network.

For the purpose of this study, the team chose to evaluate three industry-accepted queuing mechanisms: First-In First-Out (FIFO), Strict Priority Queuing and Low Latency Queuing (LLQ). Empirical data was collected through a test bed built by the team. This test bed included routers (including the unit under test), switches and workstations. The workstations housed the simulated traffic generators and data collectors. The offered load from the generators was such that the egress link would be fully saturated, so as to expose the emergent properties of the QoS mechanism under test under these conditions.

From this work the following conclusions on how the emergent properties of queuing affect QoS can be made:

- FIFO, as a queuing mechanism, does not maintain the necessary QoS to meet the needs as defined by the application's quality requirements for this study. Most applications were not delivered sufficient quality at saturation and beyond.
  - Advanced queuing, mechanisms like Strict Priority and Low Latency Queuing are necessary to deliver traffic with quality across a network element that is running near, at or beyond saturation.
- Understanding the emergent properties of queuing mechanisms provides a foundation for predicting the ability of a network element to deliver QoS. At some point all queuing mechanisms will be unable to deliver the desired QoS to all the competing traffic flows. The physics defines an upper limit to how much quality can be delivered.
  - The emergent properties of each queuing mechanism can be generalized but every traffic model (combination of application requirements and volumes of their traffic) will need to be analyzed for their potential "trades" so as to determine the best configuration.
- The total amount of available quality is finite. The amount of available quality becomes more limited as the egress capacity is reduced.
  - Advanced queuing mechanisms provide the ability for the network element to maintain application traffic priority as egress link speeds are reduced. This is important as it provides for the basis of graceful degradation.
  - While Strict Priority does deliver a priority failure order, LLQ takes it one step further by delivering a larger percentage of the higher priority traffic as the amount of total available quality becomes less.
  - While queuing mechanisms like LLQ do deliver a level of isolation, best effort and lower priority traffic can negatively influence the amount of quality traffic delivered by the network element. Policing, the preemptive discarding of best effort and lower priority traffic reduces this negative influence, increasing the ability to deliver quality to the higher priority traffic.
- For advanced queuing mechanisms like LLQ, to provide optimal performance the configuration on the network element needs to be tuned when physical link speeds are changed. LLQ will still operate if not re-configured but will not produce optimal results.
  - Depending on the desired result, LLQ can be configured to strictly protect higher priority traffic or not to starve a lower priority application.
  - Queuing mechanisms like Strict Priority have a fixed configuration which brings less complexity to the picture.

- Understanding the emergent properties of the queuing mechanisms provides a base foundation for being able to predict the ability of the network to deliver application traffic within its quality requirements.
  - Queuing mechanisms are just one part of the QoS management approach and other QoS mechanisms need to be considered to come up with an optimal solution. The evaluation of such mechanisms is out of the scope of this paper.

This study has proven a need for advanced queuing in bandwidth restricted network like the ones found in mobile Ad-Hoc networks. These queuing mechanisms do provide the ability for a network device to not only deliver traffic in a predictable fashion but also execute a graceful degradation of service if called on.

There is a difference between optimizing the network element QoS mechanism and optimizing the end-to-end network to support applications and their outcomes. The strictures of the Ad-Hoc wireless environment will drive network elements into saturation. Even though other QoS mechanisms are deployed and will manage the overall end-to-end system, the saturation at a network element will persist long enough to affect the user visible outcome of applications. It is during these periods that the ability of the network element's QoS mechanism will be critically required.

In this study we have taken a mixture of application traffic and studied their interactions, while measuring the delivered quality degradation. We have highlighted the boundaries of achieving acceptable application outcomes in a network environment which is in saturation. We have seen that there is a single trading space – the total "quality" available, and examined the effects of individual queuing mechanisms in partitioning the "quality" available at a network element amongst the set of competing applications. This has been done in terms of loss and delay as well as the more commonly used yardstick of bandwidth. The fraction of the egress link that could be delivered within the quality constraints varied from as low as around 10% (at 128kbit/sec) to around 55% (at 1Mbit/sec), these figures being a combination of the emergent properties of the queuing mechanism within the constraints imposed by the physics of the system.

Future work is needed to improve the quantity of quality that can be delivered, both at a single network element as well as across the multiple elements that make up the end-to-end network.

Appendix A Composite Test Results

This appendix contains the composite results for the tests run. All the speeds and combinations are presented in section A.1 Fixed Speed Link Test Results, however in section A.2 Mixed Speed Link Test Results only the graphs corresponding to the transient conditions are reproduced here. The steady-state graphs just duplicate the fixed speed test results

A.1 Fixed Speed Link Test Results

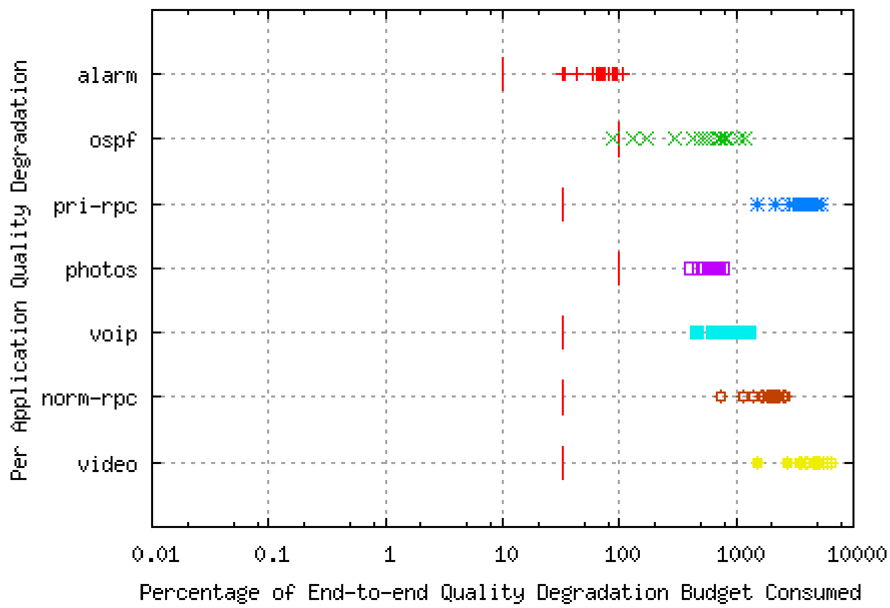
These tests were performed as described in section 6.1, with the egress capacity remaining the same for the duration of the test.

A.1.1 Consumed DQ Scatter Plots

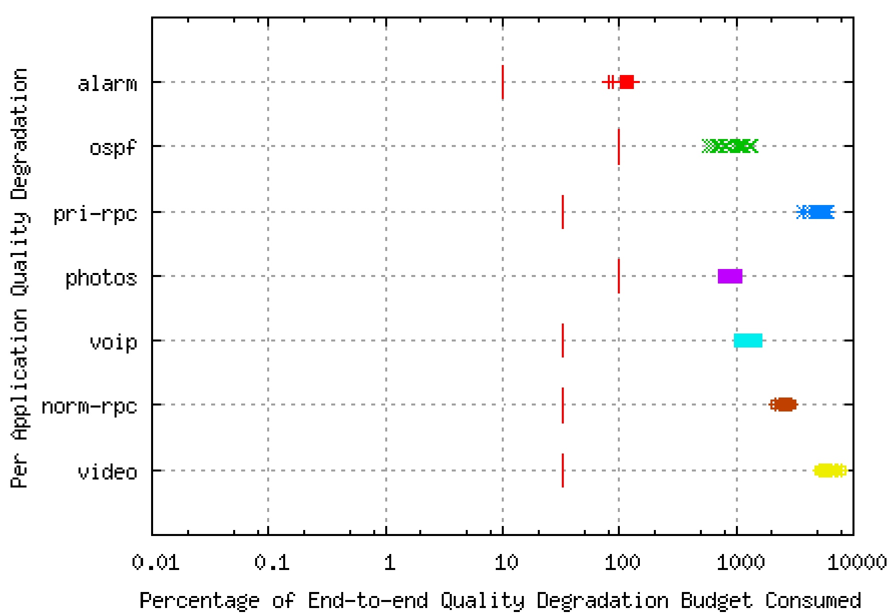
These plots represent, for a given queuing discipline at a given speed, the amount of end-to-end budget (in terms of both loss and delay) that an application has consumed. The vertical red bar represents the per-hop budget, with 100% representing the end-to-end budget. A mark to the left of the vertical red bar indicates a run which did not consume more than the per-hop budget. The abscissa is on a logarithmic scale. If all the points lay exactly on the vertical red bar this would represent “perfection” – all applications received exactly the quality they needed.

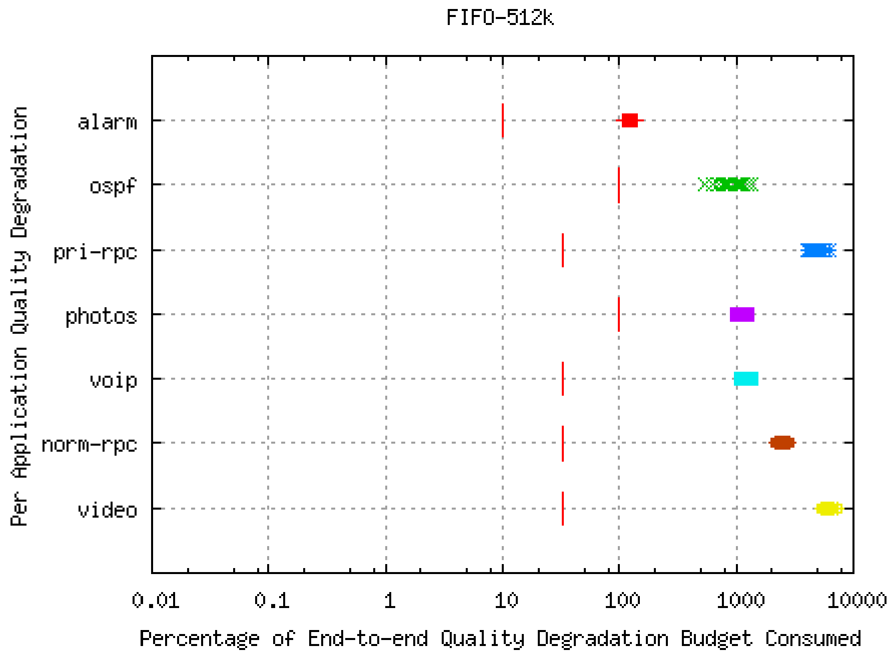
A.1.1.1 First-In, First-Out (FIFO)

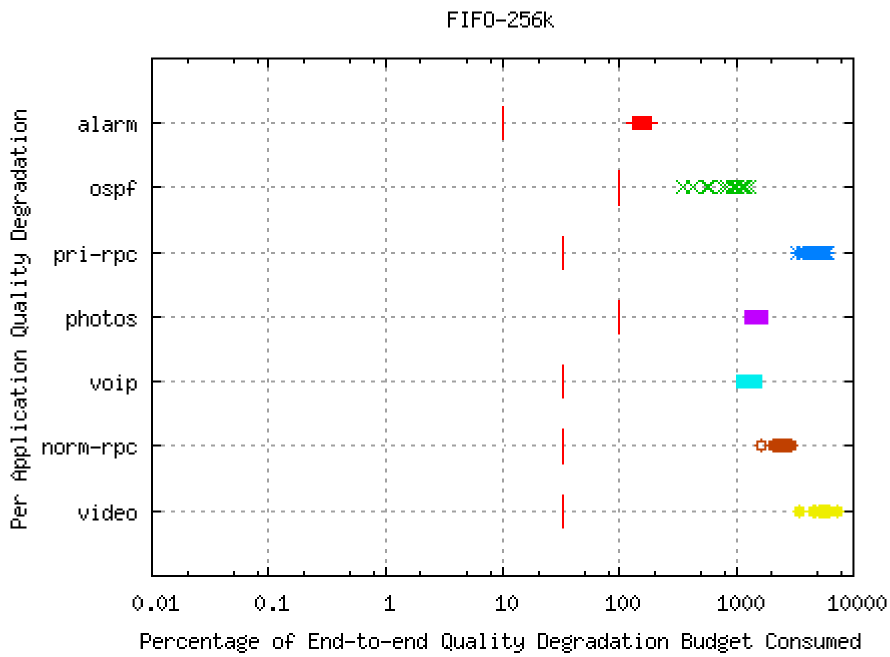
FIFO-2M

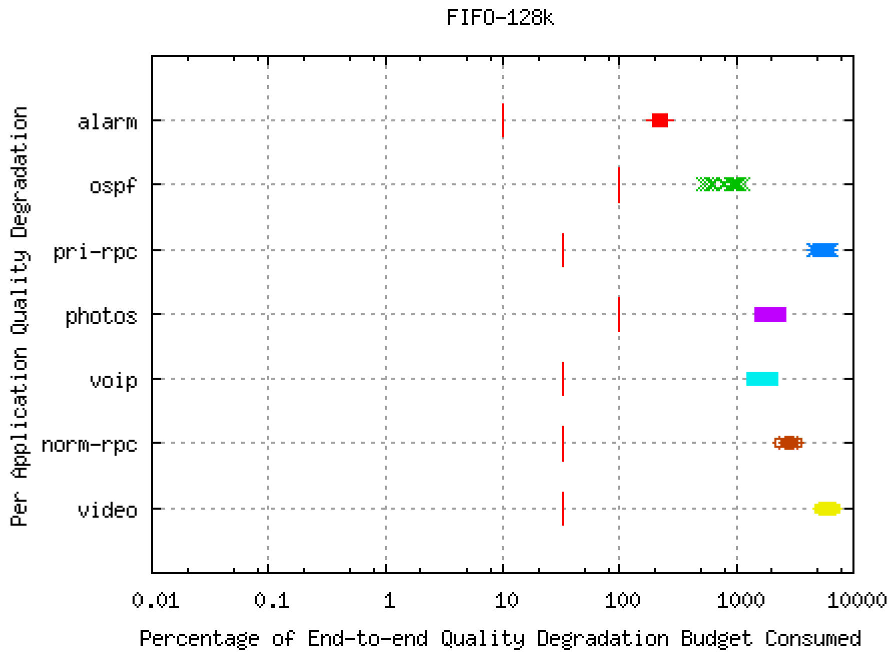


FIFO-1M

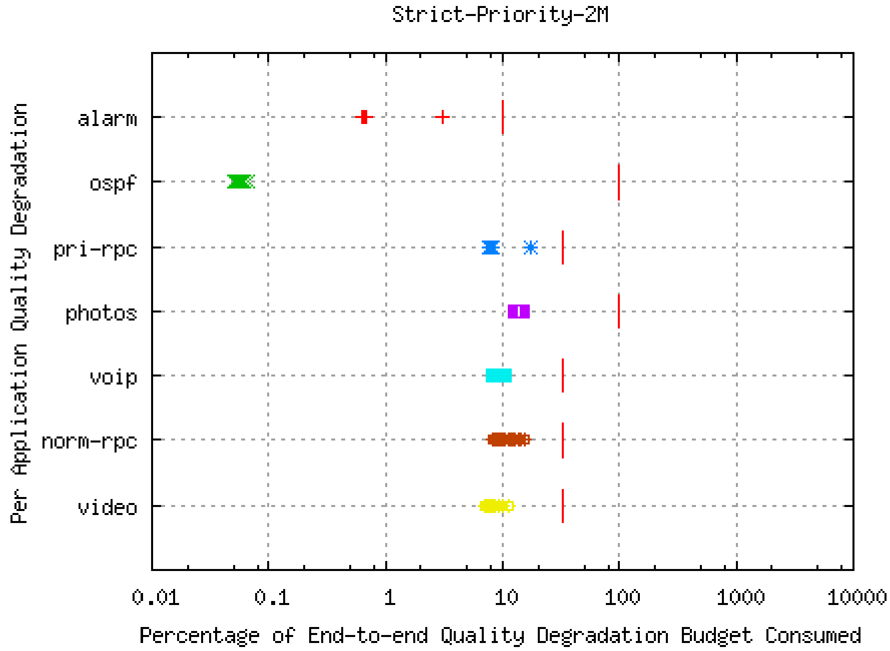


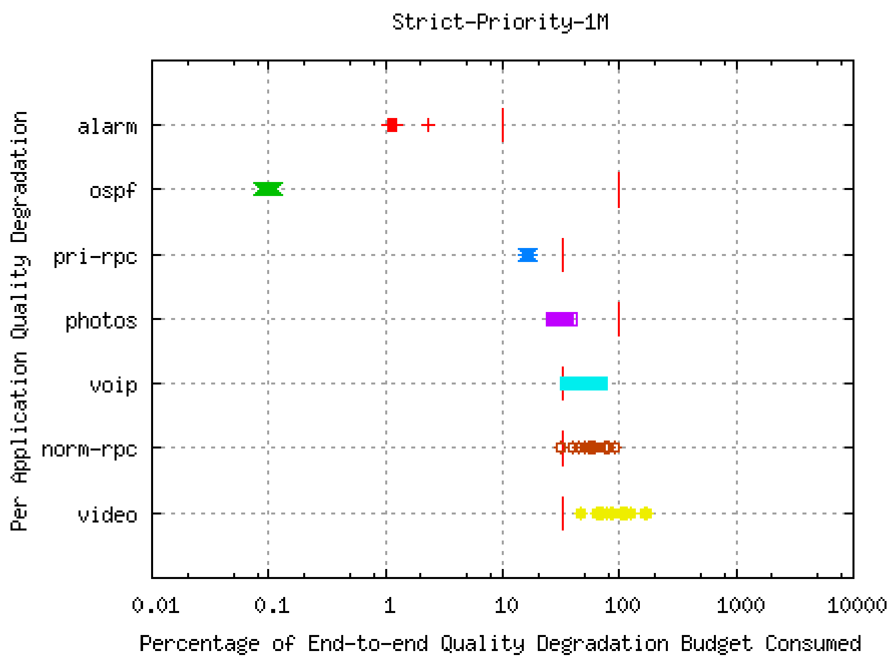

FIFO-512k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

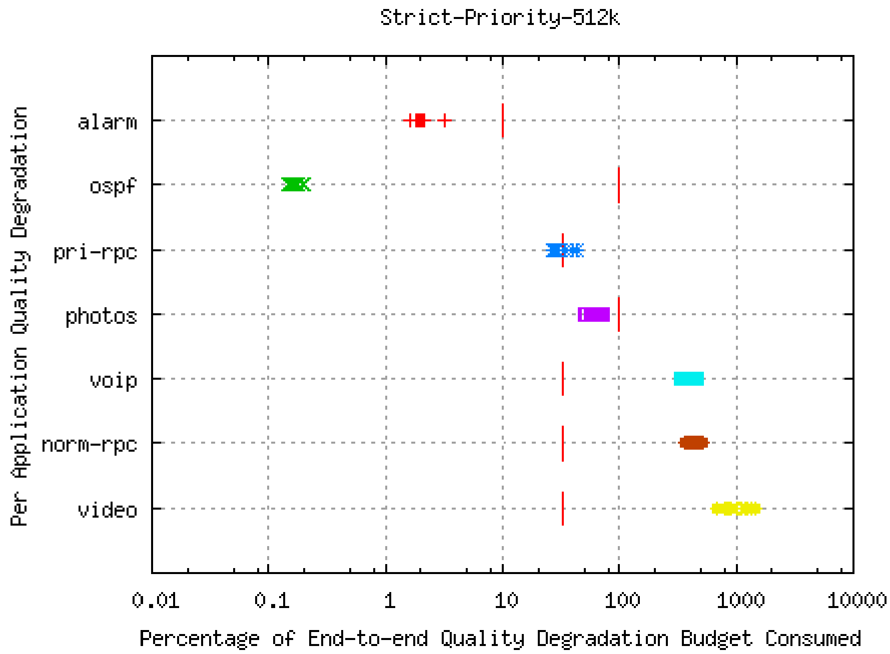

FIFO-256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

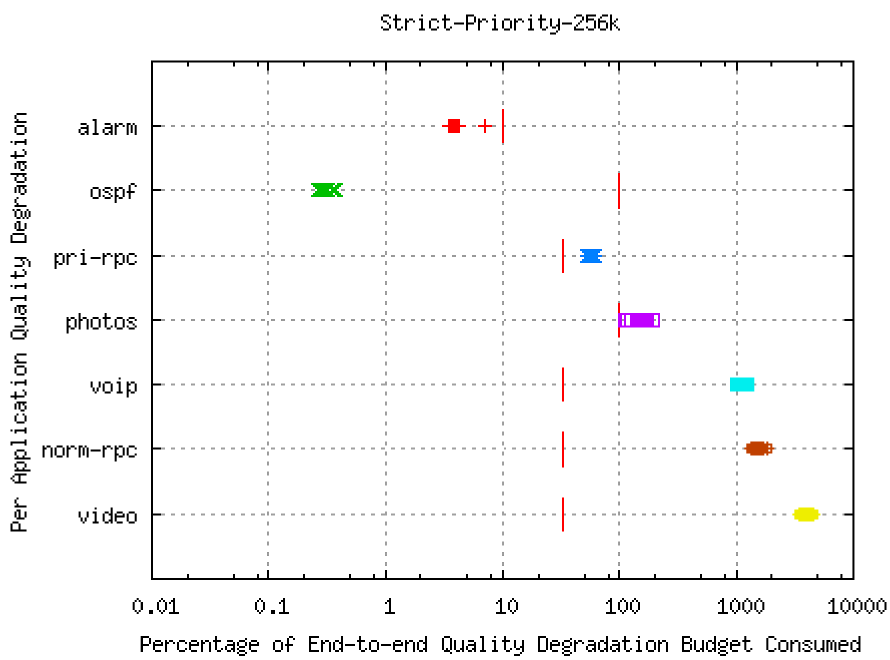

FIFO-128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

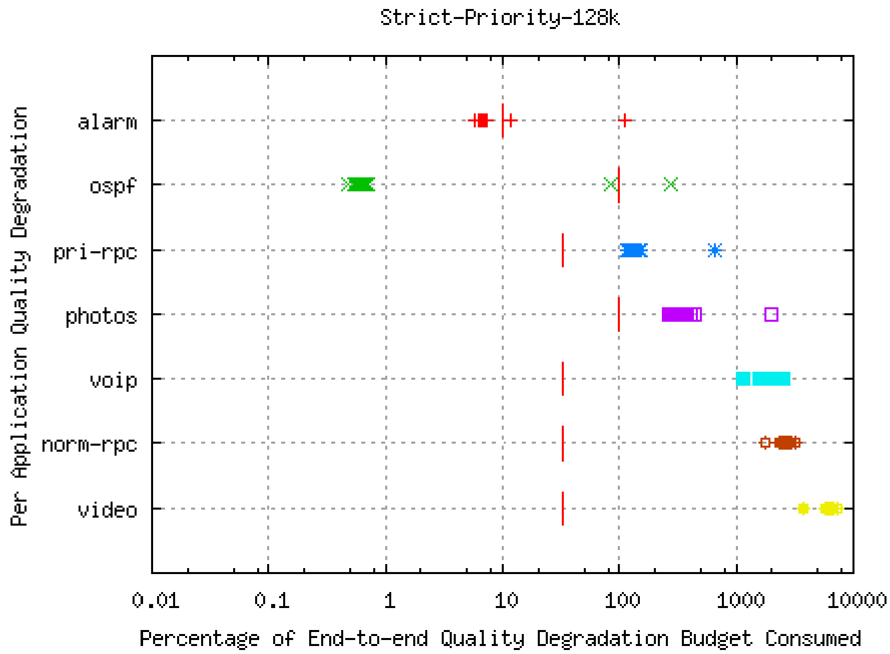
A.1.1.2 Strict Priority Queuing



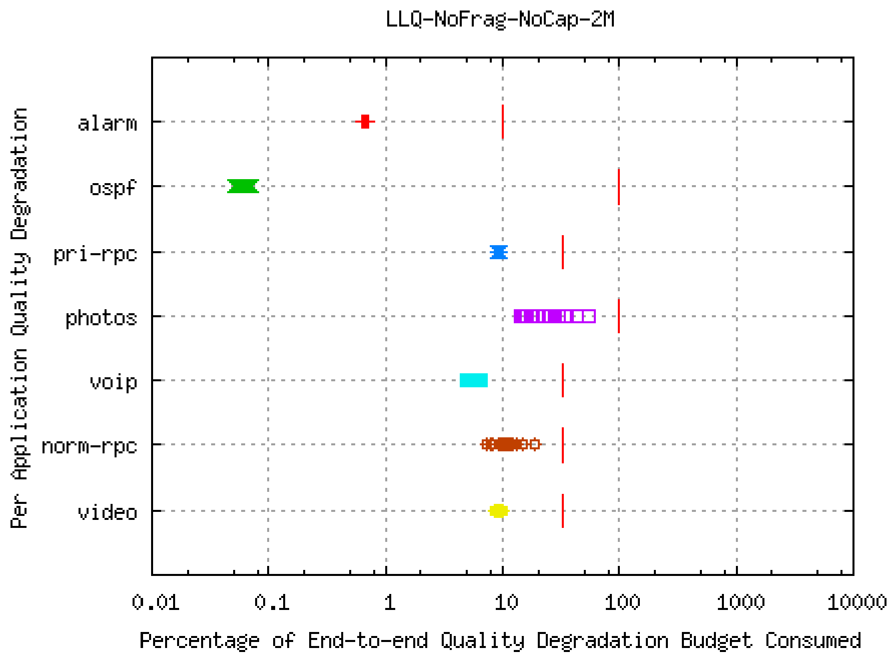


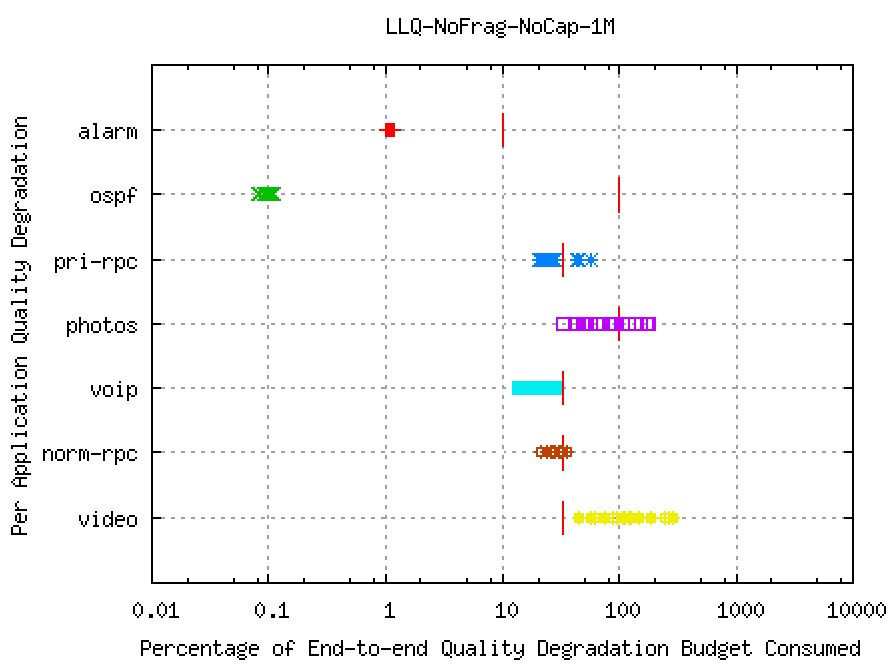

Strict-Priority-512k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

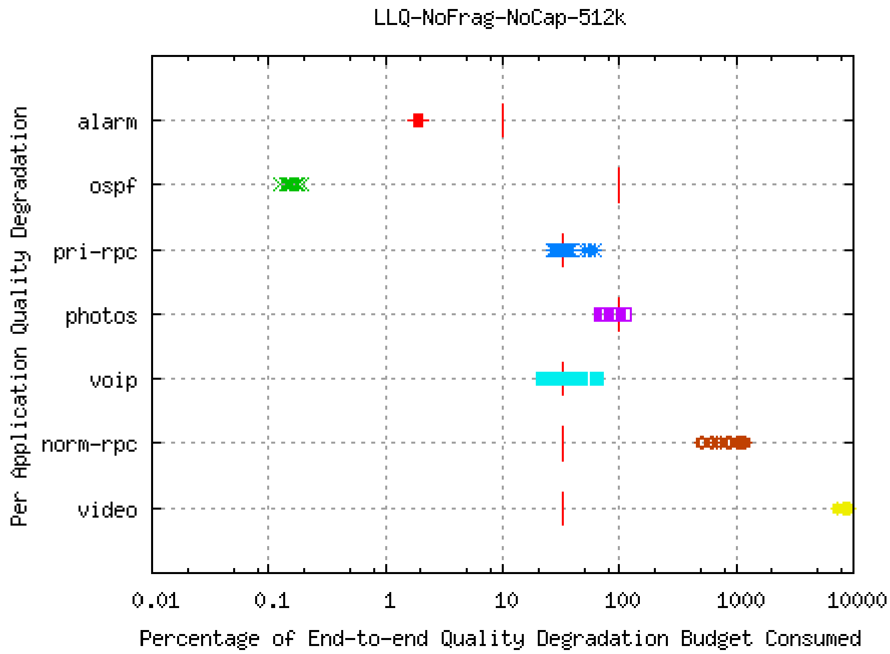

Strict-Priority-256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed


Strict-Priority-128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

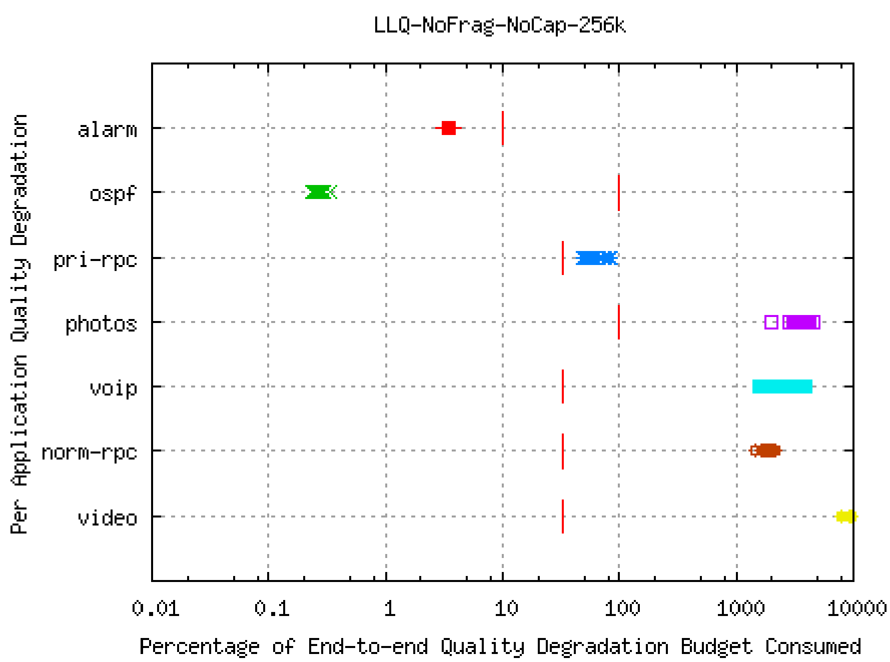
A.1.1.3 Low Latency Queuing (*no mitigation)*



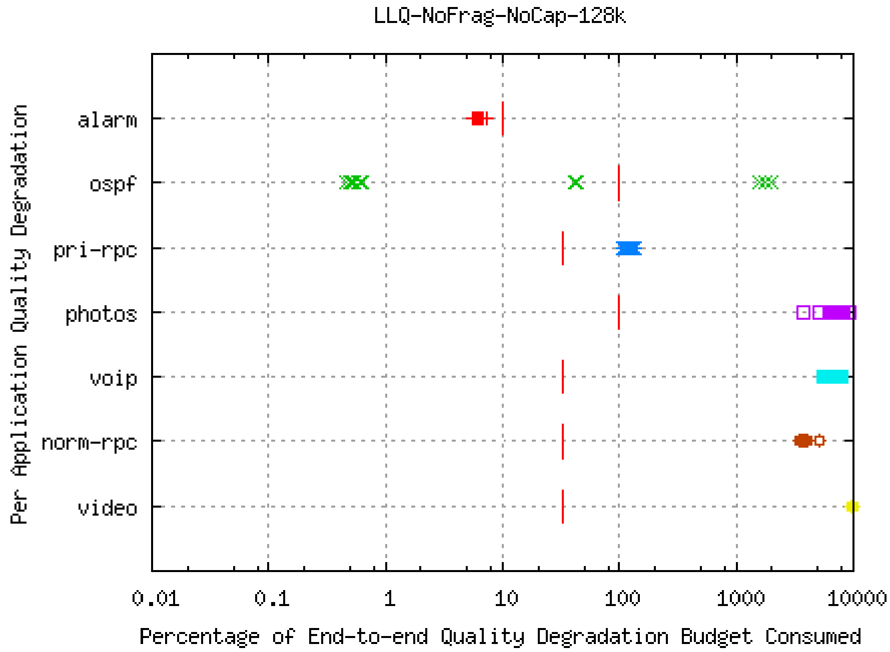




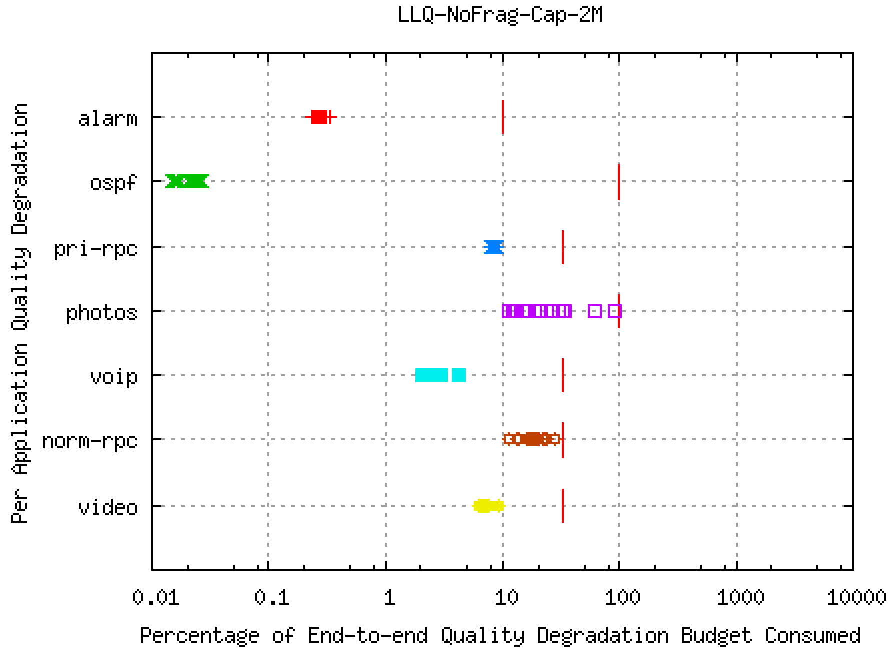
LLQ-NoFrag-NoCap-512k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

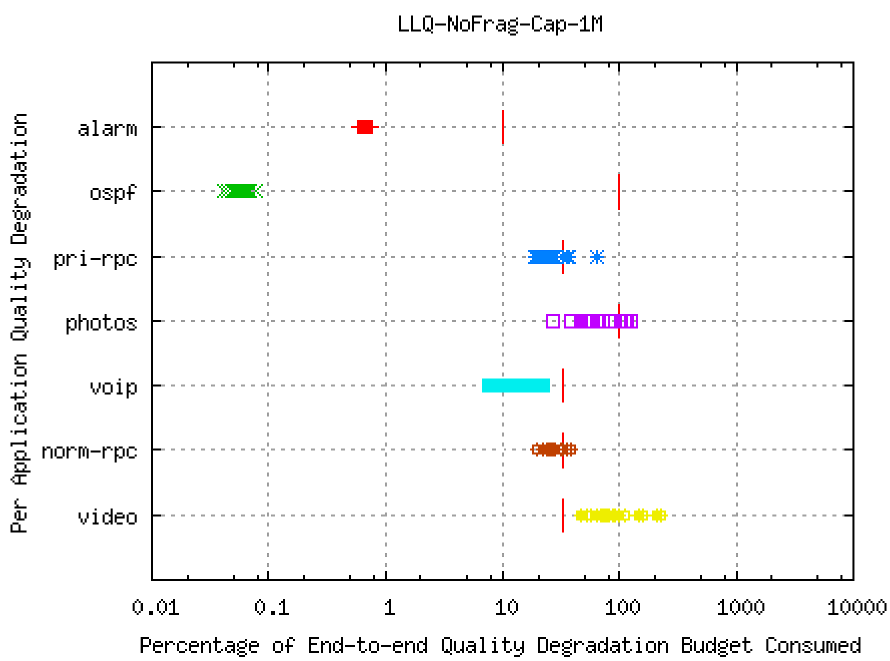


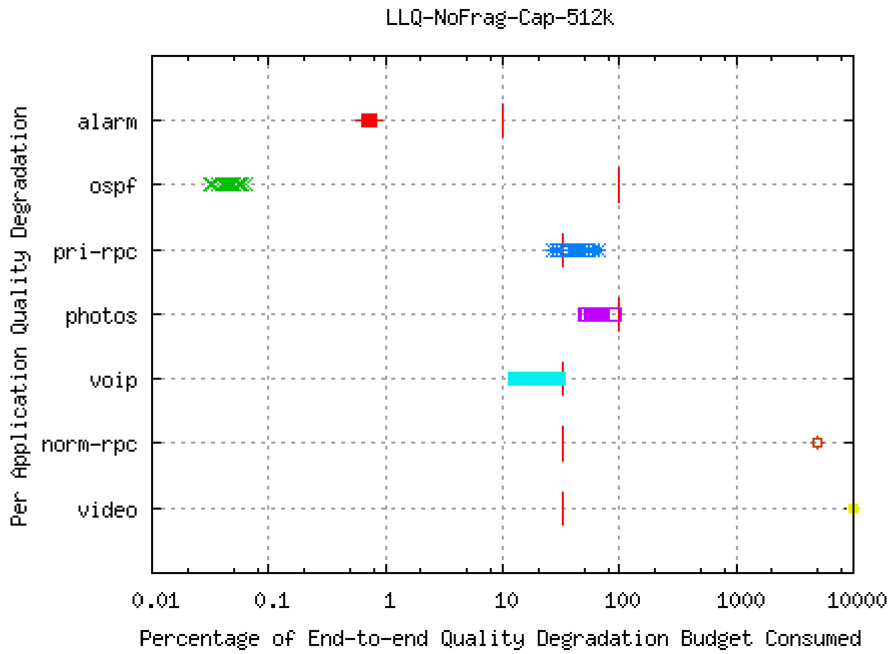
LLQ-NoFrag-NoCap-256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

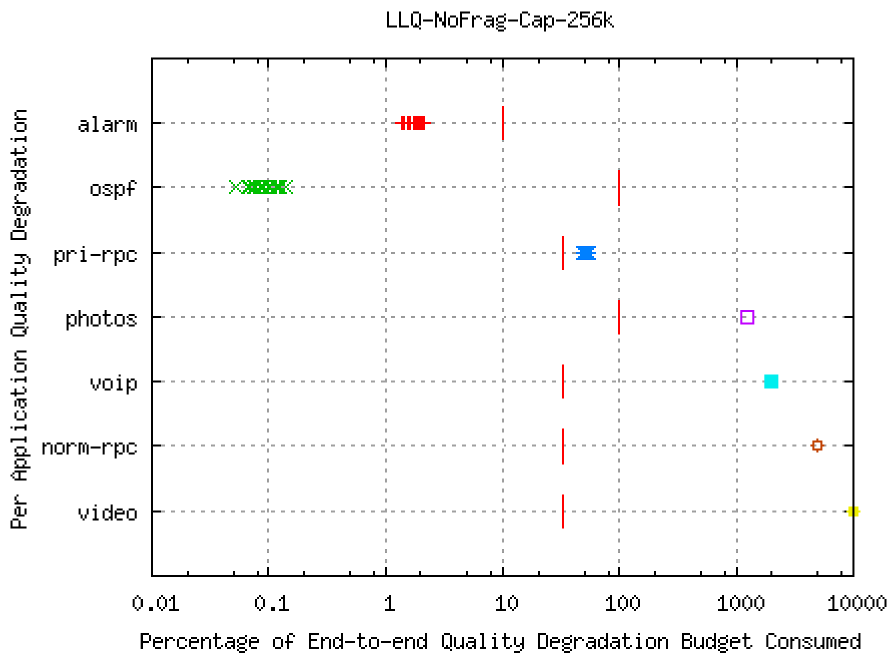

LLQ-NoFrag-NoCap-128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

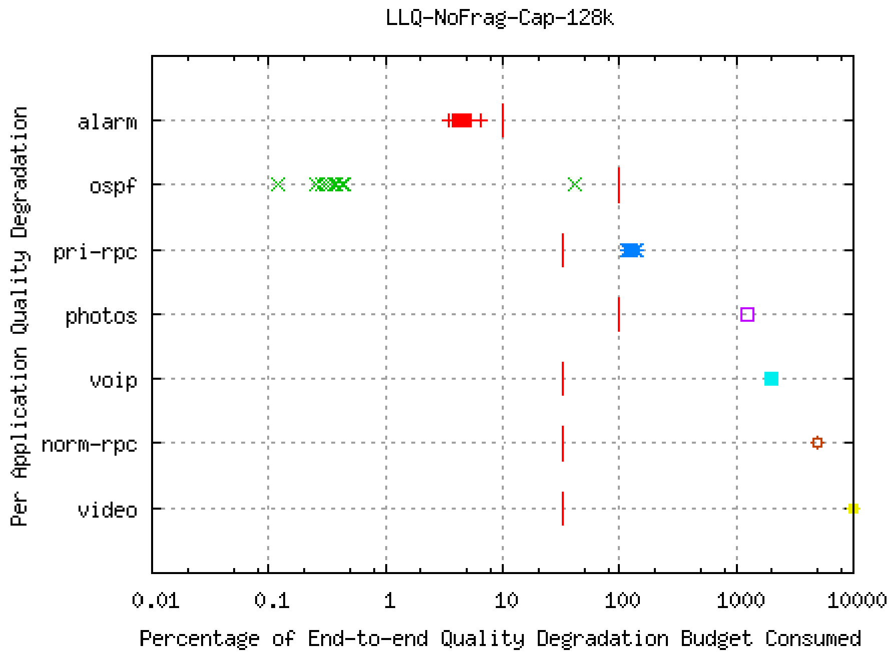
A.1.1.4 Low Latency Queuing (with selected Traffic Capped)



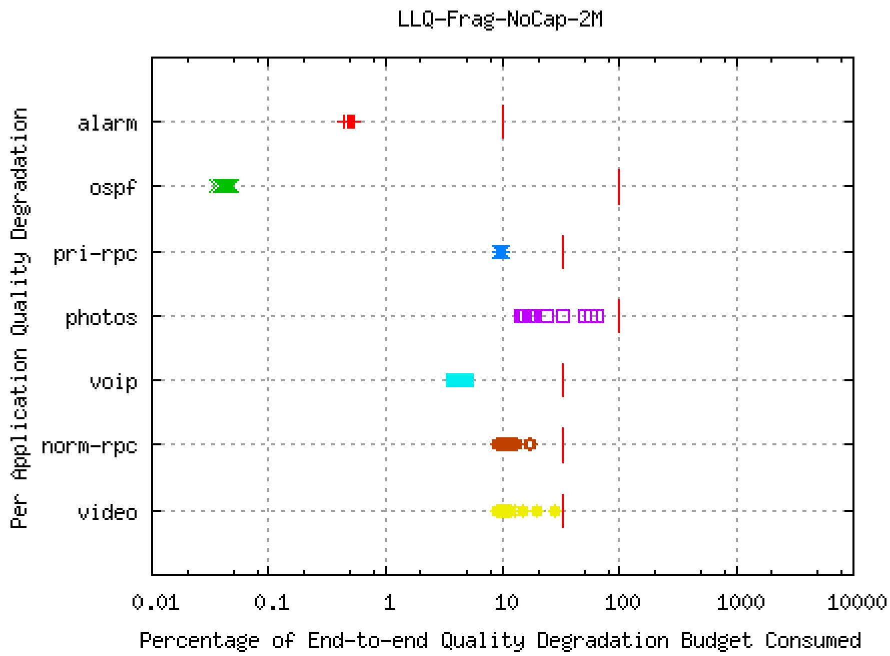


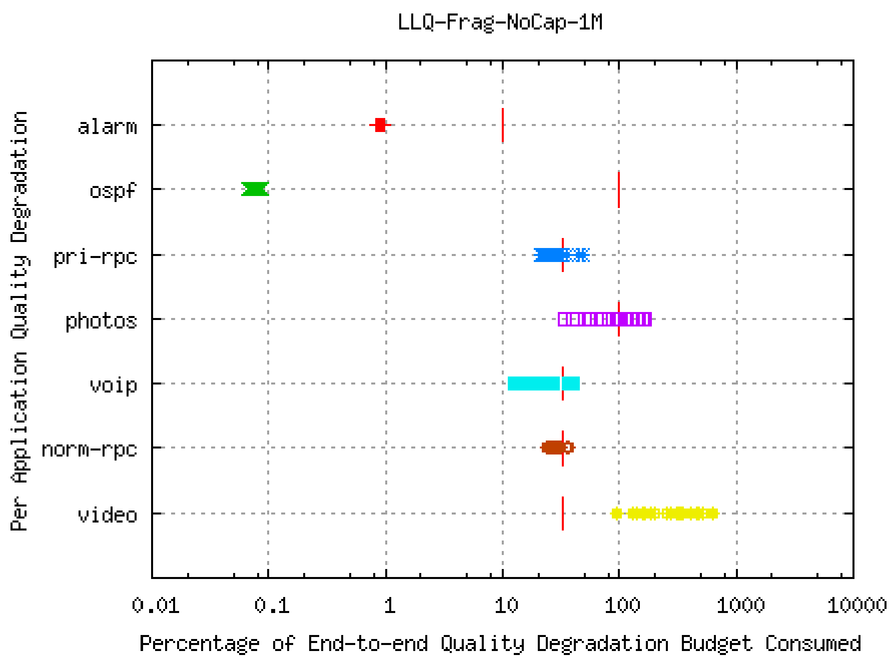

LLQ-NoFrag-Cap-512k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

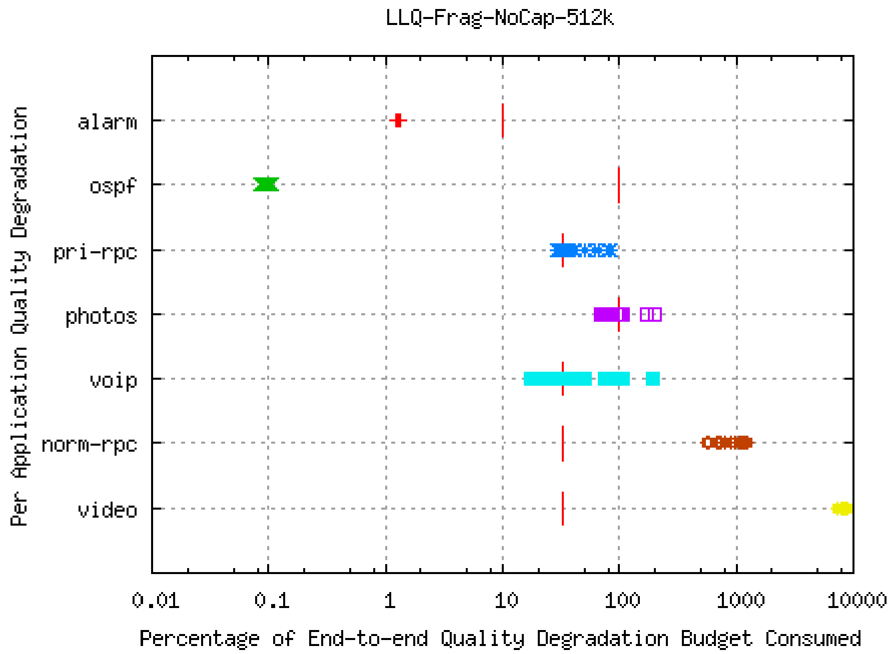

LLQ-NoFrag-Cap-256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed


LLQ-NoFrag-Cap-128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

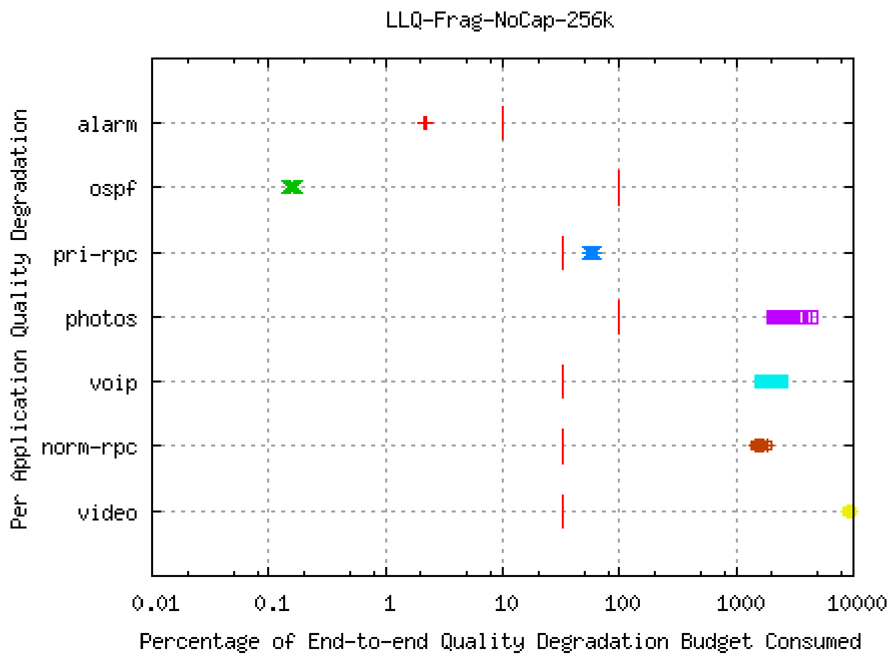
A.1.1.5 Low Latency Queuing (with Packet Fragmentation)



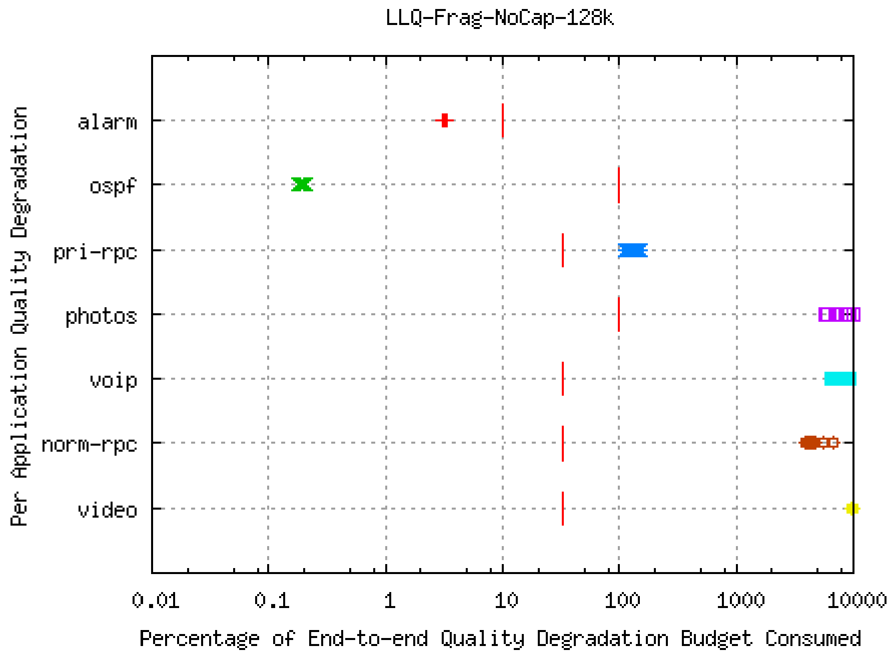




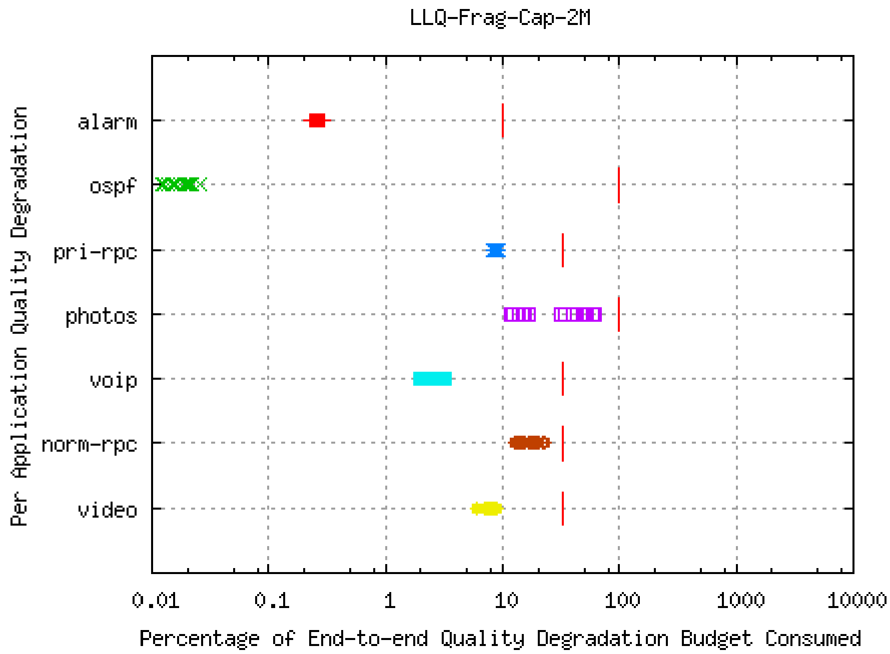
LLQ-Frag-NoCap-512k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

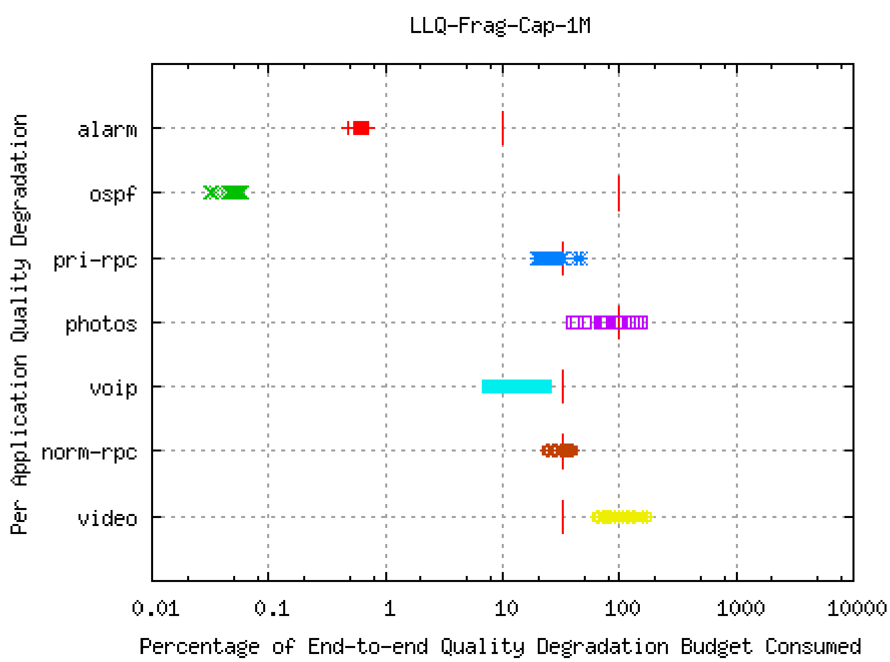


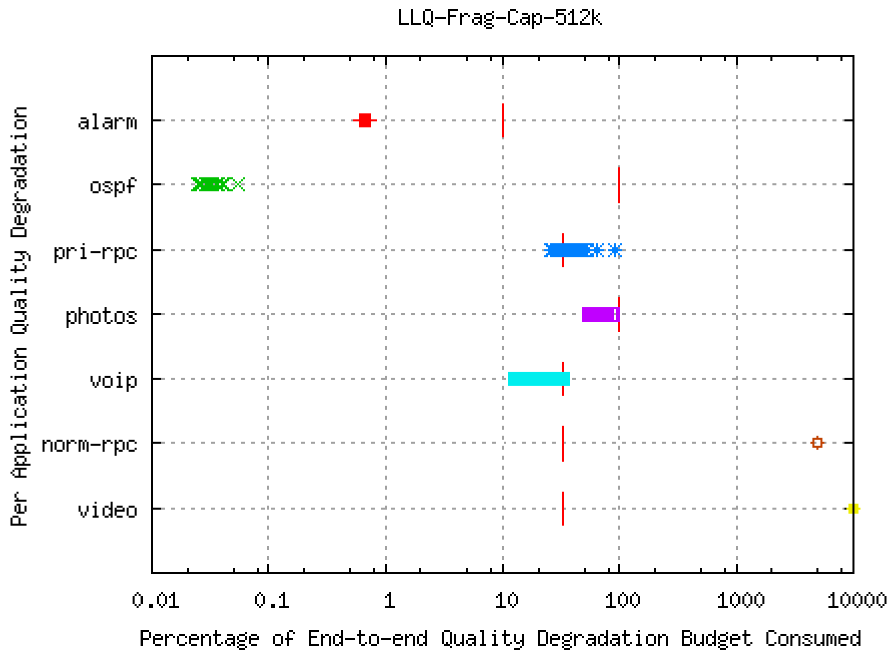
LLQ-Frag-NoCap-256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

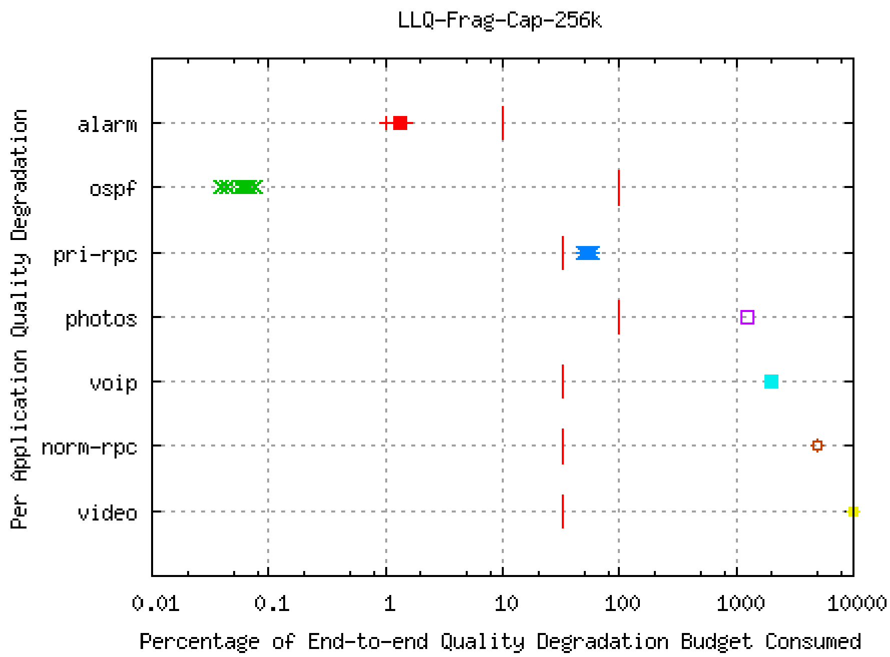

LLQ-Frag-NoCap-128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

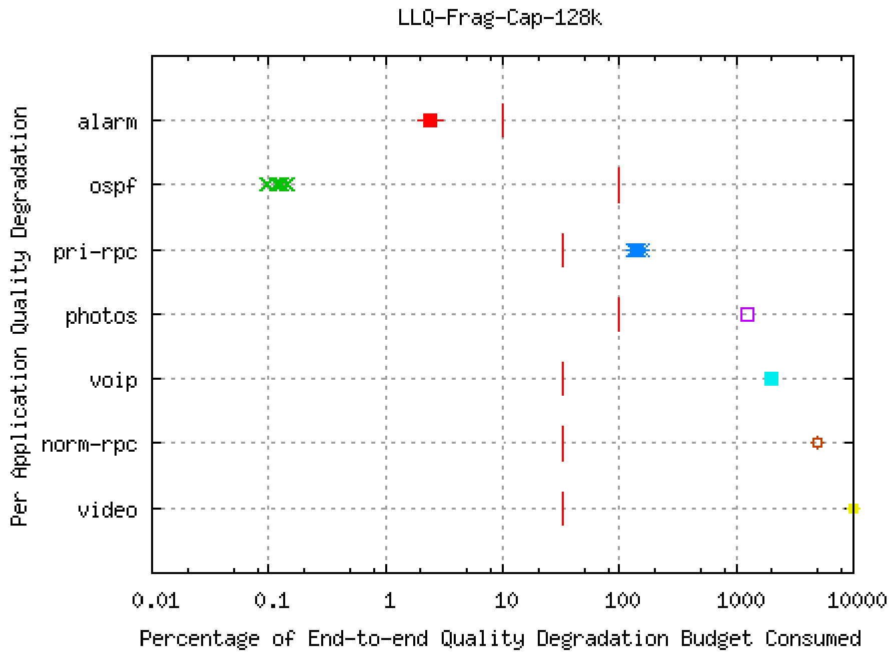
A.1.1.6 Low Latency Queuing (with Packet Fragmentation and selected Traffic Capped)






LLQ-Frag-Cap-512k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed


LLQ-Frag-Cap-256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed


LLQ-Frag-Cap-128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

A.1.2 Link Utilization Percentages

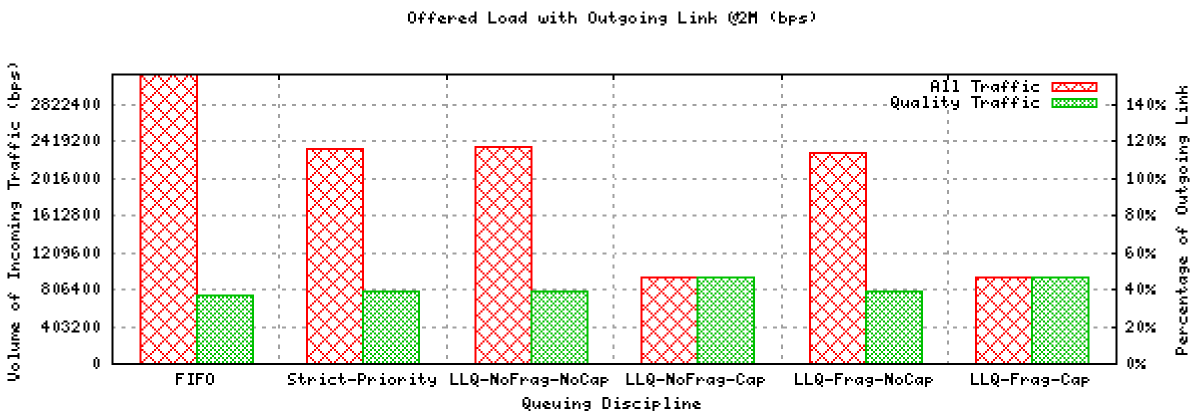
The Link Utilization Percentages are split into three graphs; *offered load*, *transported load* and *load difference*, again there is a graph for each link speed and each queuing mechanism variation. All three types of graphs have the application of interest on the x-axis, and two y-axes. The left y-axis is the absolute volume of traffic in bits per second; the right y-axis measures the fraction of the outgoing link that that volume represents at that speed.

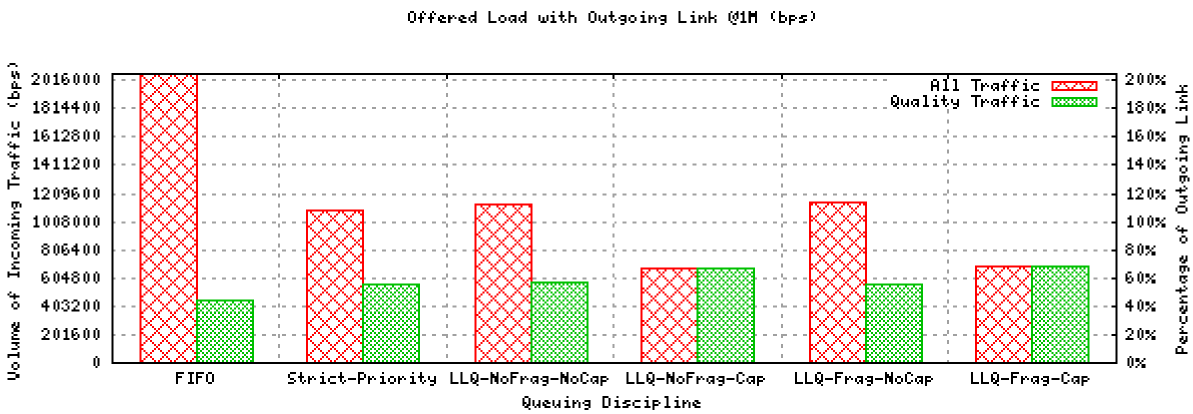
The *Offered Load* graph shows you the amount of traffic that was offered to the device under test. The two bars per queuing discipline represent the total amount of offered traffic, and the total amount of traffic offered that required quality guarantees (the "quality" traffic).

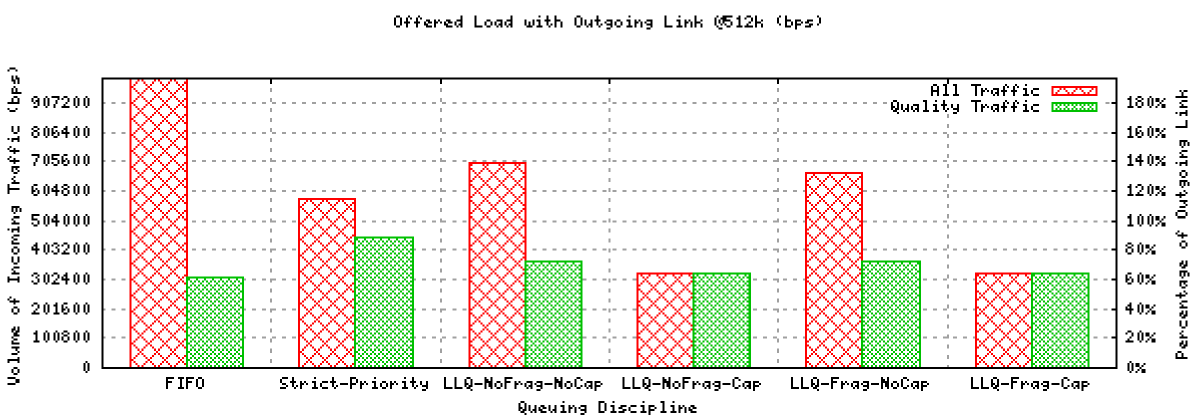
The *Transported Load* graph has three bars per queuing discipline. The fist bar *All Traffic* indicates the total amount of traffic that was transported (departed through the egress link). The second bar shows that amount of quality traffic transported. The third bar represents the amount of quality traffic that was transported in-contract; that is traversed the network element within the appropriate quality constraints. The first bar can be thought of as the total amount of “work” done, the second as the amount work done for traffic with a quality constraint (irrespective of delivered quality), and the third as the amount of useful work done on behalf of the quality traffic (delivered within the respective quality constraints).

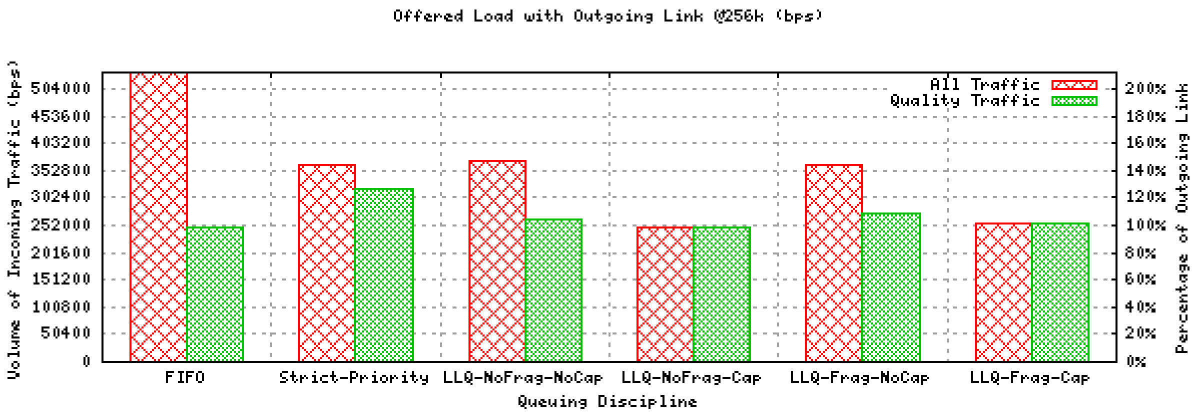
The final graph *Load Difference* has three bars per queuing discipline. The first bar *Discarded - All traffic* represents the difference between the total load offered and the total load transported. For example if the network element was offered 2.2M, and the egress link was configured at 1M and fully utilized, then this bar will be 1.2M (the total volume of traffic discarded). The second bar *Discarded - Quality Traffic* is the difference between the offered and transported load for just the quality traffic. The third bar is a little more interesting. It represents the amount of quality traffic that was transported but was “out of contract” (work done that may or may not be of use to the applications). This is the difference between the *Quality Traffic* and *In-contract Traffic* bars as shown on the *Transported Load* graph.

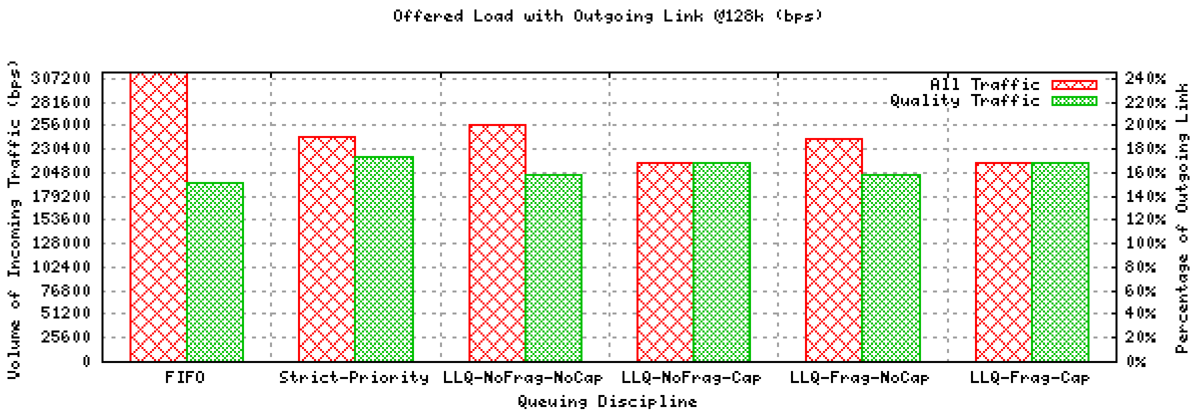
A.1.2.1 Offered Load



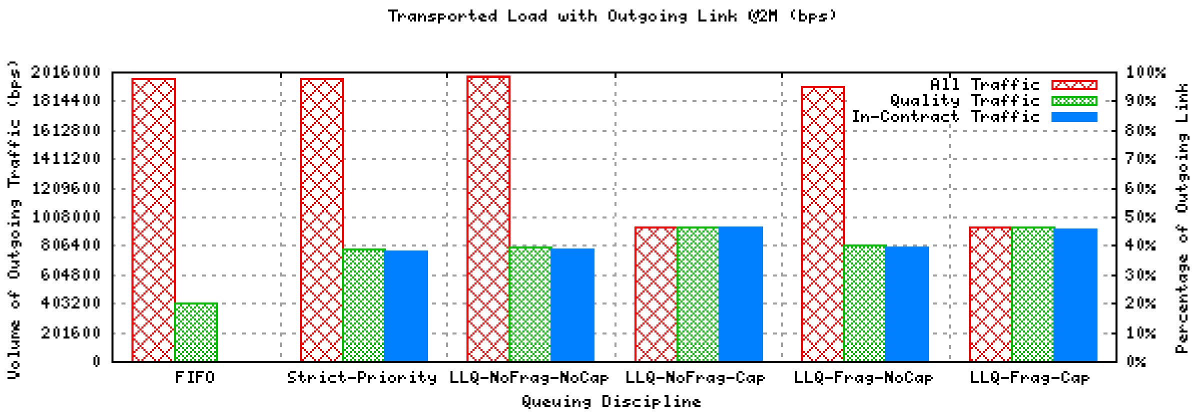


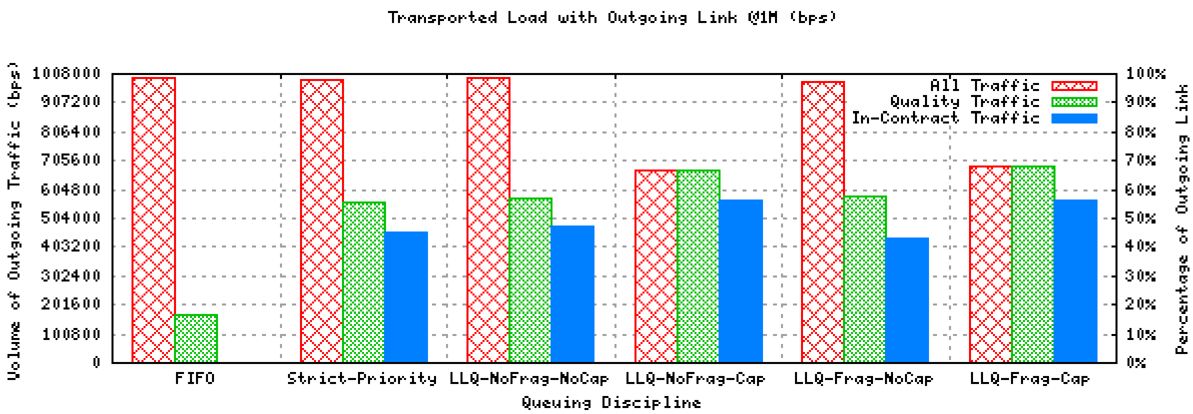


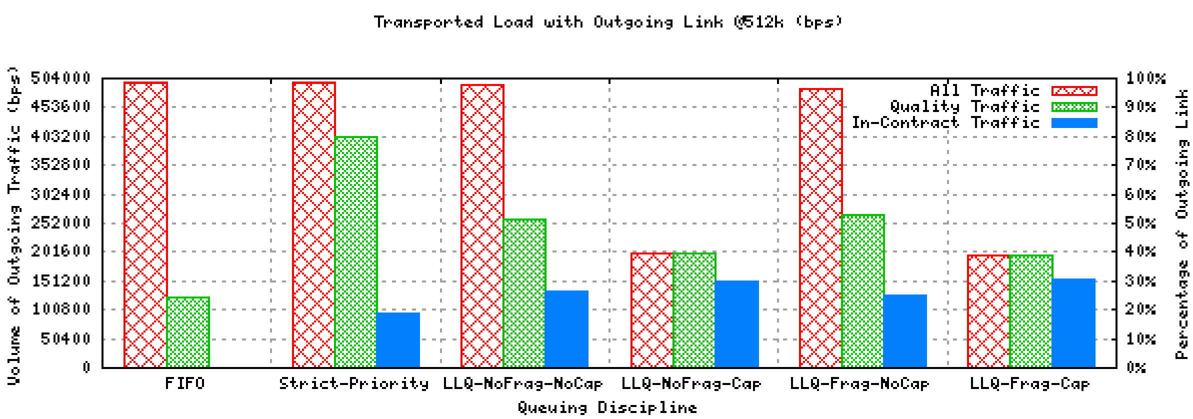

Offered Load with Outgoing Link @256k (bps)
Volume of Incoming Traffic (bps)
Percentage of Outgoing Link
All Traffic
Quality Traffic
504000
453600
403200
352800
302400
252000
201600
151200
100800
50400
0
200%
180%
160%
140%
120%
100%
80%
60%
40%
20%
0%
FIFO
Strict-Priority
LLQ-NoFrag-NoCap
LLQ-NoFrag-Cap
LLQ-Frag-NoCap
LLQ-Frag-Cap
Queuing Discipline

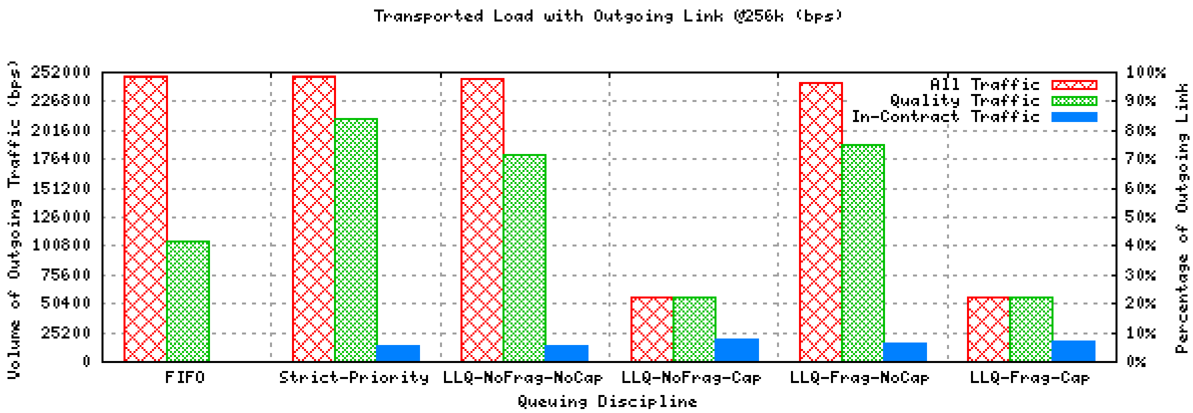

Offered Load with Outgoing Link @128k (bps)
Volume of Incoming Traffic (bps)
Percentage of Outgoing Link
All Traffic
Quality Traffic
307200
281600
256000
230400
204800
179200
153600
128000
102400
76800
51200
25600
0
240%
220%
200%
180%
160%
140%
120%
100%
80%
60%
40%
20%
0%
FIFO
Strict-Priority
LLQ-NoFrag-NoCap
LLQ-NoFrag-Cap
LLQ-Frag-NoCap
LLQ-Frag-Cap
Queuing Discipline

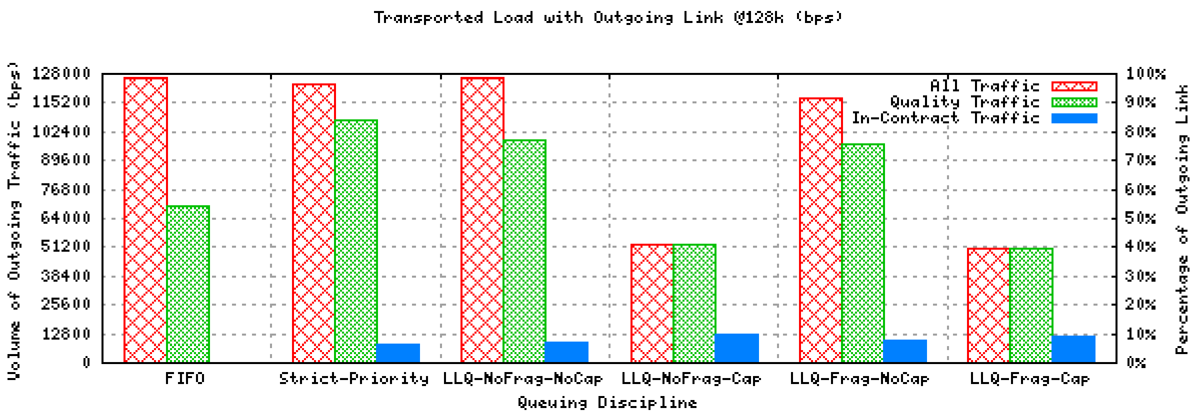
A.1.2.2 Transported Load



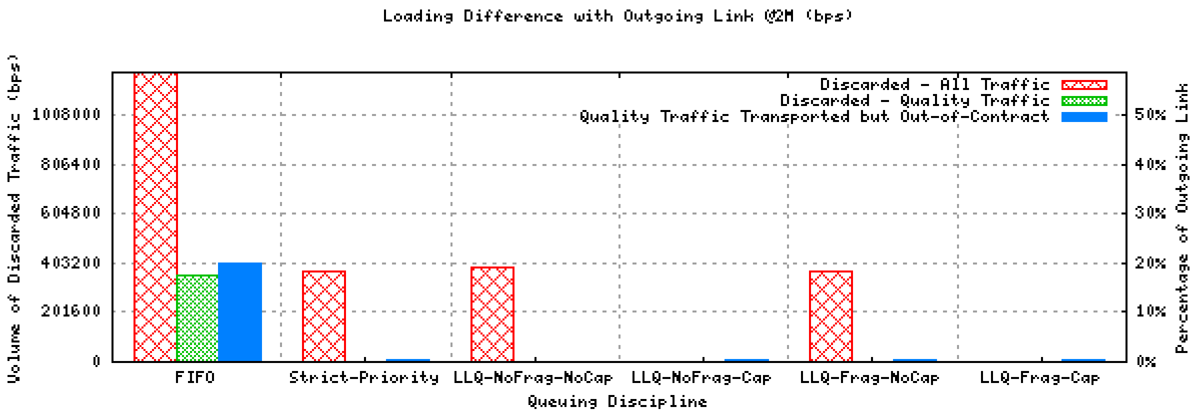


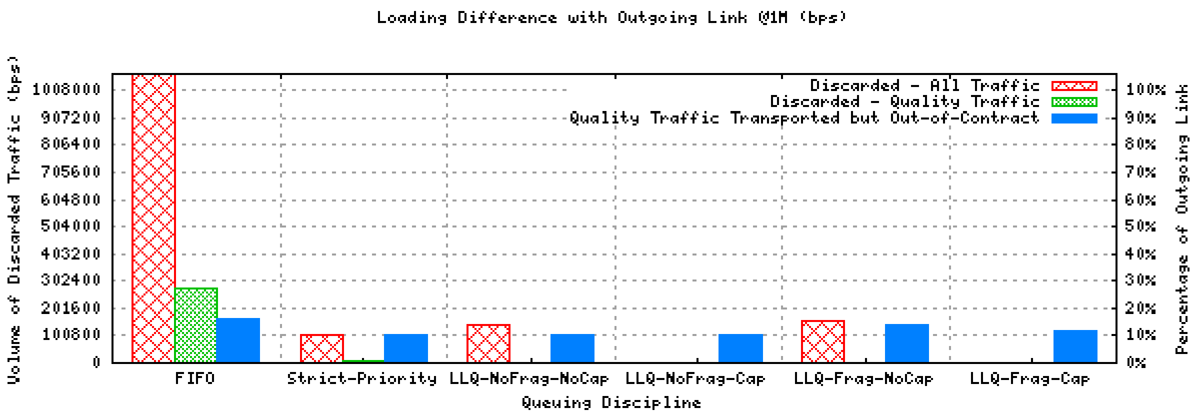


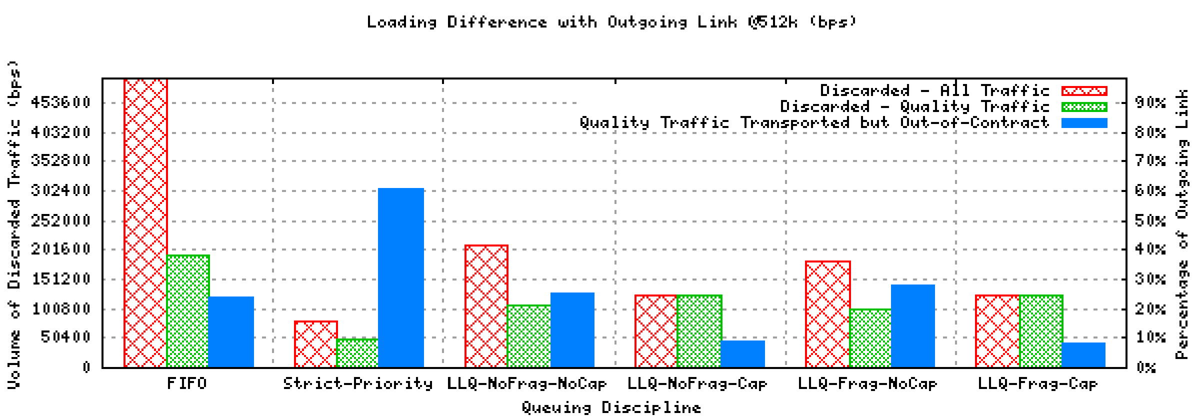

Transported Load with Outgoing Link @256k (bps)
Volume of Outgoing Traffic (bps)
252000
226800
201600
176400
151200
126000
100800
75600
50400
25200
0
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Percentage of Outgoing Link
All Traffic
Quality Traffic
In-Contract Traffic
FIFO
Strict-Priority
LLQ-NoFrag-NoCap
LLQ-NoFrag-Cap
LLQ-Frag-NoCap
LLQ-Frag-Cap
Queuing Discipline

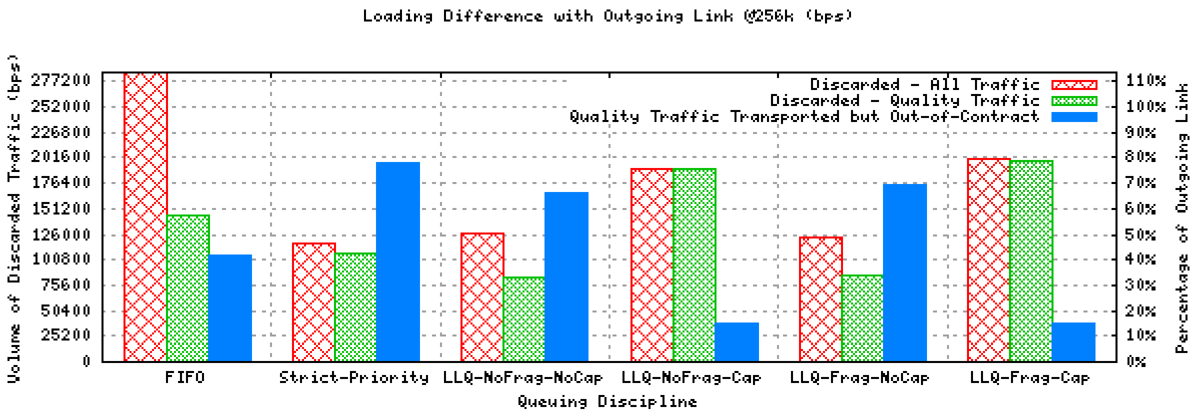

Transported Load with Outgoing Link @128k (bps)
Volume of Outgoing Traffic (bps)
128000
115200
102400
89600
76800
64000
51200
38400
25600
12800
0
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Percentage of Outgoing Link
All Traffic
Quality Traffic
In-Contract Traffic
FIFO
Strict-Priority
LLQ-NoFrag-NoCap
LLQ-NoFrag-Cap
LLQ-Frag-NoCap
LLQ-Frag-Cap
Queuing Discipline

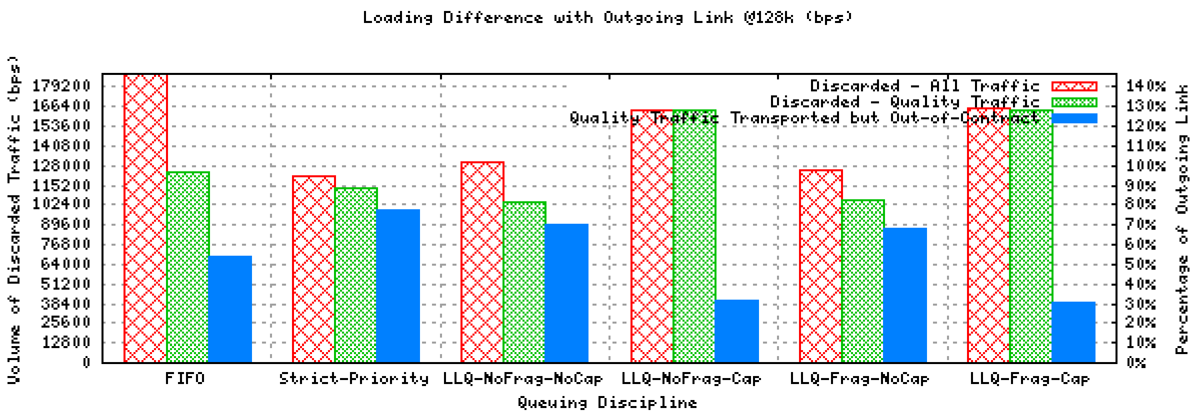
A.1.2.3 Load Difference








Loading Difference with Outgoing Link @256k (bps)
Volume of Discarded Traffic (bps)
277200
252000
226800
201600
176400
151200
126000
100800
75600
50400
25200
0
Discarded - All Traffic
Discarded - Quality Traffic
Quality Traffic Transported but Out-of-Contract
110%
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Percentage of Outgoing Link
FIFO
Strict-Priority
LLQ-NoFrag-NoCap
LLQ-NoFrag-Cap
LLQ-Frag-NoCap
LLQ-Frag-Cap
Queuing Discipline

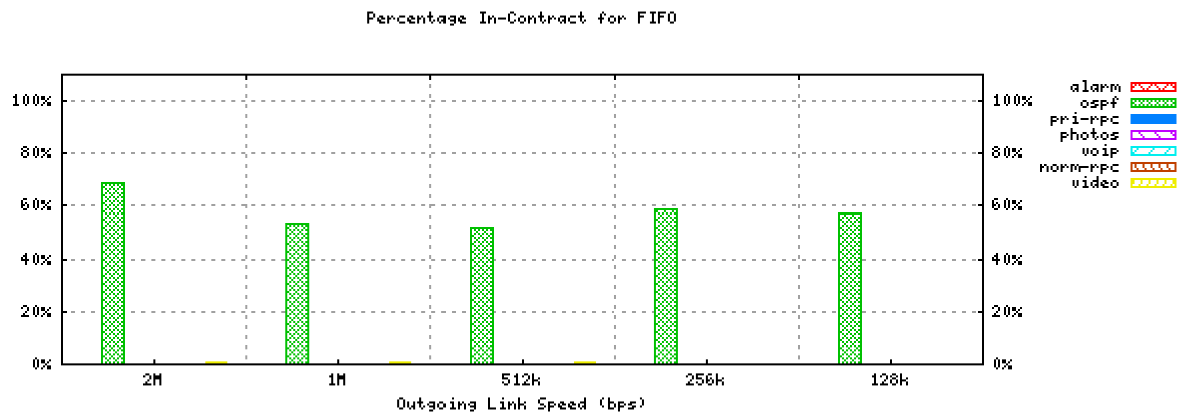

Loading Difference with Outgoing Link @128k (bps)
Volume of Discarded Traffic (bps)
179200
166400
153600
140800
128000
115200
102400
89600
76800
64000
51200
38400
25600
12800
0
Discarded - All Traffic
Discarded - Quality Traffic
Quality Traffic Transported but Out-of-Contract
140%
130%
120%
110%
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Percentage of Outgoing Link
FIFO
Strict-Priority
LLQ-NoFrag-NoCap
LLQ-NoFrag-Cap
LLQ-Frag-NoCap
LLQ-Frag-Cap
Queuing Discipline

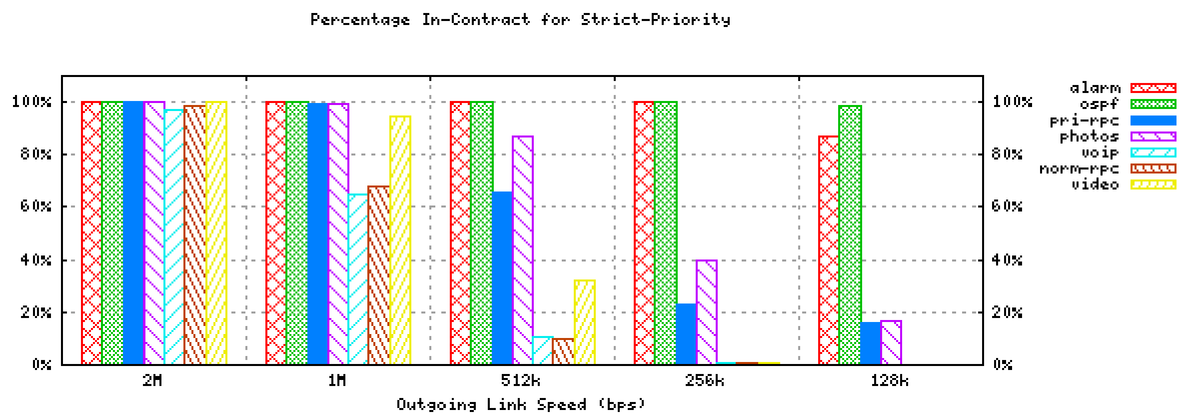
A.1.3 Per App Percentage In-Contract by Queuing Discipline

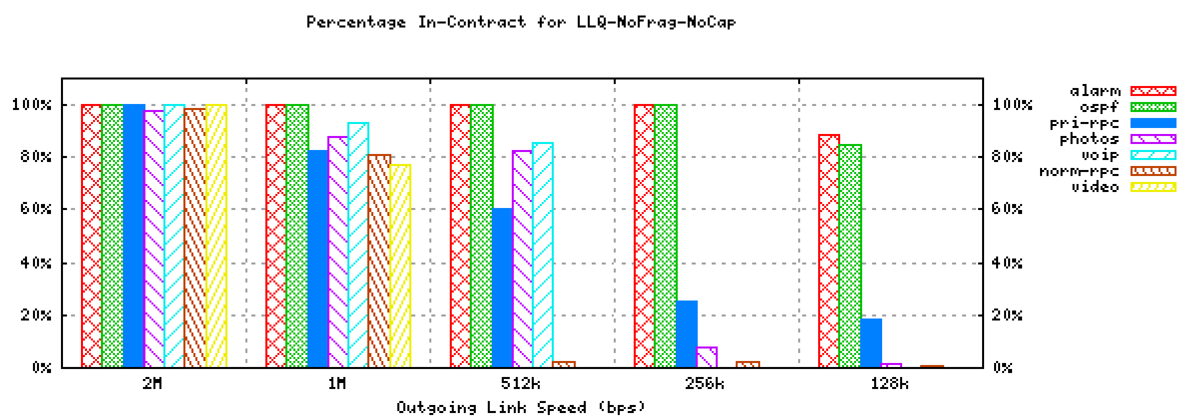
These graphs view the data from an application's perspective (as opposed to a total volume perspective in the graphs above). The individual, per application, bars capture what fraction of the time the application's traffic was delivered within the per-hop quality degradation budget. The graphs are per-queuing mechanism with all of the link speed variations on a single graph.

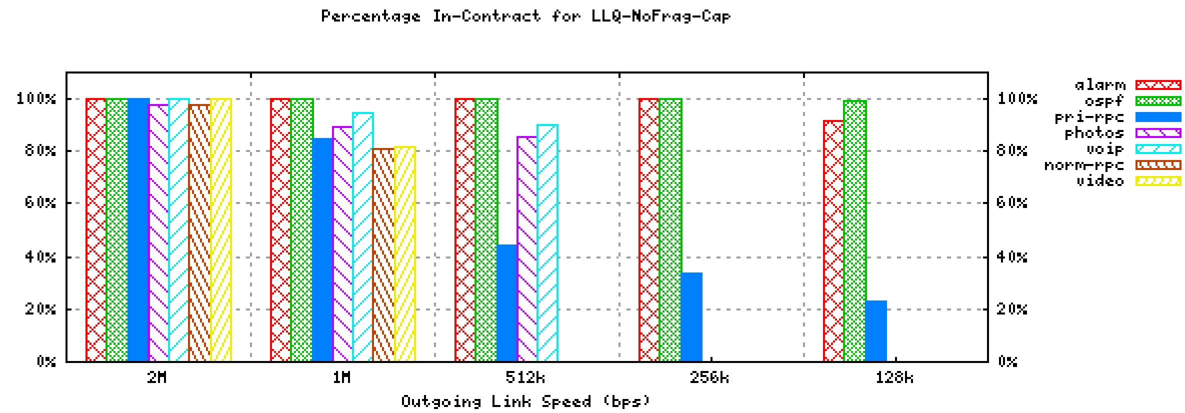
From these graphs the actual order in which the applications failed to be delivered quality, as the egress link speed was reduced, can be observed.

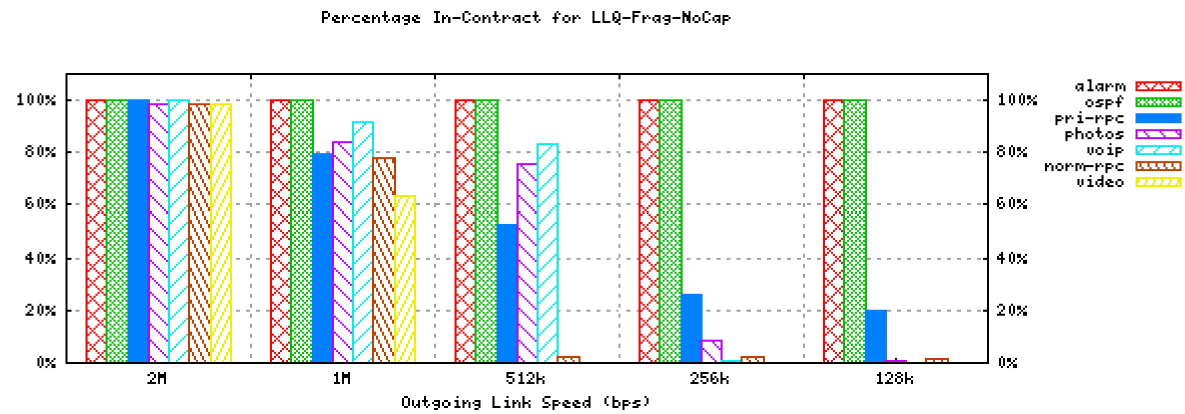
A.1.3.1 In-Contract

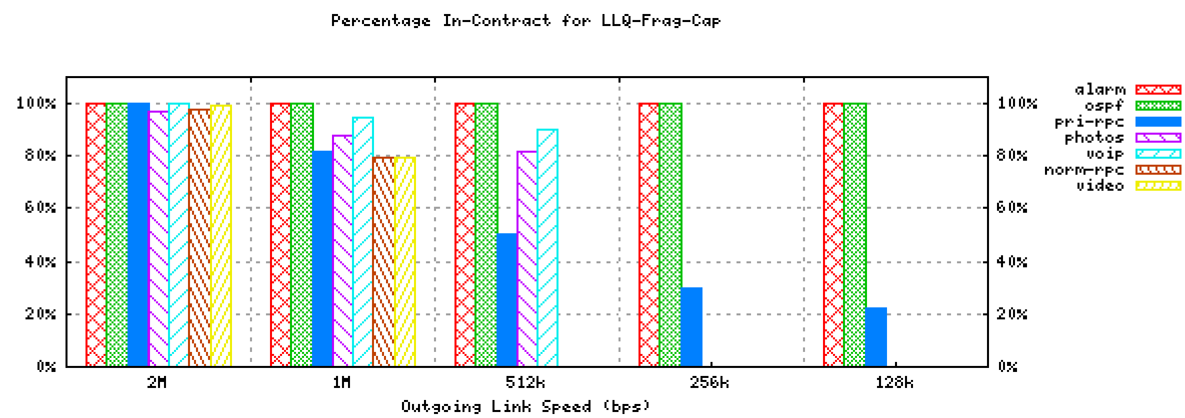




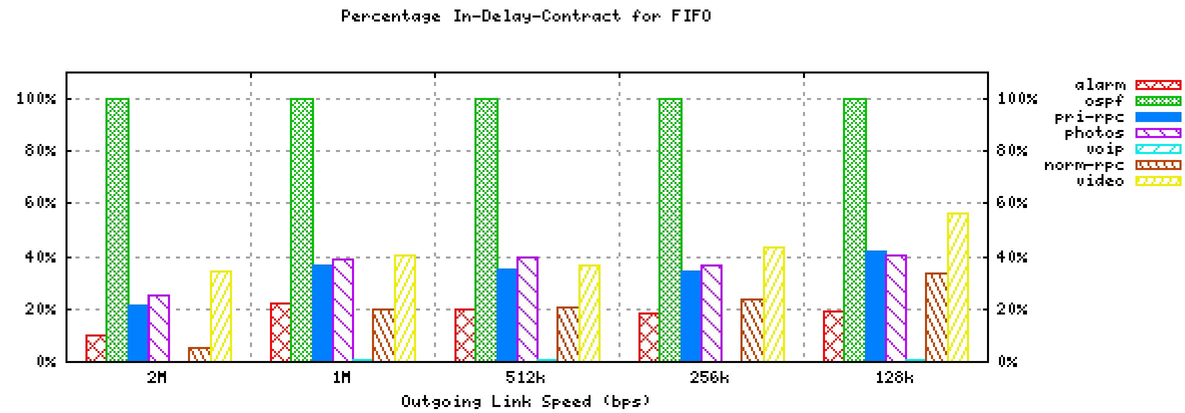


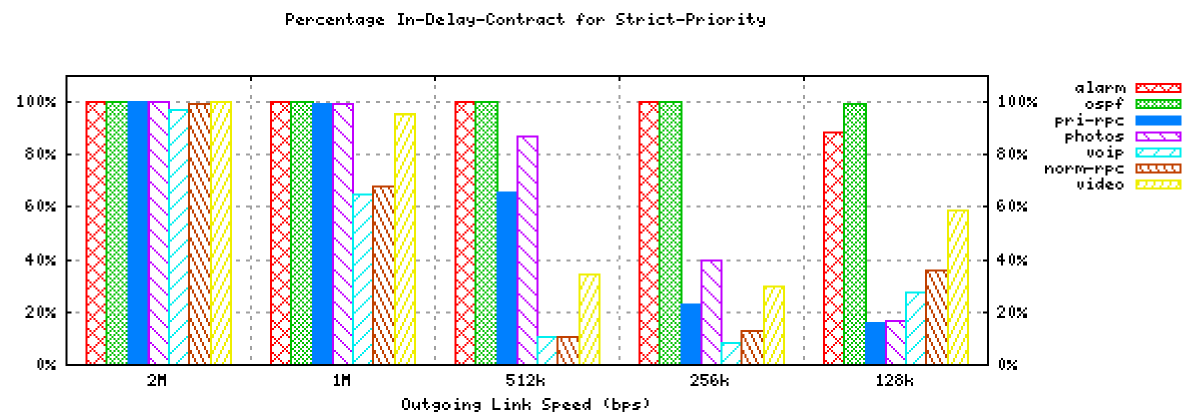

Percentage In-Contract for LLQ-NoFrag-Cap
100%
80%
60%
40%
20%
0%
2M
1M
512k
256k
128k
Outgoing Link Speed (bps)
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video

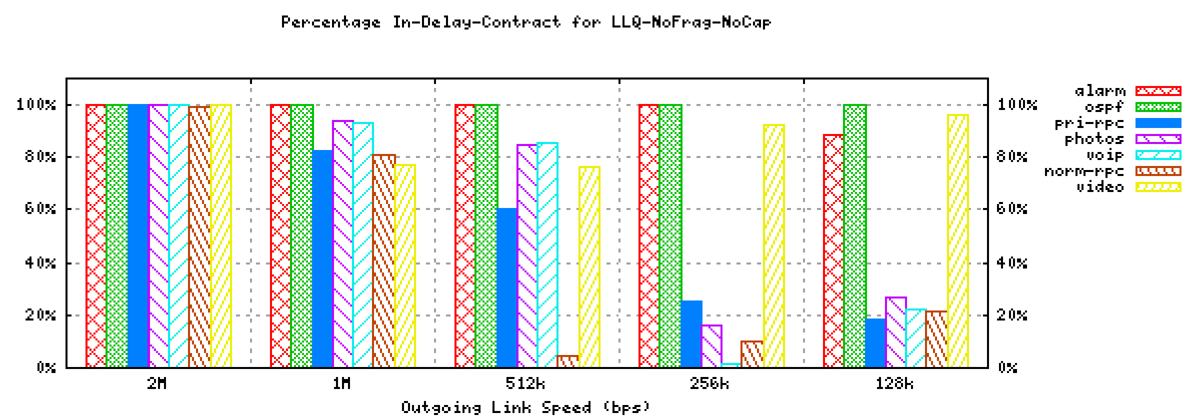

Percentage In-Contract for LLQ-Frag-NoCap
100%
80%
60%
40%
20%
0%
2M
1M
512k
256k
128k
Outgoing Link Speed (bps)
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video

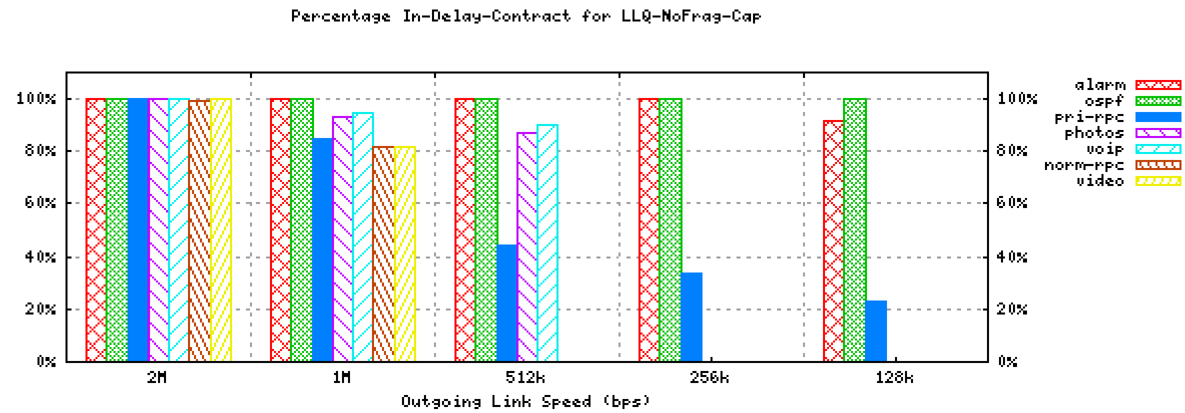

Percentage In-Contract for LLQ-Frag-Cap
100%
80%
60%
40%
20%
0%
2M
1M
512k
256k
128k
Outgoing Link Speed (bps)
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video

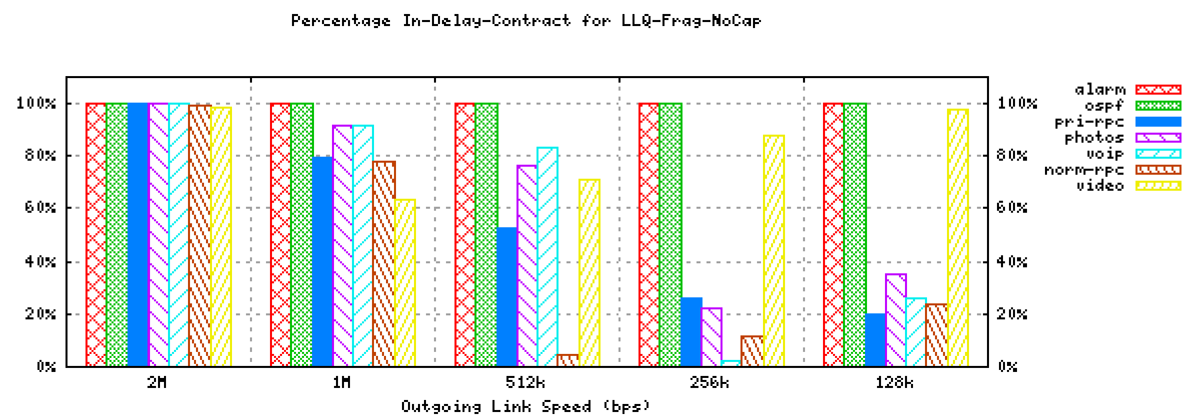
A.1.3.2 In-Delay Contract

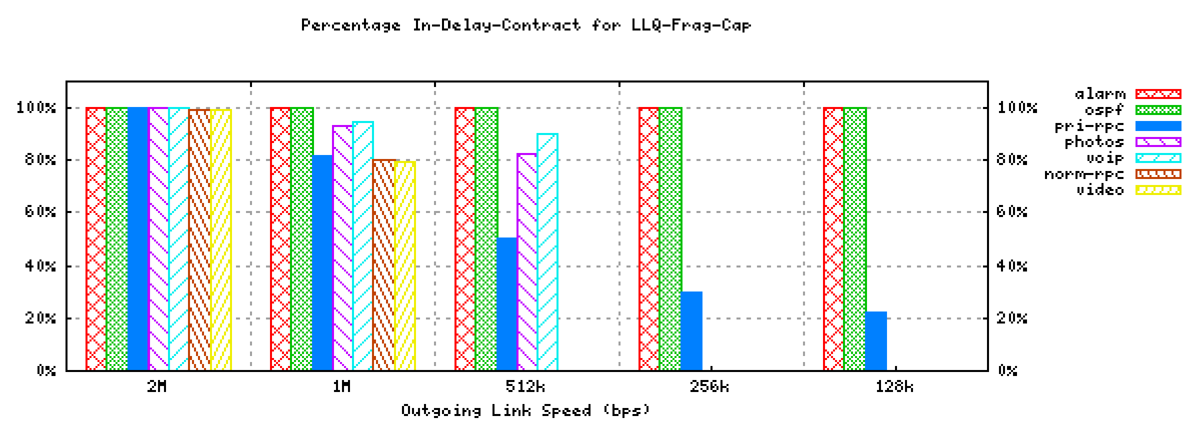




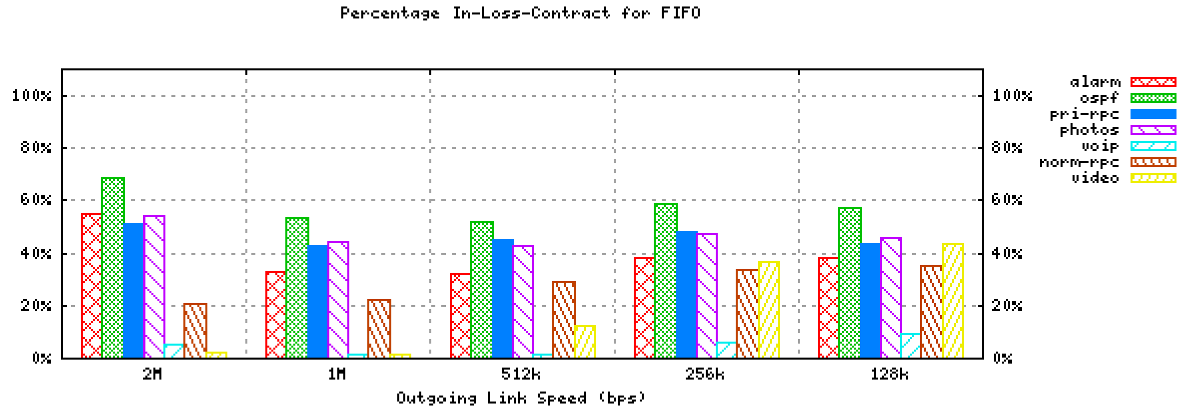


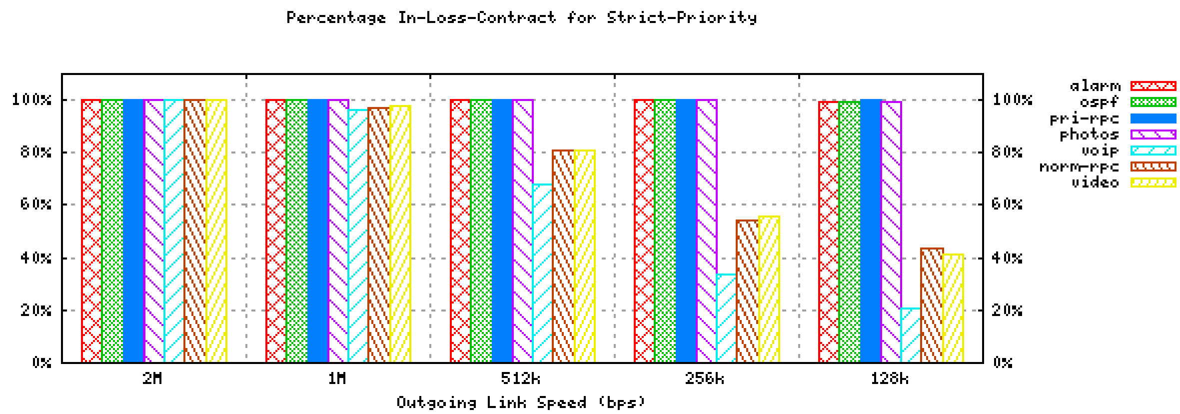

Percentage In-Delay-Contract for LLQ-NoFrag-Cap
100%
80%
60%
40%
20%
0%
2M
1M
512k
256k
128k
Outgoing Link Speed (bps)
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video

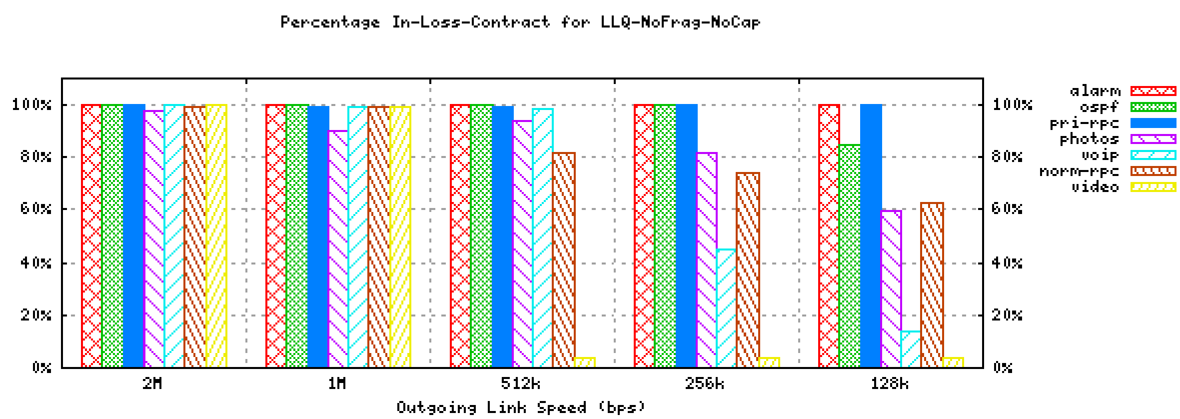

Percentage In-Delay-Contract for LLQ-Frag-NoCap
100%
80%
60%
40%
20%
0%
2M
1M
512k
256k
128k
Outgoing Link Speed (bps)
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video


Percentage In-Delay-Contract for LLQ-Frag-Cap
100%
80%
60%
40%
20%
0%
2M
1M
512k
256k
128k
Outgoing Link Speed (bps)
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video

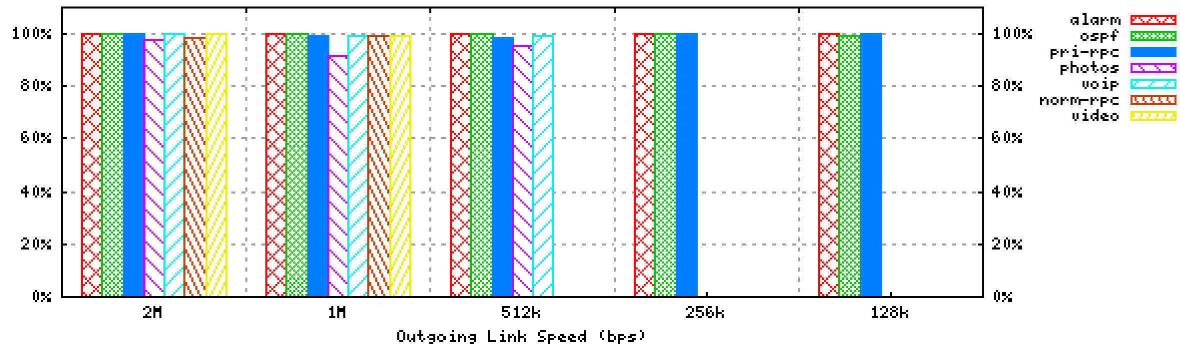
A.1.3.3 In-Loss Contract



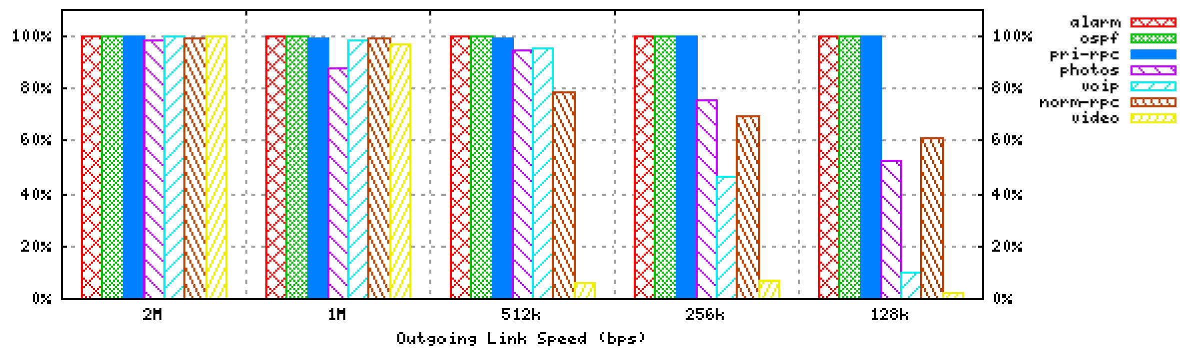




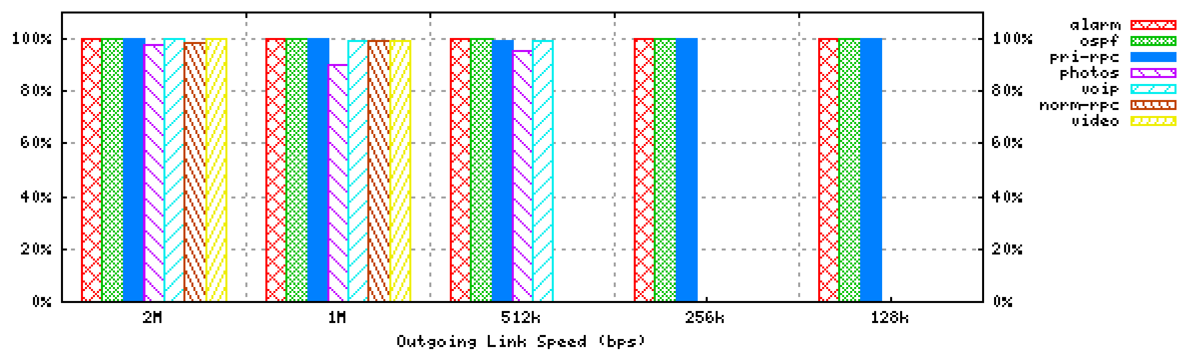
Percentage In-Loss-Contract for LLQ-NoFrag-Cap



Percentage In-Loss-Contract for LLQ-Frag-NoCap



Percentage In-Loss-Contract for LLQ-Frag-Cap



A.2 Mixed Speed Link Test Results

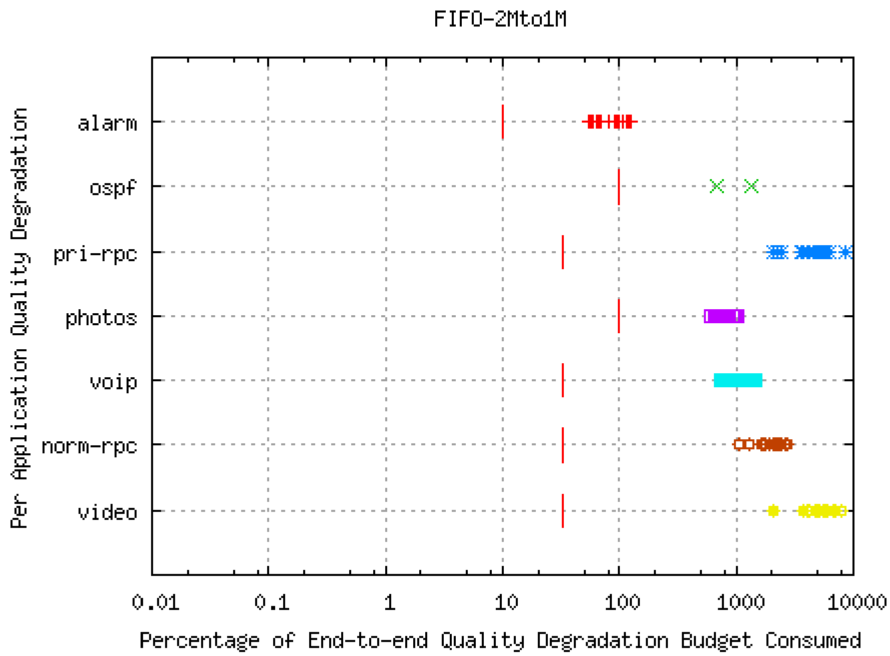
These tests were performed as described in section 6.2, the egress capacity was stepped down with the configuration first remaining the same, then being changed to match the link speed (the same configuration as used in the fixed speed link tests).

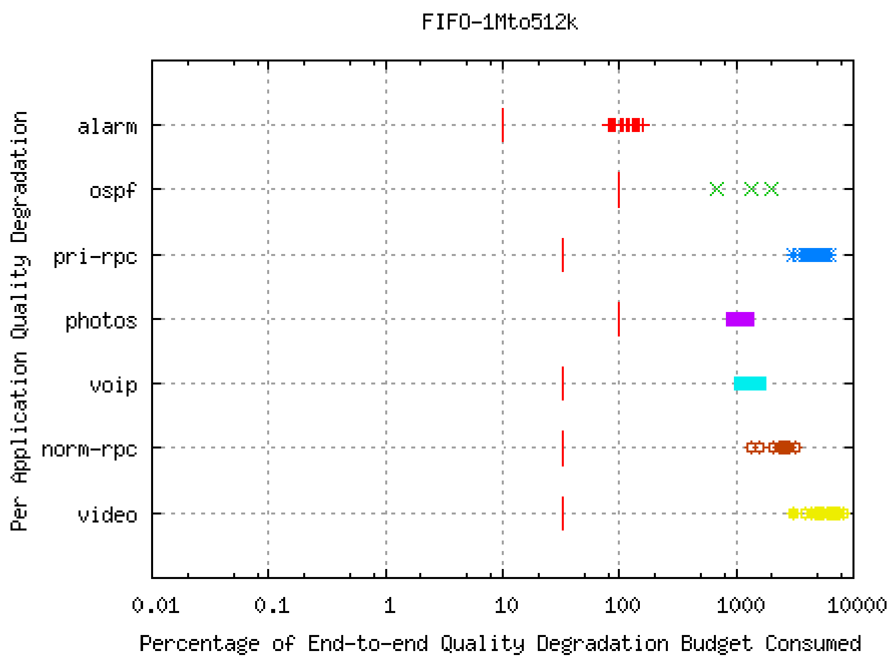
A.2.1 Consumed DQ Scatter Plots

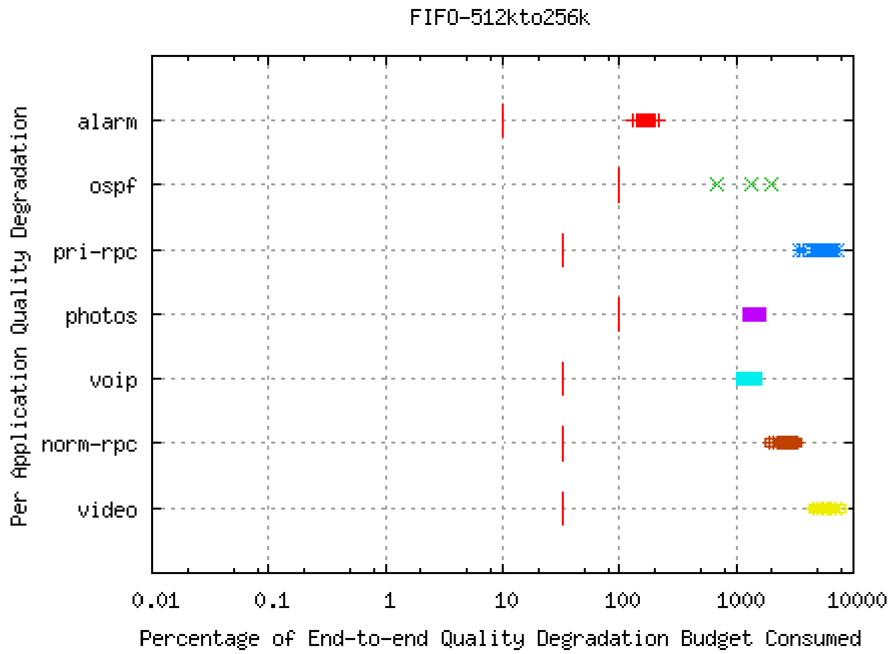
These plots represent, for a given queuing discipline at a given speed, the amount of end-to-end budget (in terms of both loss and delay) that an application has consumed. The vertical red bar represents the per-hop budget, with 100% representing the end-to-end budget. A mark to the left of the vertical red bar indicates a run which did not consume more than the per-hop budget. The abscissa is on a logarithmic scale. If all the points lay exactly on the vertical red bar this would represent “perfection” – all applications received exactly the quality they needed.

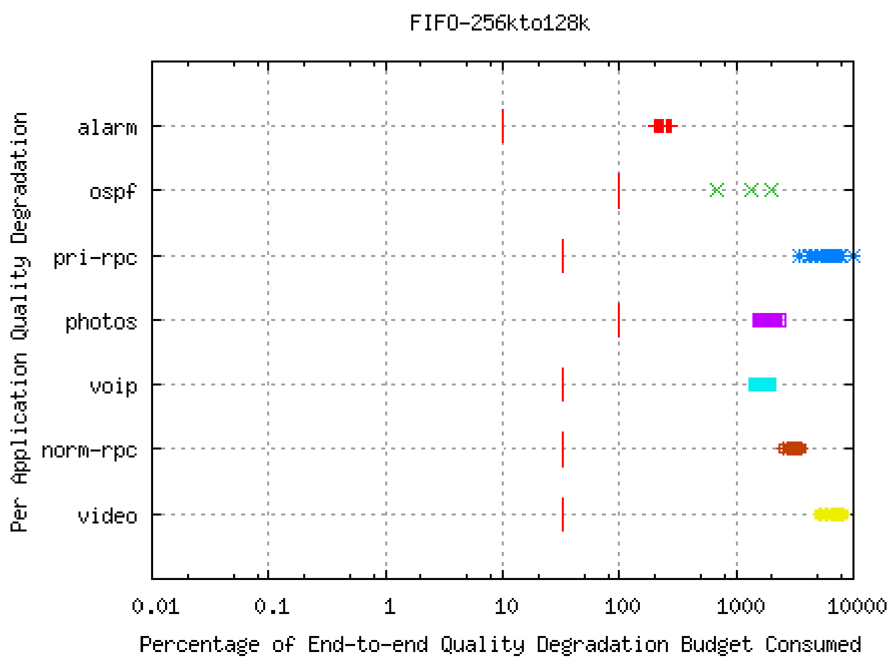
Where the link speed and the configuration were the same as the fixed speed link the graphs have been omitted – they are available in the dataset, but did not add any new information over and above those fixed link speed cases. Only the transition case is reproduced here.

A.2.1.1 First-In-First-Out (FIFO)

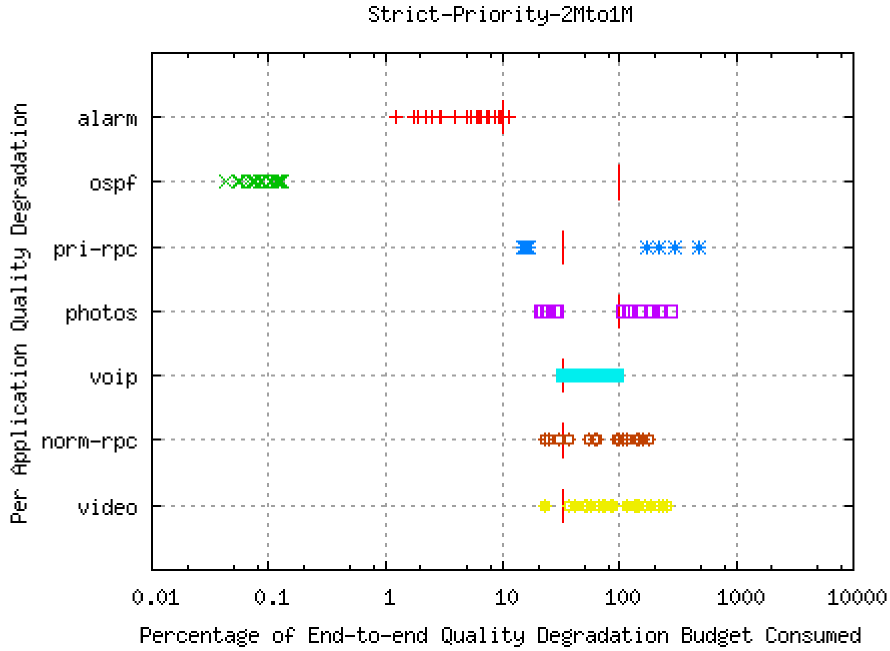


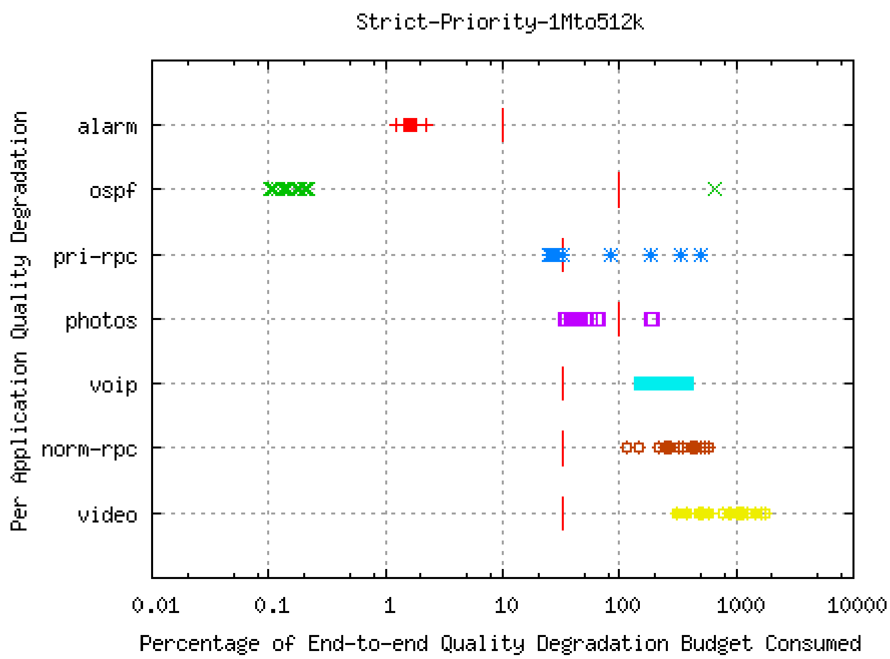


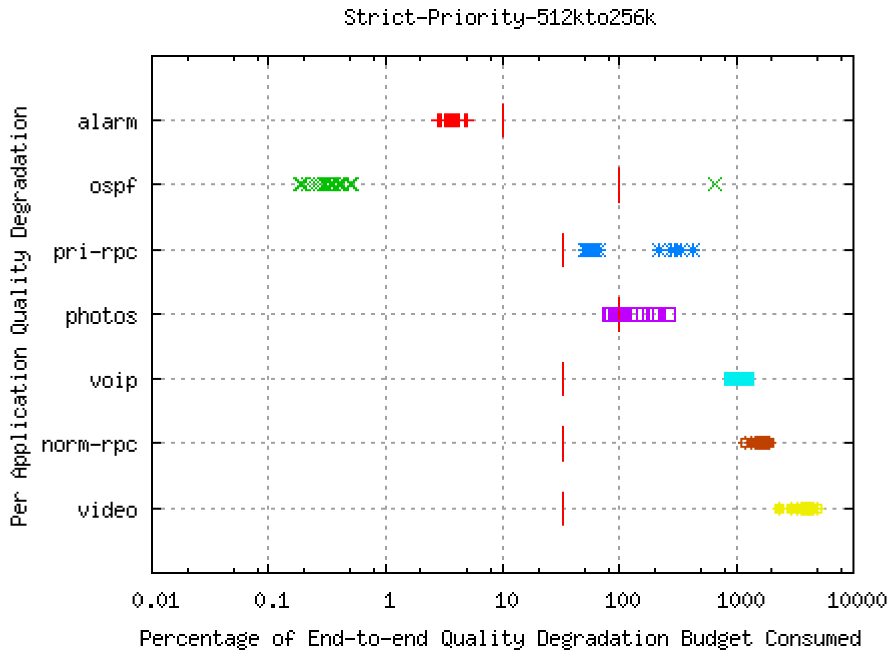

FIFO-512kto256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

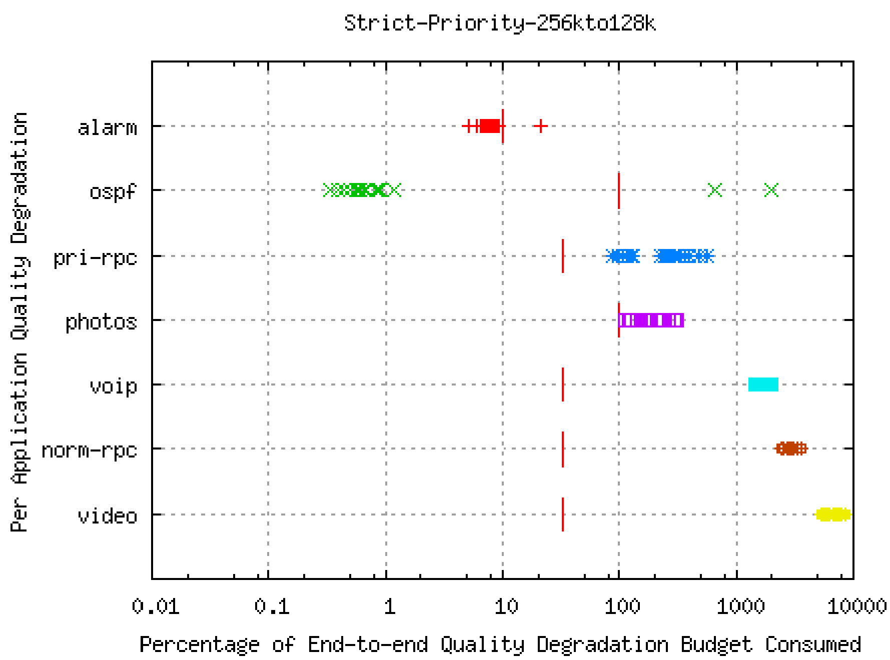

FIFO-256kto128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

A.2.1.2 Strict Priority Queuing

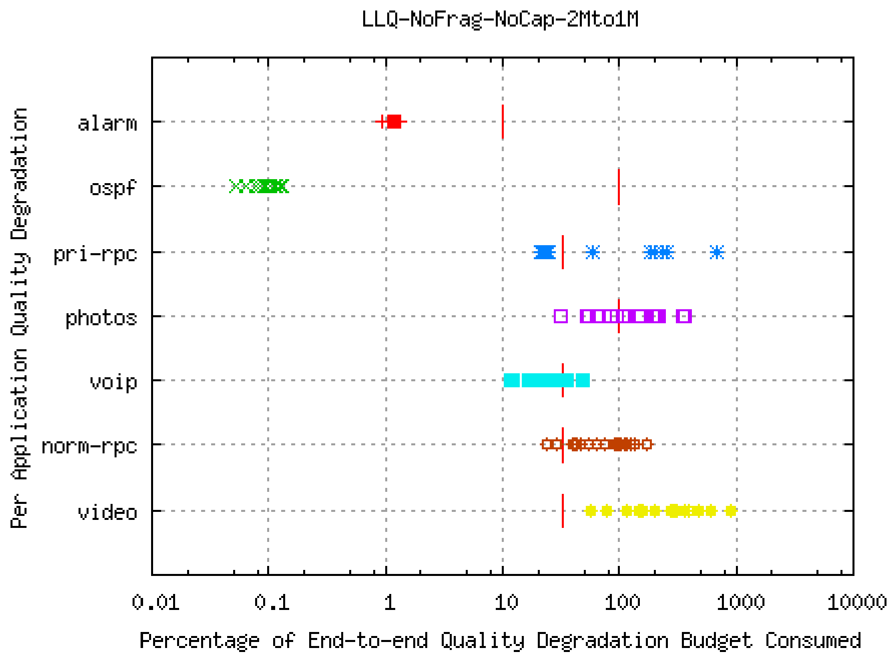


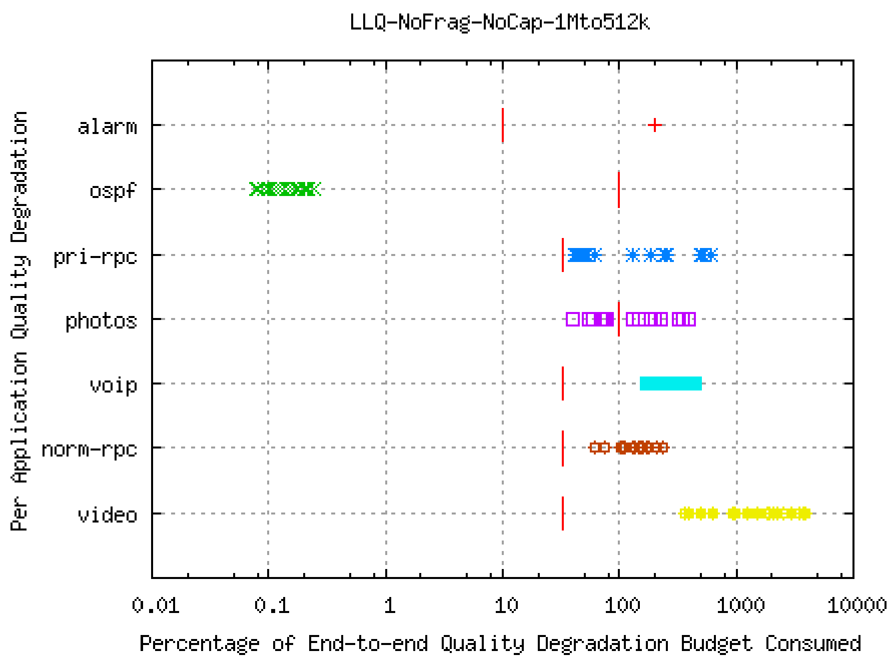


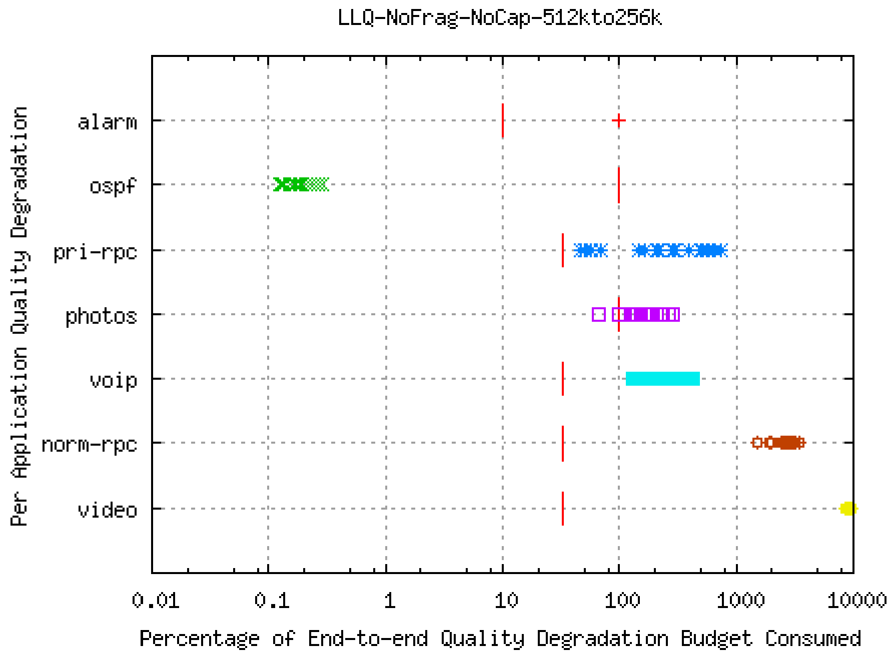

Strict-Priority-512kto256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

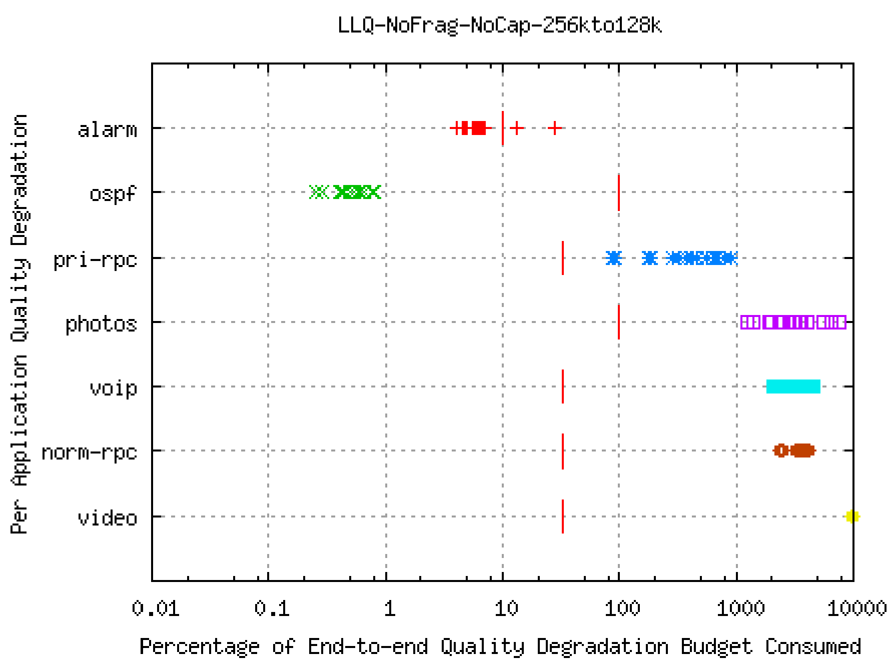

Strict-Priority-256kto128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

A.2.1.3 Low Latency Queuing (no mitigation)

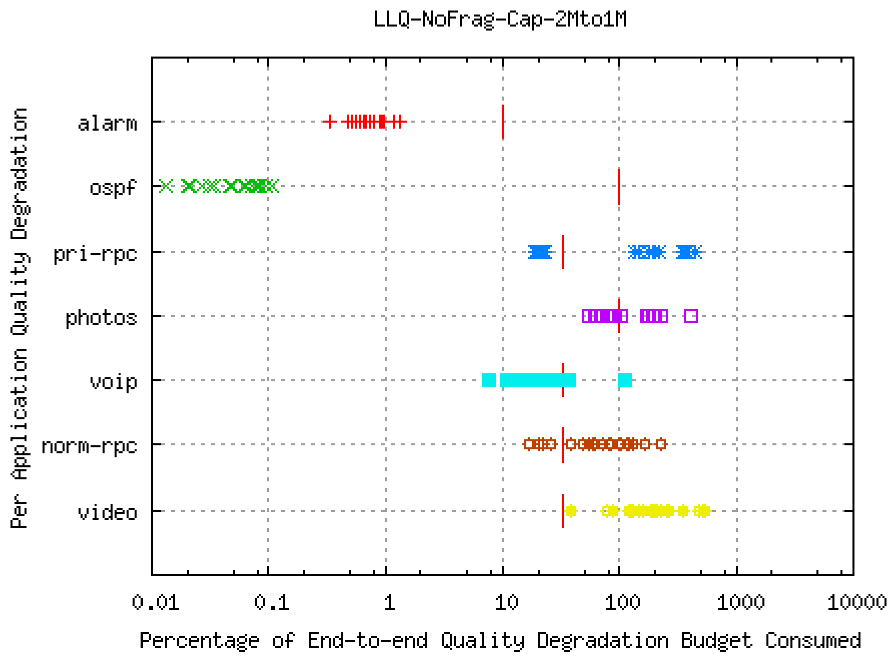


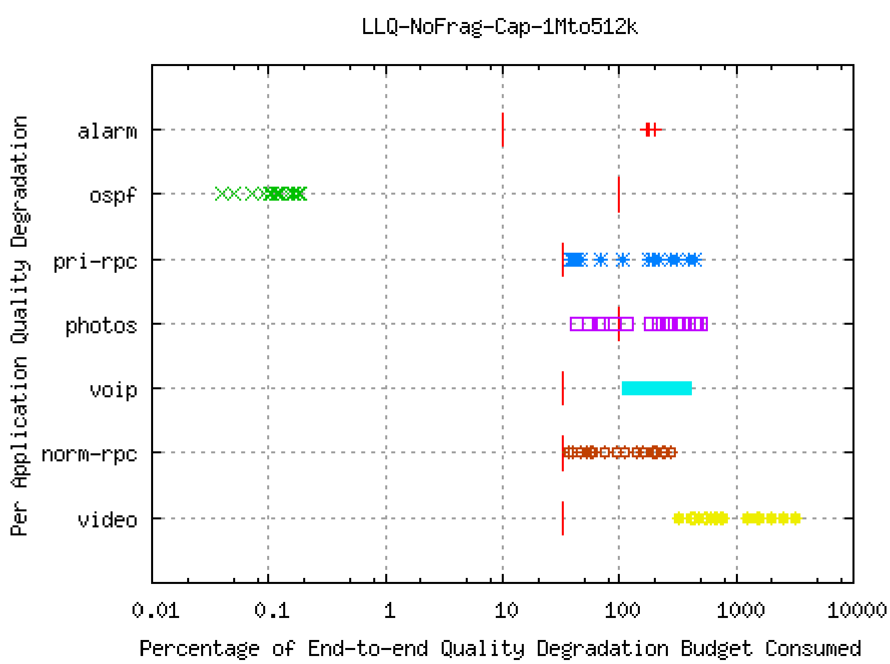


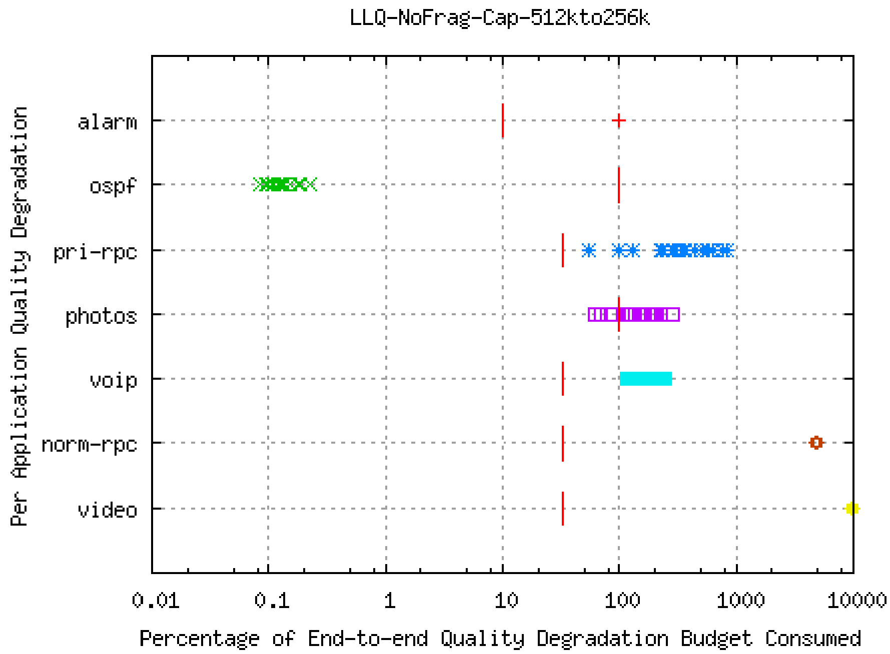

LLQ-NoFrag-NoCap-512kto256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

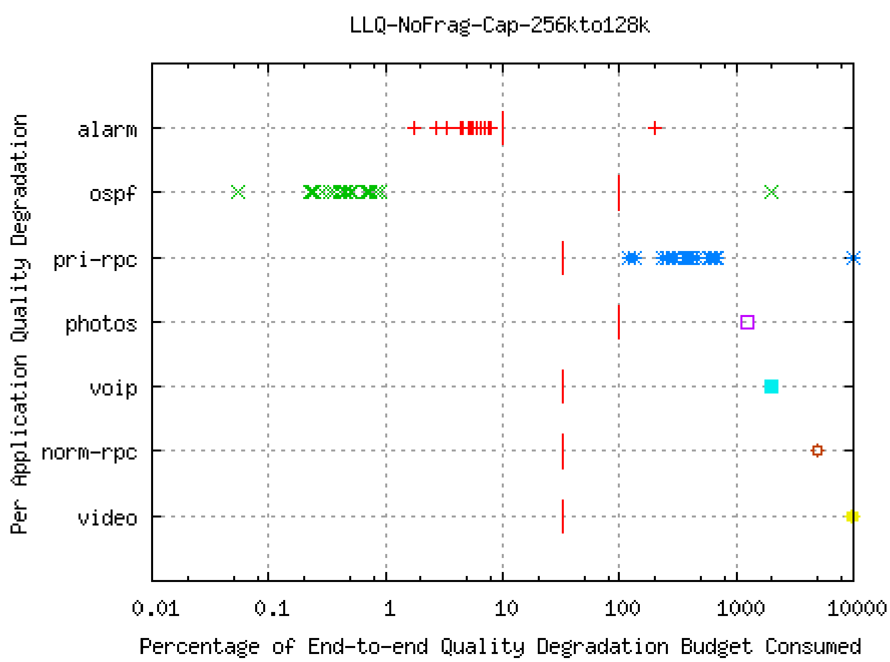

LLQ-NoFrag-NoCap-256kto128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

A.2.1.4 Low Latency Queuing (with selected Traffic Capped)

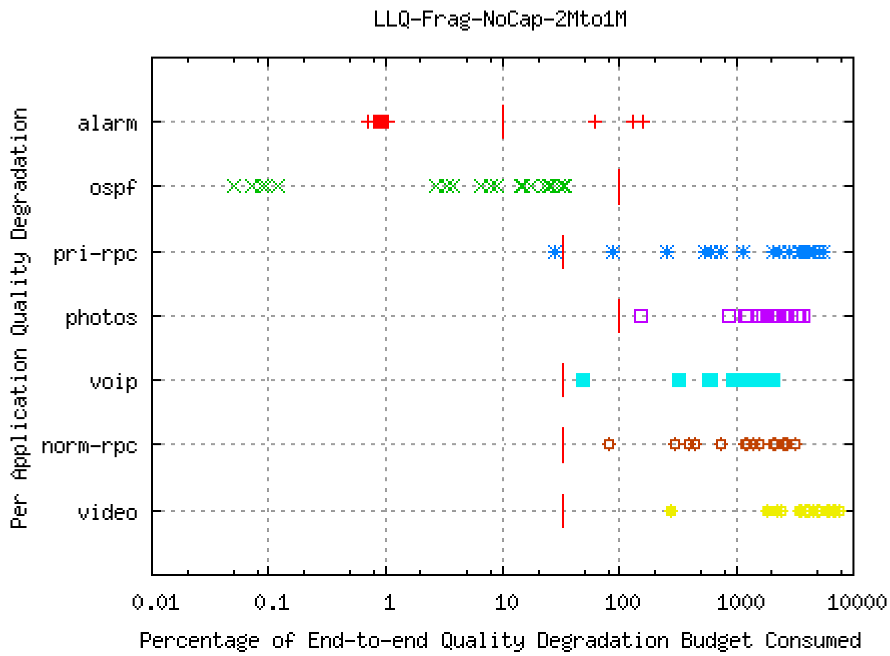


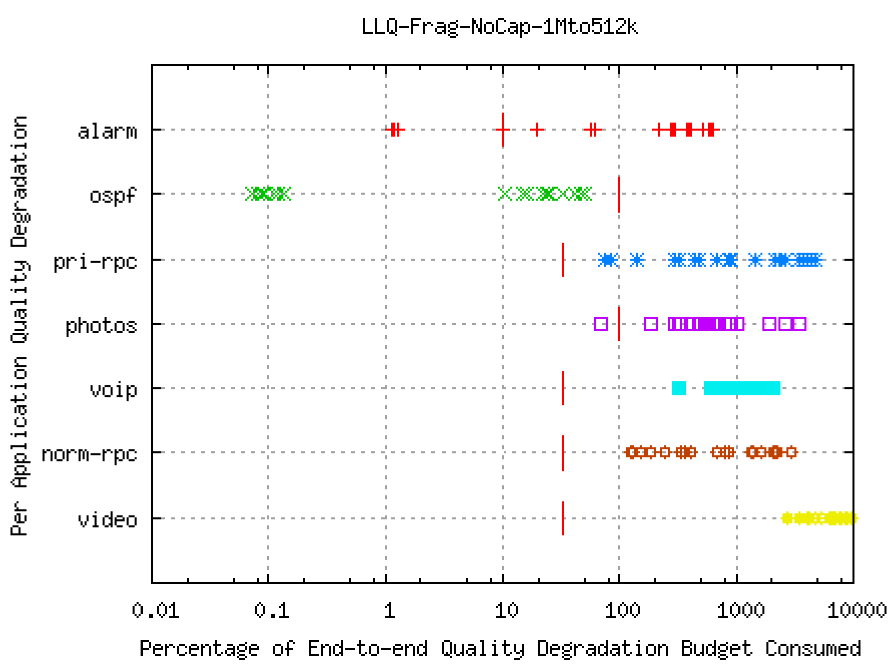


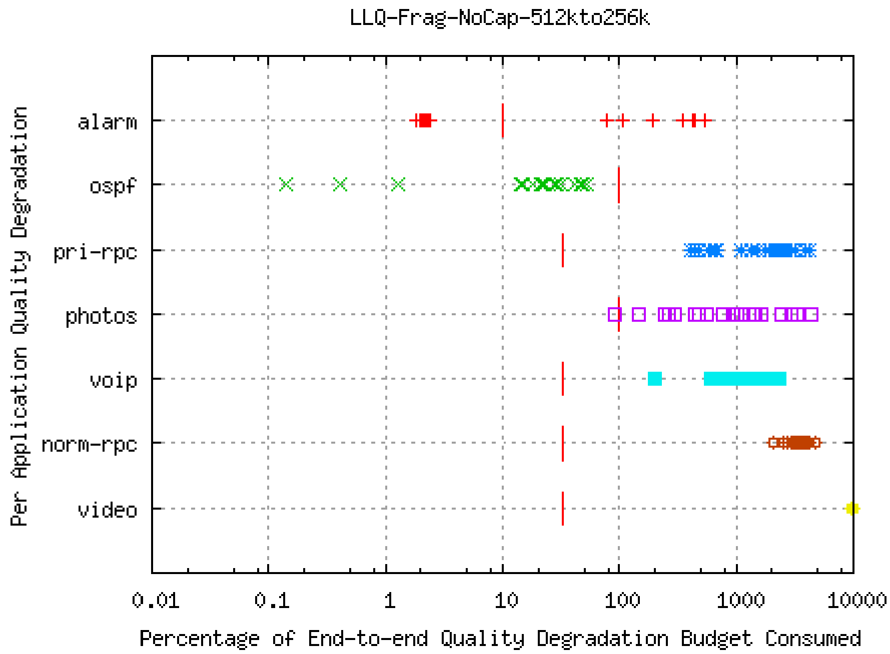

LLQ-NoFrag-Cap-512kto256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

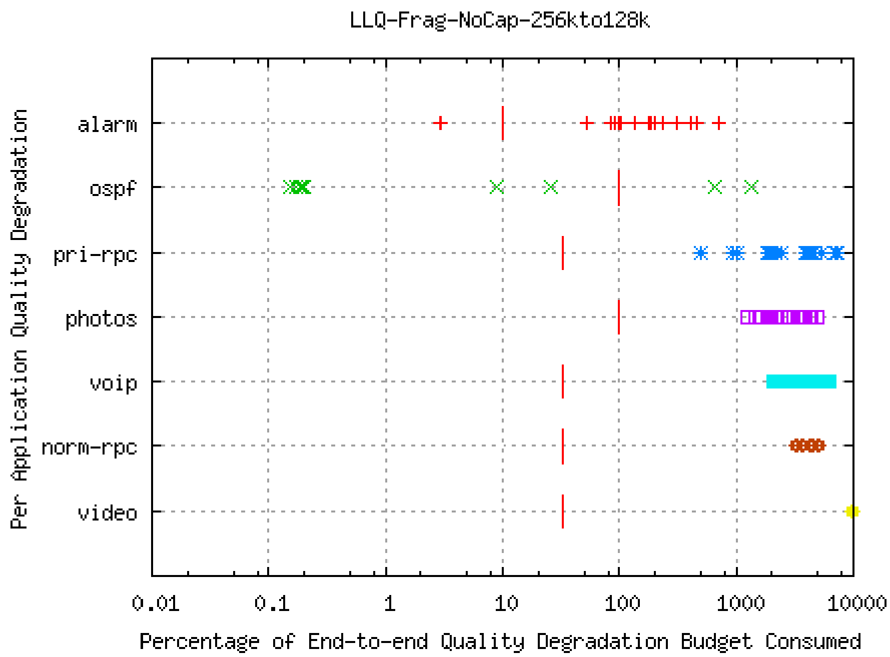

LLQ-NoFrag-Cap-256kto128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

A.2.1.5 Low Latency Queuing (with Packet Fragmentation)

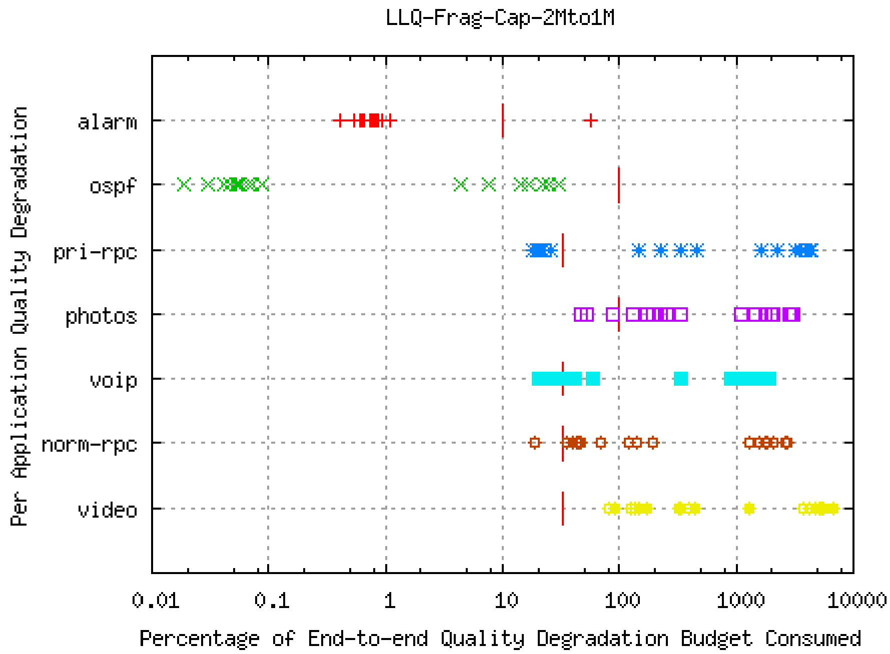


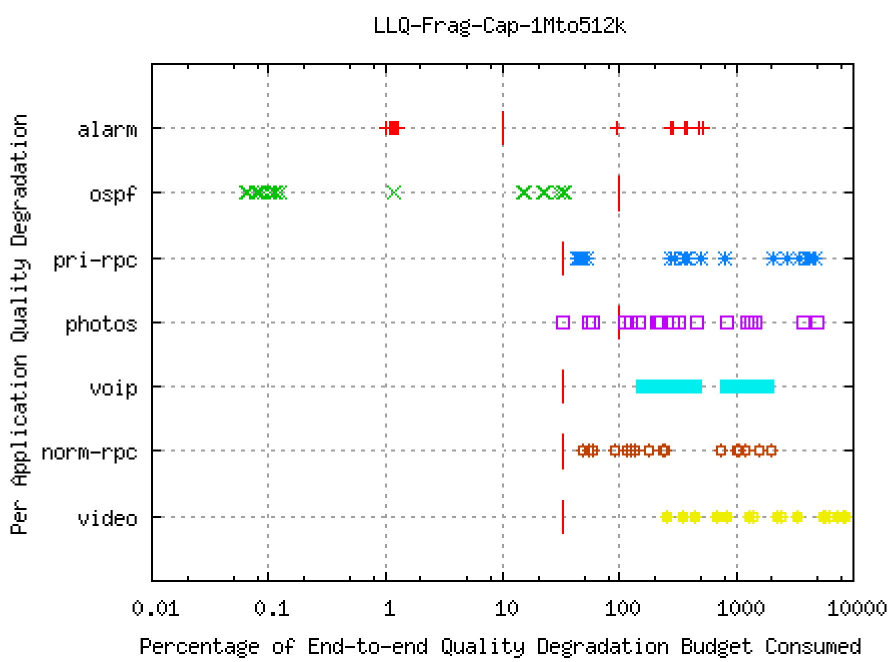


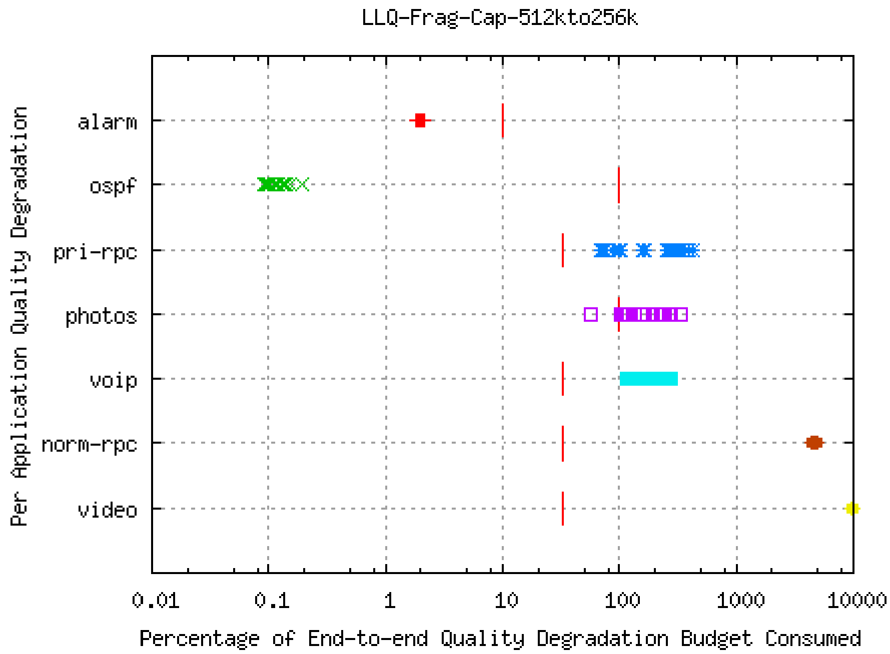

LLQ-Frag-NoCap-512kto256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

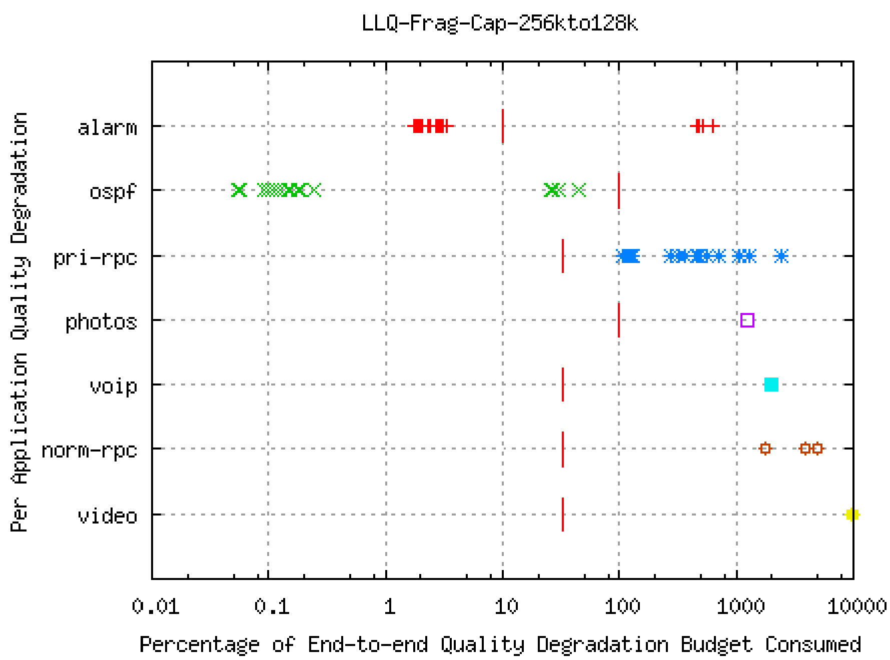

LLQ-Frag-NoCap-256kto128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

A.2.1.6 Low Latency Queuing (with Packet Fragmentation and selected Traffic Capped*)*






LLQ-Frag-Cap-512kto256k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed


LLQ-Frag-Cap-256kto128k
Per Application Quality Degradation
alarm
ospf
pri-rpc
photos
voip
norm-rpc
video
0.01
0.1
1
10
100
1000
10000
Percentage of End-to-end Quality Degradation Budget Consumed

A.2.2 Link Utilization Percentages

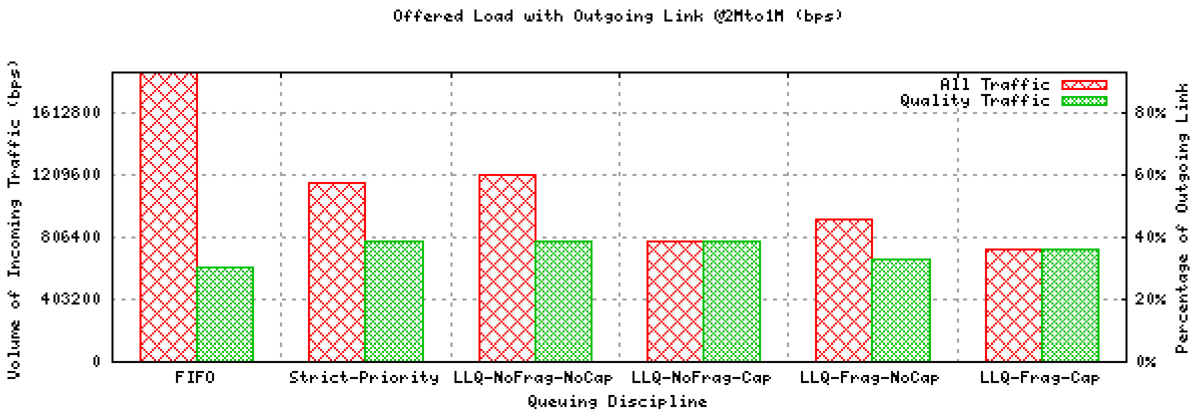
The Link Utilization Percentages are split into three graphs; *offered load*, *transported load* and *load difference*, again there is a graph for each link speed and each queuing mechanism variation. All three types of graphs have the application of interest on the x-axis, and two y-axes. The left y-axis is the absolute volume of traffic in bits per second; the right y-axis measures the fraction of the outgoing link that that volume represents at that speed.

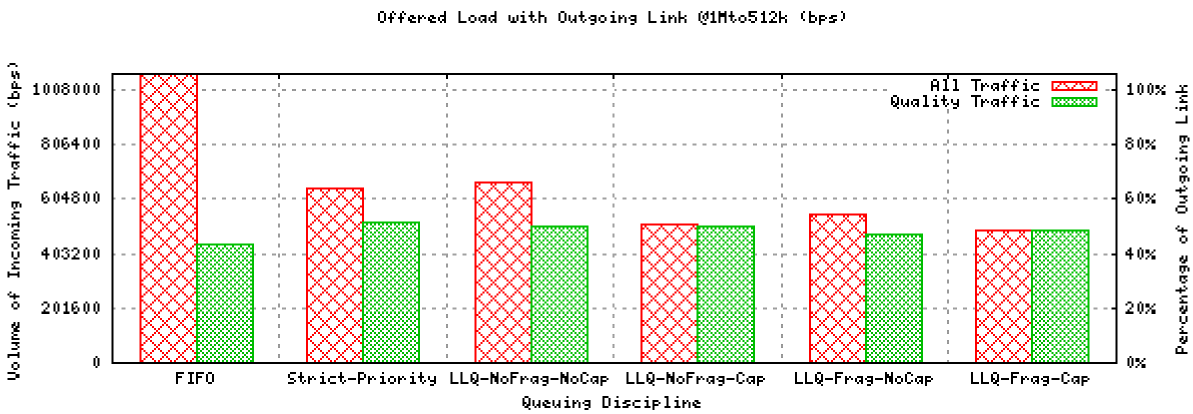
The *Offered Load* graph shows you the amount of traffic that was offered to the device under test. The two bars per queuing discipline represent the total amount of offered traffic, and the total amount of traffic offered that required quality guarantees (the "quality" traffic).

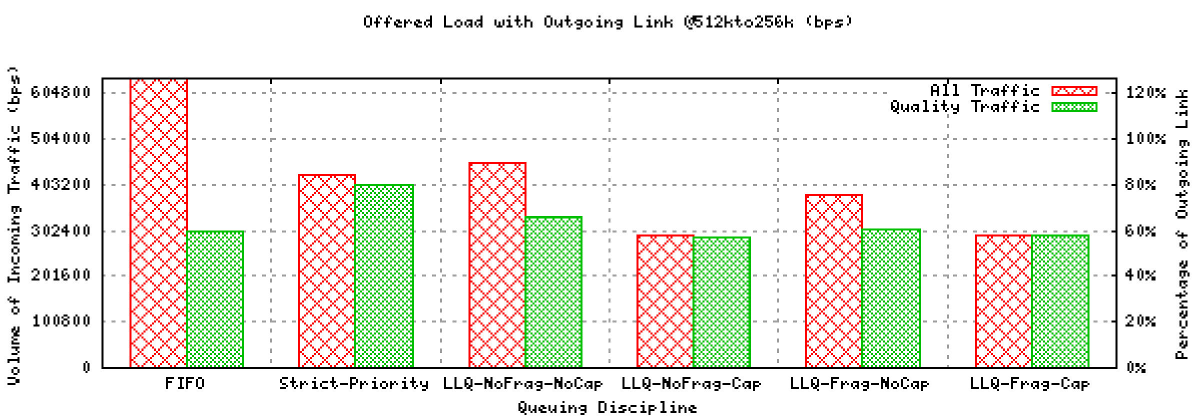
The *Transported Load* graph has three bars per queuing discipline. The fist bar *All Traffic* indicates the total amount of traffic that was transported (departed through the egress link). The second bar shows that amount of quality traffic transported. The third bar represents the amount of quality traffic that was transported in-contract; that is traversed the network element within the appropriate quality constraints. The first bar can be thought of as the total amount of “work” done, the second as the amount work done for traffic with a quality constraint (irrespective of delivered quality), and the third as the amount of useful work done on behalf of the quality traffic (delivered within the respective quality constraints).

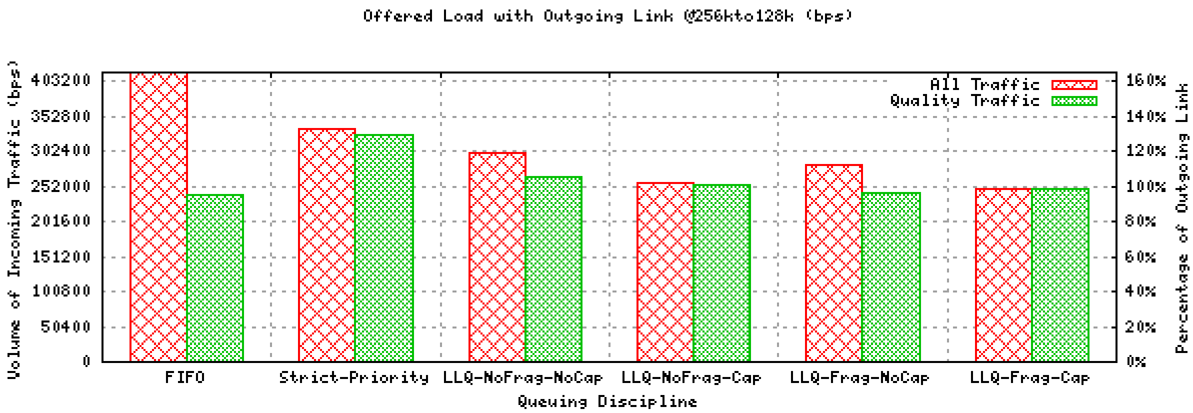
The final graph *Load Difference* has three bars per queuing discipline. The first bar *Discarded - All traffic* represents the difference between the total load offered and the total load transported. For example if the network element was offered 2.2M, and the egress link was configured at 1M and fully utilized, then this bar will be 1.2M (the total volume of traffic discarded). The second bar *Discarded - Quality Traffic* is the difference between the offered and transported load for just the quality traffic. The third bar is a little more interesting. It represents the amount of quality traffic that was transported but was “out of contract” (work done that may or may not be of use to the applications). This is the difference between the *Quality Traffic* and *In-contract Traffic* bars as shown on the *Transported Load* graph.

A.2.2.1 Offered Load



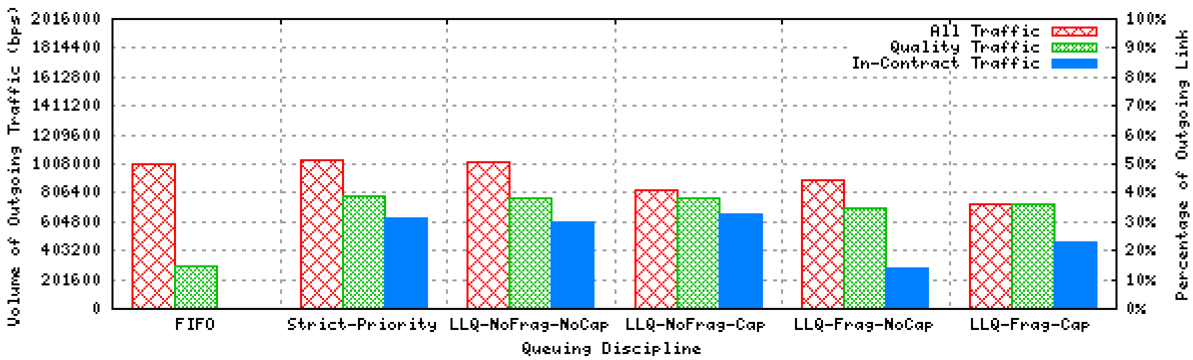




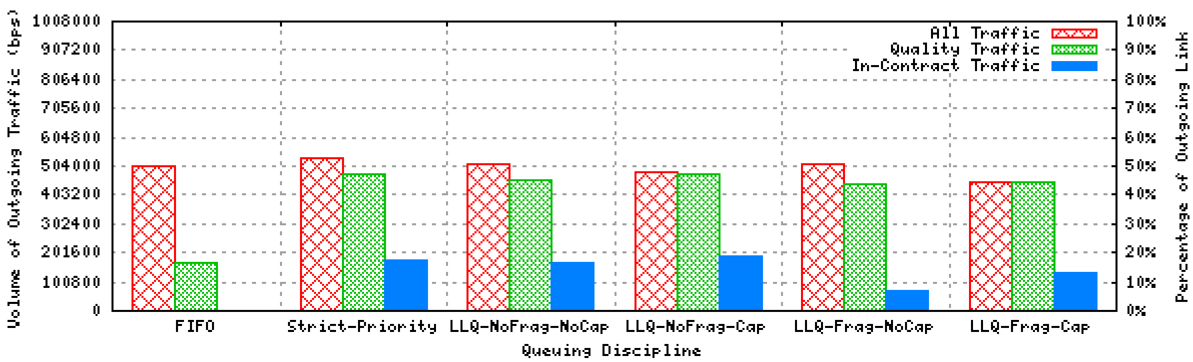

Offered Load with Outgoing Link @256kto128k (bps)
Volume of Incoming Traffic (bps)
Percentage of Outgoing Link
403200
352800
302400
252000
201600
151200
100800
50400
0
160%
140%
120%
100%
80%
60%
40%
20%
0%
All Traffic
Quality Traffic
FIFO
Strict-Priority
LLQ-NoFrag-NoCap
LLQ-NoFrag-Cap
LLQ-Frag-NoCap
LLQ-Frag-Cap
Queuing Discipline

A.2.2.2 Transported Load

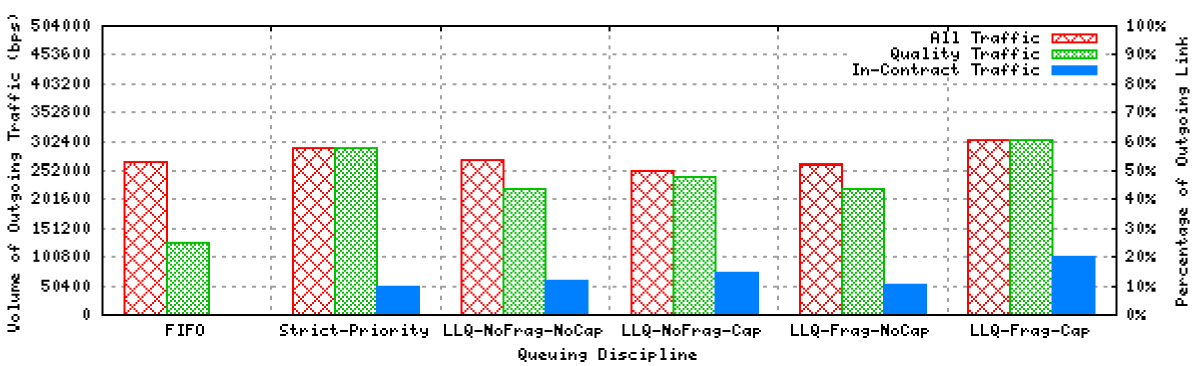
Transported Load with Outgoing Link @2Mto1M (bps)

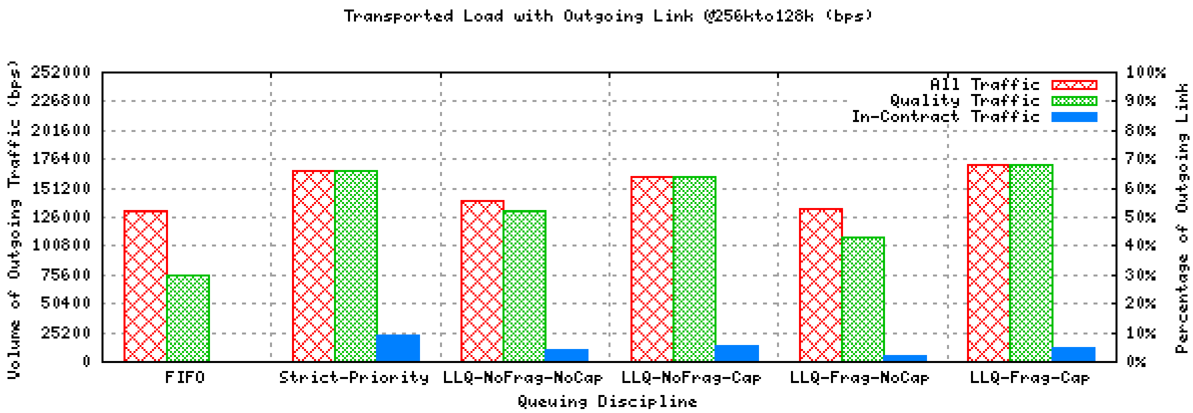


Transported Load with Outgoing Link @1Mto512k (bps)

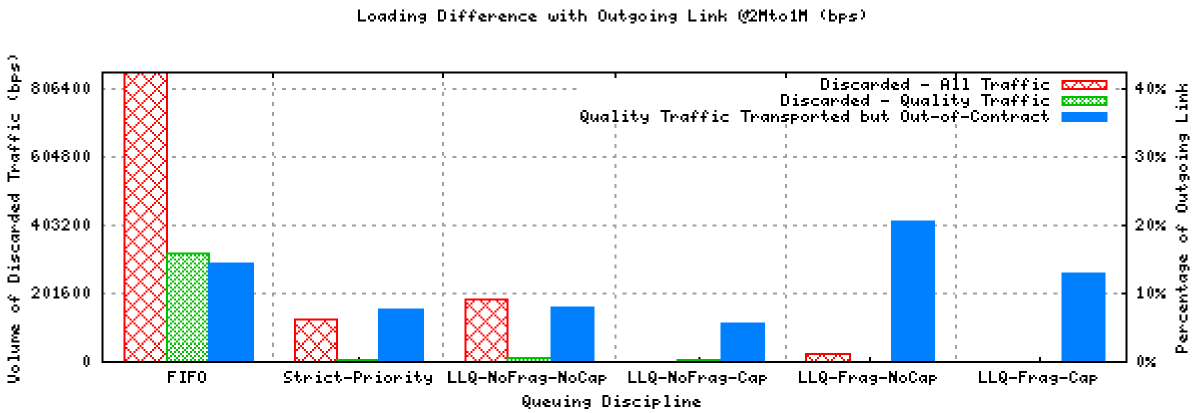


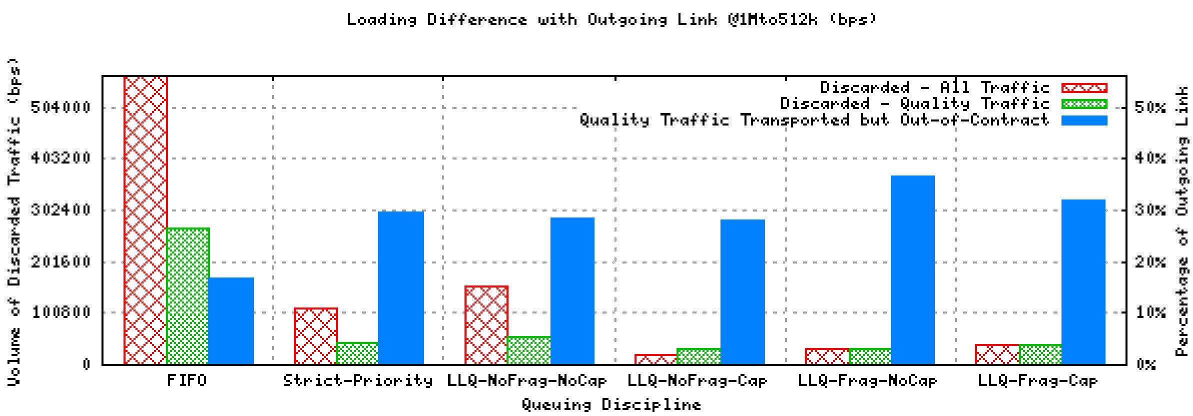
Transported Load with Outgoing Link @512kto256k (bps)

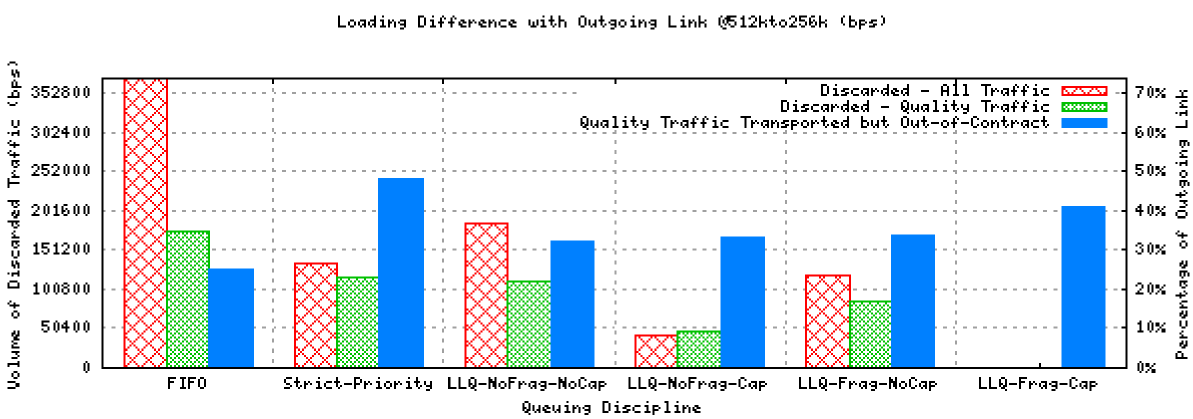


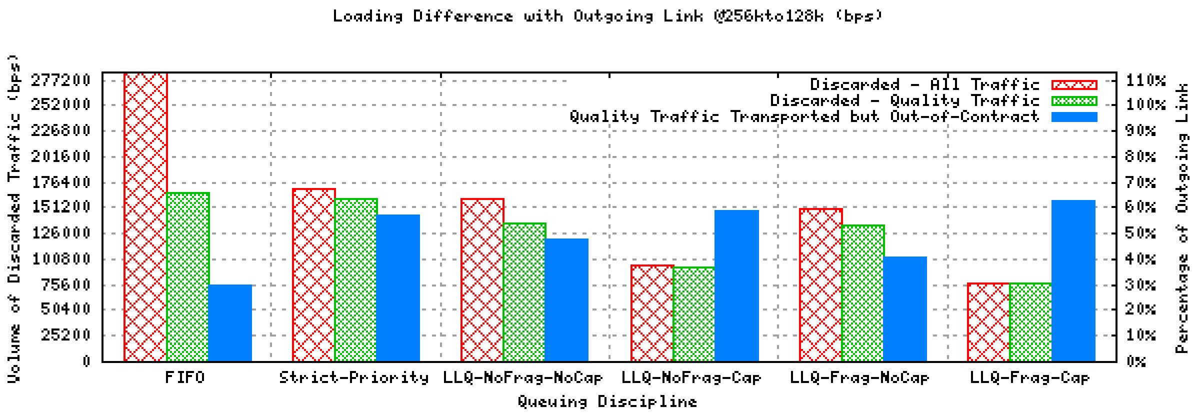

Transported Load with Outgoing Link @256kto128k (bps)
Volume of Outgoing Traffic (bps)
252000
226800
201600
176400
151200
126000
100800
75600
50400
25200
0
All Traffic
Quality Traffic
In-Contract Traffic
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Percentage of Outgoing Link
FIFO
Strict-Priority
LLQ-NoFrag-NoCap
LLQ-NoFrag-Cap
LLQ-Frag-NoCap
LLQ-Frag-Cap
Queuing Discipline

A.2.2.3 Load Difference









Loading Difference with Outgoing Link @256kto128k (bps)
Volume of Discarded Traffic (bps)
277200
252000
226800
201600
176400
151200
126000
100800
75600
50400
25200
0
Discarded - All Traffic
Discarded - Quality Traffic
Quality Traffic Transported but Out-of-Contract
110%
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Percentage of Outgoing Link
FIFO
Strict-Priority
LLQ-NoFrag-NoCap
LLQ-NoFrag-Cap
LLQ-Frag-NoCap
LLQ-Frag-Cap
Queuing Discipline

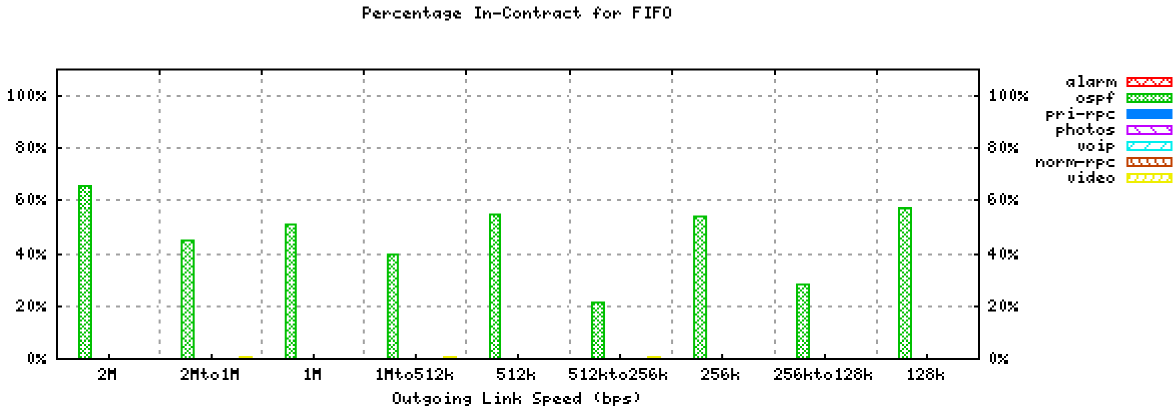
A.2.3 Per App Percentage In-Contract by Queuing Discipline

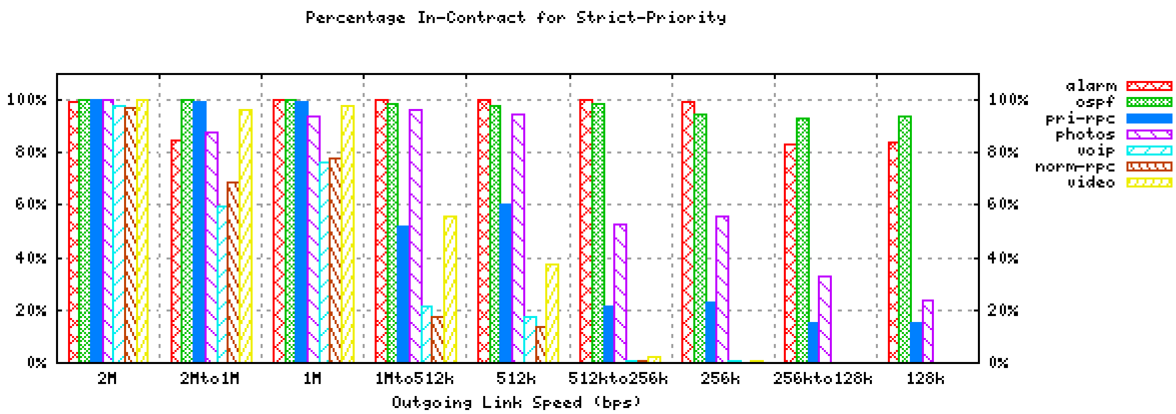
These graphs view the data from an application's perspective (as opposed to a total volume perspective in the graphs above). The individual, per application, bars capture what fraction of the time the application's traffic was delivered within the per-hop quality degradation budget. The graphs are per-queuing mechanism with all of the link speed variations on a single graph.

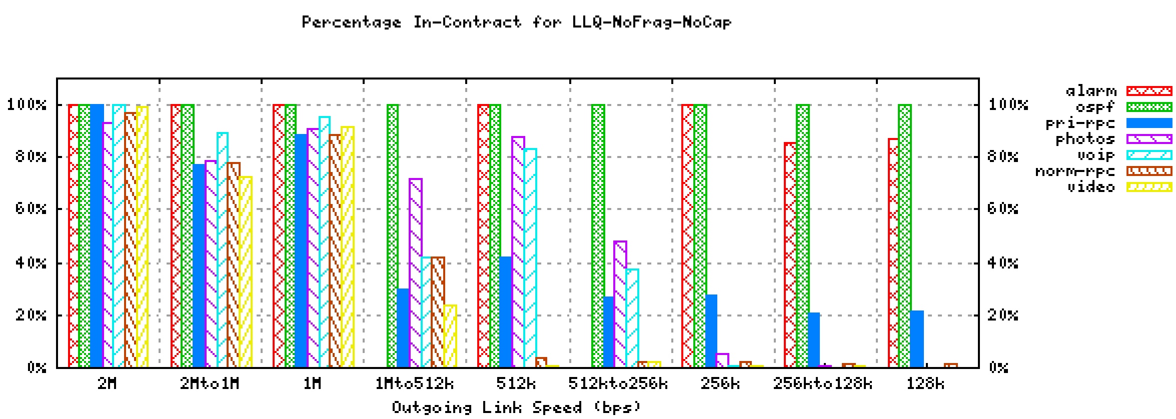
From these graphs the actual order in which the applications failed to be delivered quality, as the egress link speed was reduced, can be observed[3].

[3] The absence of in-contract alarm traffic for LLQ when fragmentation was off at 1M to 512K and 512K to 256K was noted during analysis. By inspection of the underlying traffic traces for the constituent runs it was determined that the packets for the alarm class were being lost/delayed beyond the transition period. The underlying cause has been left for further investigation. This phenomenon did not occur when fragmentation was enabled

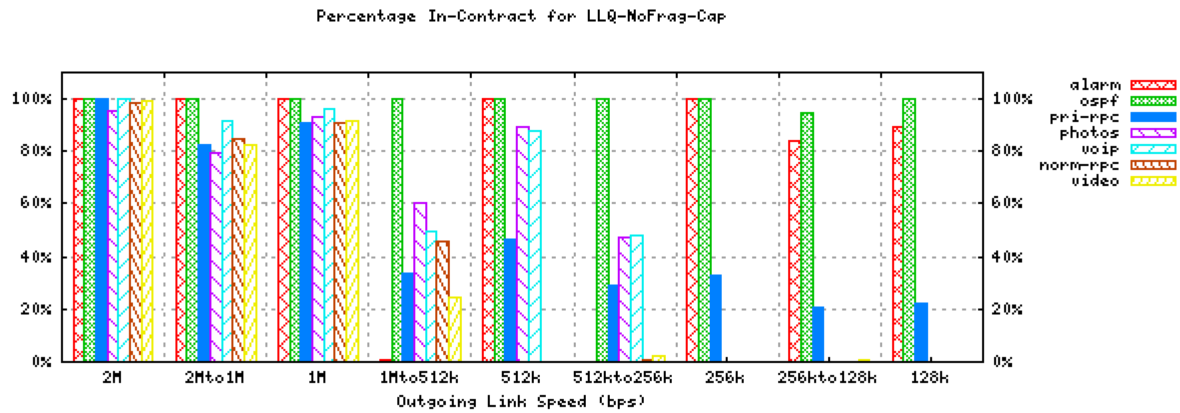
A.2.3.1 In-Contract



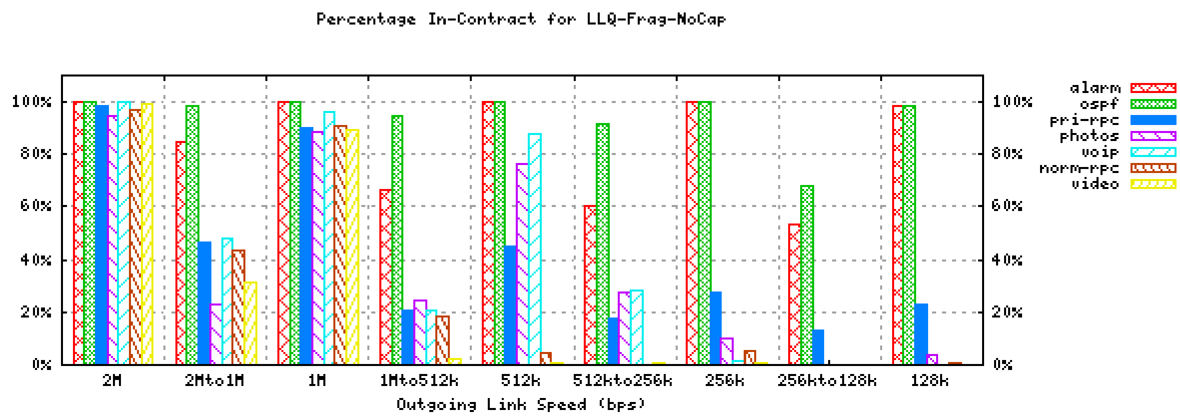


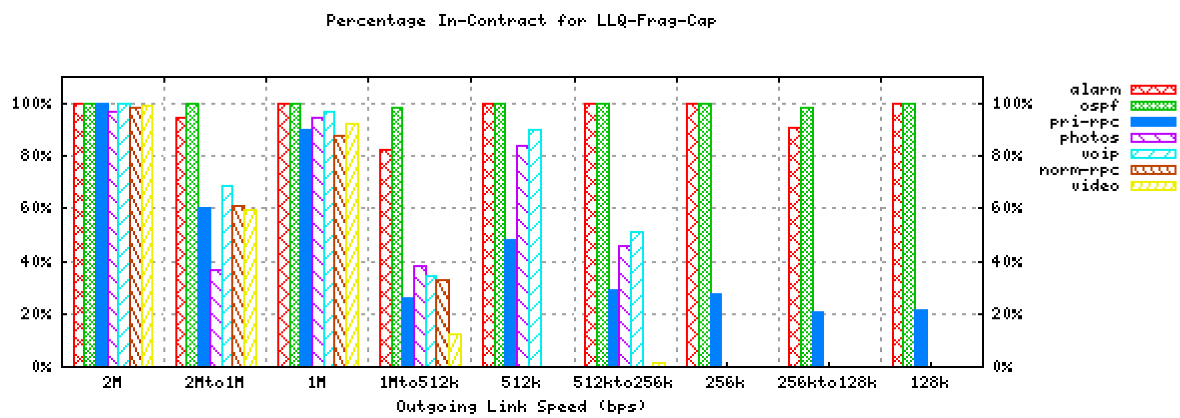


See Footnote 3 on page 93 regarding alarm traffic.

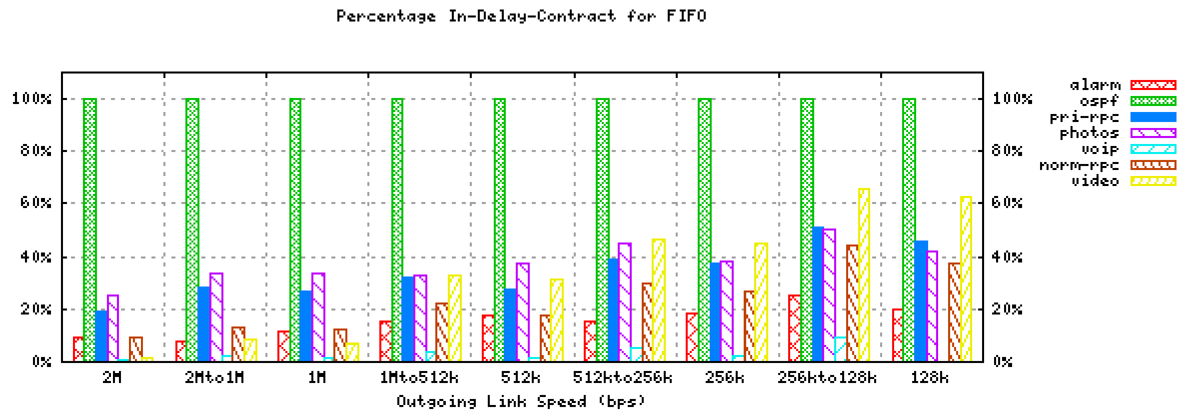


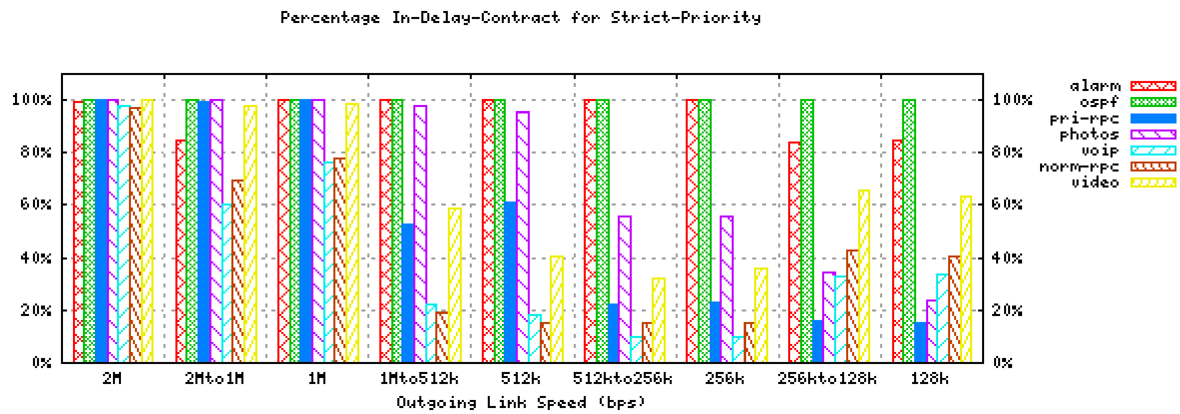
See Footnote 3 on page 93 regarding alarm traffic.

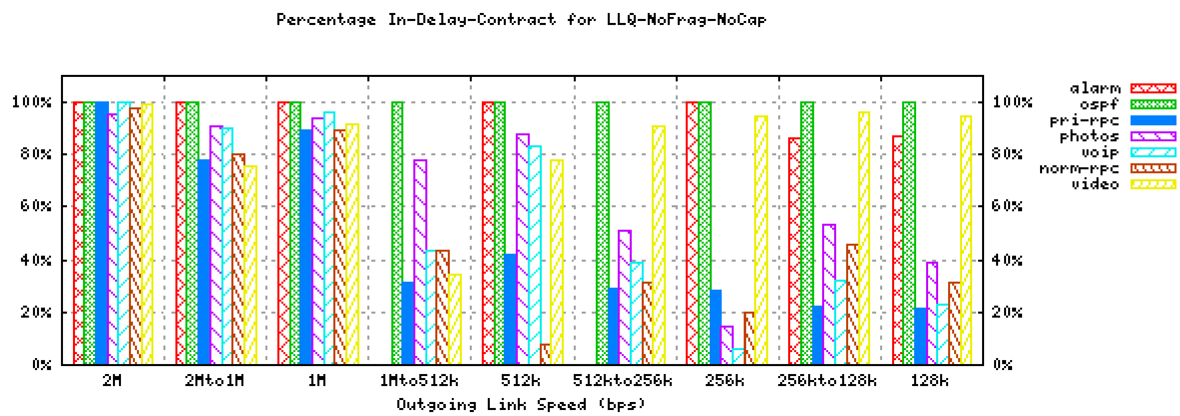




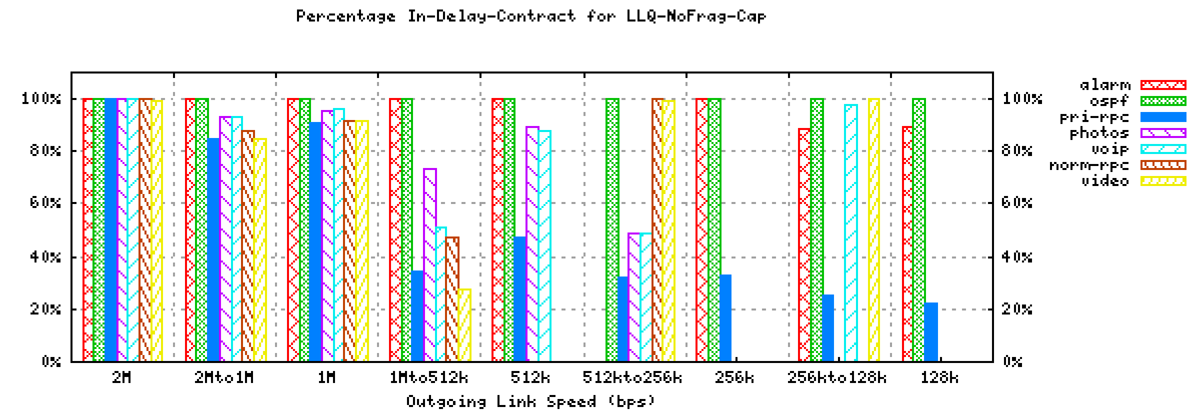
A.2.3.2 In-Delay Contract



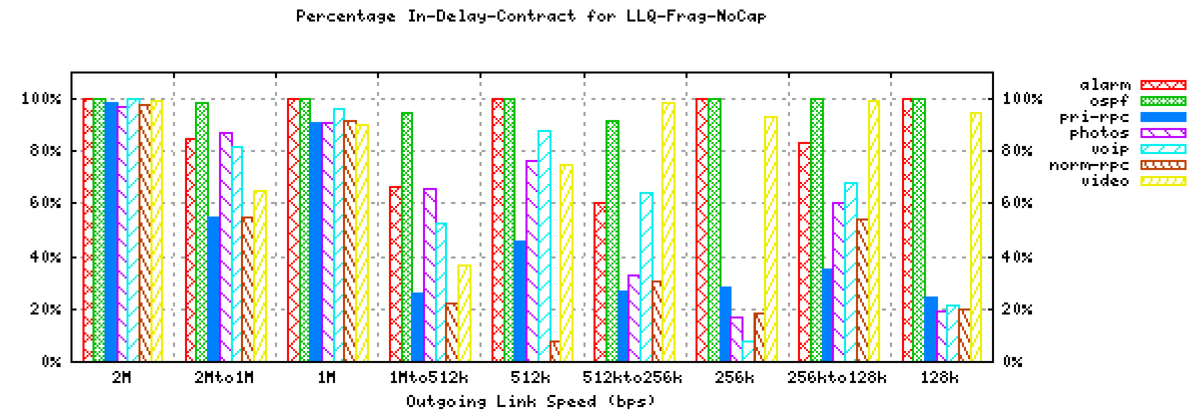


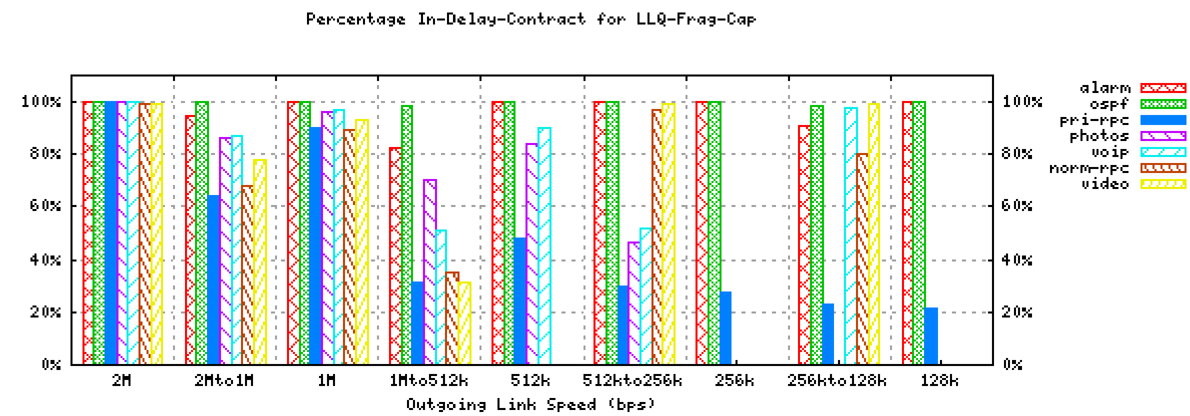


See Footnote 3 on page 93 regarding alarm traffic.

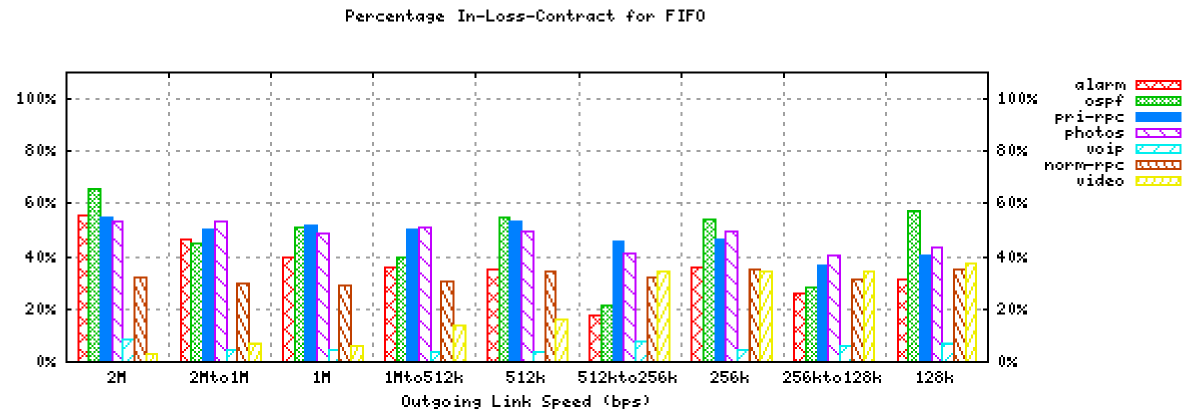


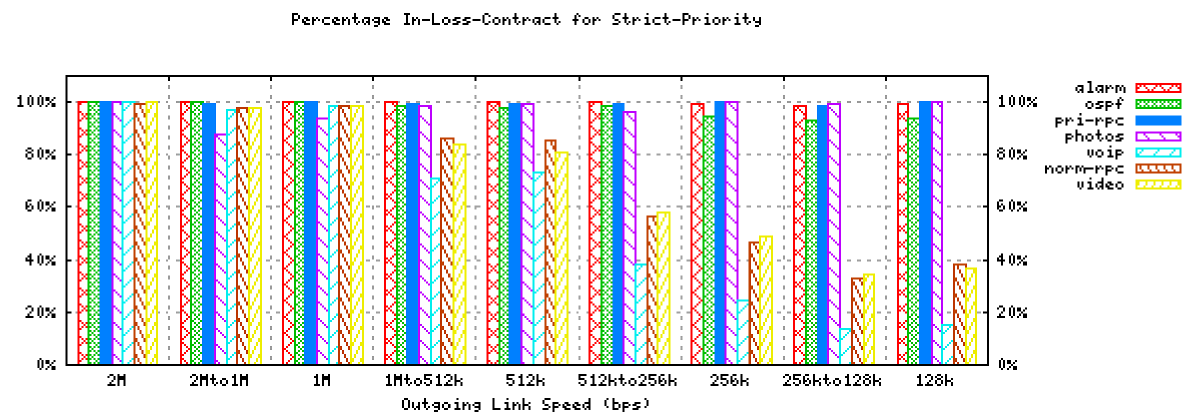
See Footnote 3 on page 93 regarding alarm traffic.

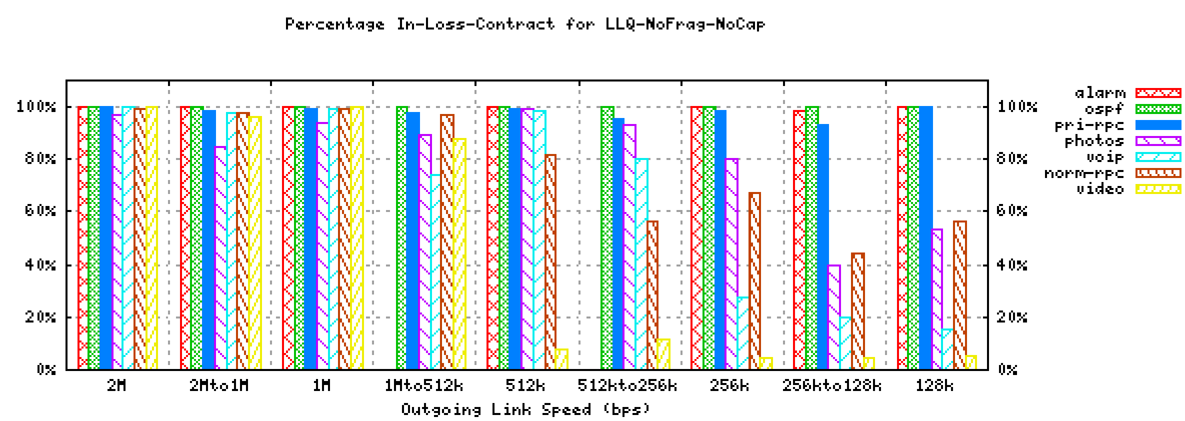




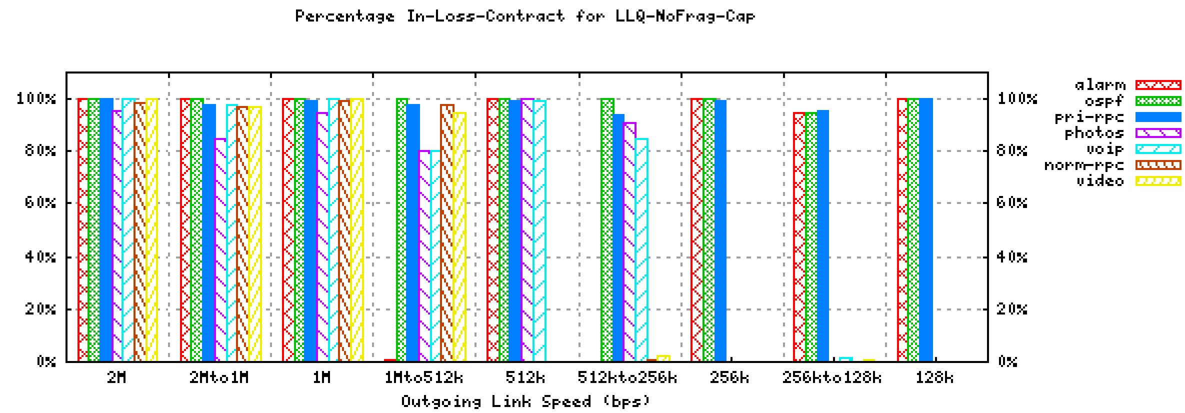
A.2.3.3 In-Loss Contract



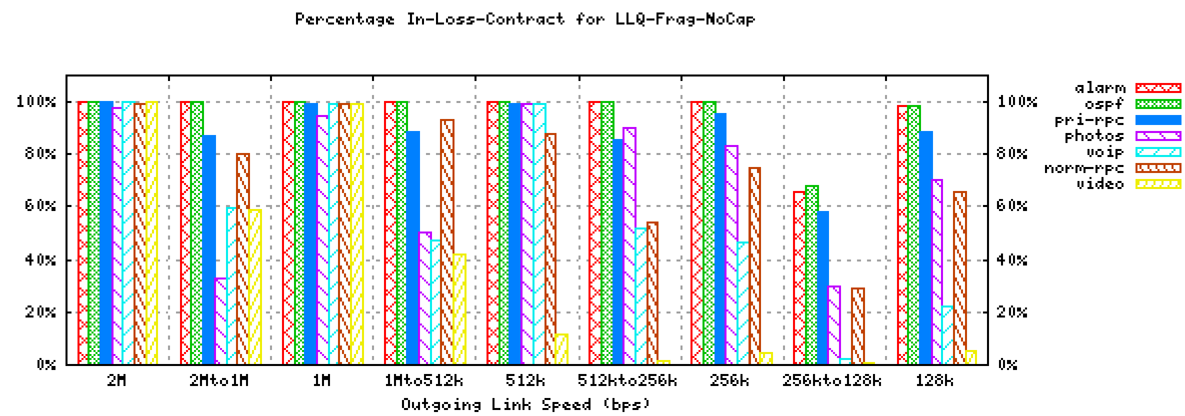


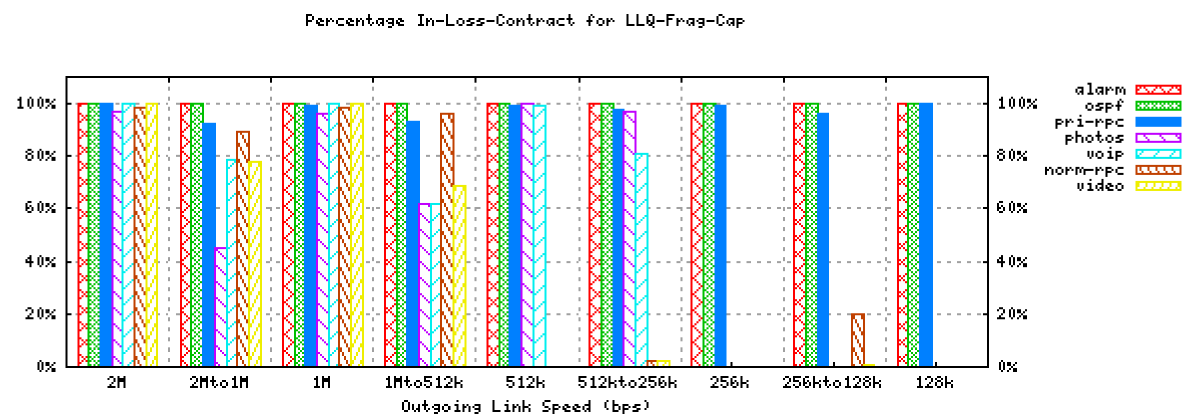


See Footnote 3 on page 93 regarding alarm traffic.



See Footnote 3 on page 93 regarding alarm traffic.





Appendix B Acquiring the Dataset

A copyrighted data set of the captured packet traces along with the analysis results for all traffic run is available in DVD format. Please send an email request with your contact details including:

- Name,
- Company,
- Mailing address. and
- Phone number(s).

Send your email to:

- Bob Francis, Cisco Systems, Inc. (bfrancis@cisco.com) or
- Fred Hammond, Predictable Network Solutions, Inc (fred.hammond@pnsol.com).

In your email use a subject line that reads as follow:

- "DVD Request for Emergent Properties of Cisco QoS Mechanisms."

Emergent QoS Properties: Traffic Model

David C. Reeve
<David.Reeve@pnsol.com>

6th May 2005

Abstract

This document describes a set of test traffic, its constraints, bandwidth requirements and behaviour. The traffic is used to evaluate the emergent properties of a number of queueing and scheduling algorithms. It is one of many documents that have been produced as the result of a collaboration between Boeing Phantom Works, Cisco Systems, and Predictable Network Solutions.

1 Application Analysis Approach

This section describes the applications that are present on the network. The applications and their usage are synthetic; however, they have been selected to highlight the kind of constraints under which traffic may behave in a wireless arena. For each application, we give a brief description of the application's purpose in an attempt to make the situation more realistic. Then we concentrate on calculating the per-instance quality constraints.

For each of the applications we outline the amount of quality degradation that an application's packet exchanges can experience and still deliver an acceptable outcome to the user. We use this concept of quality degradation frequently and refer to it as ΔQ[1]. Our focus is on the treatment (bythe individual network elements and hence the end-to-end network) of the application's generated packets; we are not, in this analysis, considering application processing costs.

Given an end-to-end ΔQ budget[2] for an application we can derive a per-hop ΔQ budget. This is required as the packets from an application may traverse more than one hop. Thus, by configuring each hop to deliver a quality degradation that is no worse than the application's

[1] Think of ΔQ as the "change in quality".

[2] The budget is the limit on the quality degradation for this application's data packets, in terms of their experienced loss and delay.

per-hop ΔQ budget, the total quality degradation experienced by the application's packets (as it sums over several hops) will be less than the associated application's end-to-end ΔQ budget. By receiving this "quality" treatment for its packets an application will deliver an acceptable outcome to the user.

The purpose of this analysis is to define the end-to-end ΔQ, and hence, for a given number of hops, the per-hop ΔQ budget. Measuring the delivered per-hop ΔQ to a collection of application traffic flows, initially at a single element, is the aim of this work.

2 Applications

The individual applications will have different restrictions on the total number of network hops they can sucessfully operate over. In this analysis we have made the assumption that the network is made up of identical elements each being allocated their share of the ΔQ budget.

2.1 Alarm

The alarm application is intended to convey some sort of emergency signal to a number of subscribers. This could be for example a fire alarm; when activated the klaxons will receive the signal and start sounding. Clearly this kind of application will be rarely used; however, we will ensure that it is used at least once in the tests. The alarm application must work over 10 hops, this is to ensure that an emergency signal can be propagated over a network with a potentially large diameter.

To achieve acceptable levels of reliability, the alert will be repeated every 10th of a second, for one second. We are assuming that this application is UDP based, and has a total packet size of 256 bytes. The constraint is that at least one, out of the ten, packets is received in 2s, 99.9% of the time. The calculation, in outline, is as follows:

For there to be a failure, 10 packets in a row must be lost, the probability of the occurence of which must be less than 99.9%. The probability of losing a packet end-to-end is therefore:

$$1 - P_{e2e}^{10} \leq 99.9\% \therefore P_{e2e} \leq 50\%$$

If we then assume that this same process has to work over a maximum number of 10 hops, we can work out the loss rate for a single hop. In order for the application to be successful, transmission will have to succeed for 10 hops in a row. Thus the loss probability per hop is:

$$(1 - P_{hop})^{10} \leq 50\% \therefore P_{hop} \leq 6\%$$

Given that the application has to work over 10 hops, and we have to succeed in 2s, the maximum diameter of the network cannot exceed 1s, as the last packet in the 1s burst still has to be able to sound the alarm. Therefore the delay per-hop is 100ms.

Summary:

End-to-end ΔQ Budget:	Delay: 1s, Loss: 50%
Number of Hops:	10
Per-hop ΔQ Budget:	Delay: 100ms, Loss: 6%

2.2 OSPF

The routing protocol OSPF will be used between routers in this network. OSPF is a relatively complex protocol to model; for the purposes of this experiment, we will assume that the network is stable. This means that there will be no link state updates for the duration of the tests. As such we can assume that the only communication happening on the network will be the Hello packets between the routers.

Assuming a hello time of 10s and a dead time of 40s, the delay would need to be less that 40s. 10s would seem adequate as a lower bound constraint. If four packets in a row are lost then a failure will occur. Given that we want to achieve a 99.999% success rate then we can calculate the loss as follows[3]:

$$1 - P_{loss}^4 \leq 99.999\% \therefore P_{loss} \leq 5\%$$

This protocol shall run continuously throughout the test. It will generate packets of at least of 64 bytes in size.

Summary:

End-to-end ΔQ Budget:	Delay: 10s, Loss: 5%
Number of Hops:	1
Per-hop ΔQ Budget:	Delay: 10s, Loss: 5%

2.3 PTT VoIP

The users of this wireless network use Push-To-Talk (PTT) Voice-over-IP (VoIP) to communicate. This type of VoIP requires no control traffic; it simply passes the audio data

[3]This is not quite how we would do this for real, the target should be 99.999% over an operating day. This can be solved using a 5 state Markov chain, where the last state is absorbing with the probability of entering it no higher than 0.001 within one day.

between the parties. This application uses the G.729 codec [2], which generates packets at a rate of 12.5pps. The packet size is 170 bytes, where 112 bytes are data, 30bytes RTP, and 28 bytes are IP overhead. This means the resulting data rate is 17Kbps.

For the quality of the VoIP to be acceptable, it must have a PESQ score of greater than 3.0, which is considered acceptable by most users. Using figure 1, which was obtained from [4], we can see that, to achieve the target, we require an end-to-end loss of less than 5%, and a jitter of less than 30ms. Additionally the delay must be less than 150ms (assuming a 100ms jitter buffer).

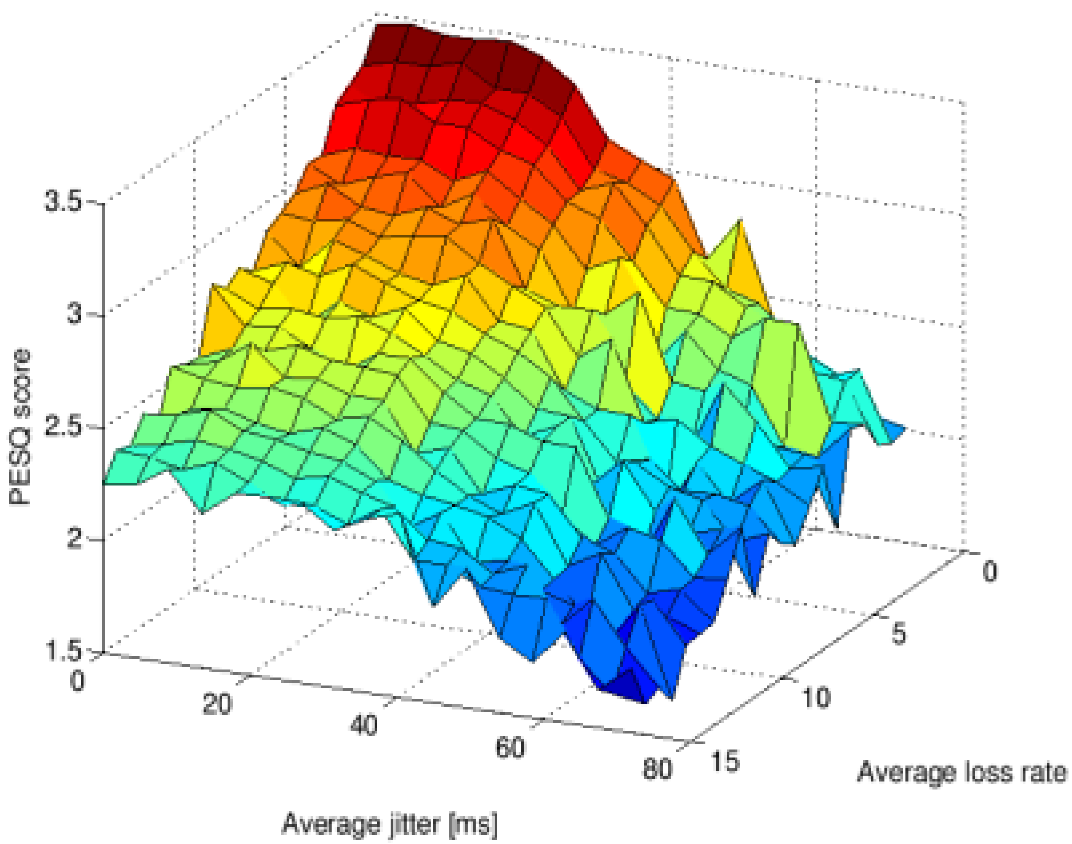


Figure 1: G.729 PESQ score vs. loss and jitter

We are assuming that this activity has to be supported over no more than 3 hops. As the delay and jitter are additive quantities we can dived the end-to-end budget by the number of hops. Clearly the per-hop budget could also be split in other ways, so long as the total does not exceed the end-to-end budget. In this case we are treating all three hops are having equal responsibility for delivering the quality.The per-hop budget is therefore 10ms for Jitter and 50ms for delay. The loss per-hop can be calculated as follows:

$$(1 - P_{loss})^3 \leq 1 - 5\% \therefore P_{loss} \leq 1.5\%$$

Summary:

End-to-end ΔQ Budget:	Delay: 150ms, Jitter: 30ms, Loss: 5%
Number of Hops:	3
Per-hop ΔQ Budget:	Delay: 50ms, Jitter: 10ms, Loss: 1.5%

2.4 Photo Capture

Video cameras placed around the network take still photos of their environment every 30s. The cameras take photos at QCIF resolution (144x176) at 24bps. Each of the frames is compressed at a ratio of 15:1, resulting in 5k per frame, which is 4 packets of 1300 bytes. The photos have to reach their destination within an average of 3s.

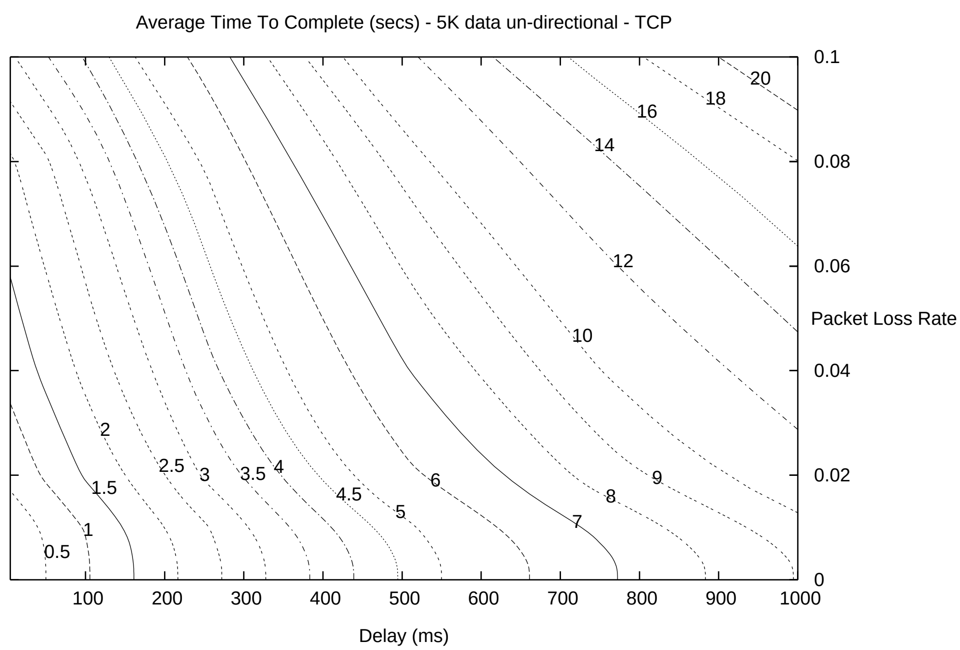


Figure 2: Contour of still photo transfer time

Figure 2 shows a contour plot of the time to complete a 5k transfer using TCP for a matrix of loss and delay values. This data was obtained by simulation using ns2 [1]. To satisfy the requirement to transfer the frames in an average of 3s, over a single hop, the loss should be <8%, and the delay <100ms. This can be found by drawing a box constructed of, left of the line of 100ms and below the line of 8% loss; the contours in this region are less than 3s.

Summary:

End-to-end ΔQ Budget:	Delay: 100ms, Loss: 8%
Number of Hops:	1
Per-hop ΔQ Budget:	Delay: 100ms, Loss: 8%

2.5 Streaming Video

In addition to the still photo capture, presented above, it is possible to stream video over a single hop. This comprises of a maximum of 30fps of QCIF 24bps video. The stream compression achieves a compression ratio of about 30:1. At 30fps this results in a data rate of 608Kbps, with a packet size of 1500 bytes.

To cope with changing conditions, frame rate will drop as the loss increases; the steps are: 30, 15, 10 and 5 fps. This results in the following data rates at those steps: 608, 204, 202, and 101Kbps.

Given that video is uni-directional, a large delay can be tolerated, in the order of a second. The jitter must be less than 250ms, as beyond this the size of the de-jittering buffer becomes an issue. The loss, to achieve less than 5% frame error rate, must be less than 1%; see [3].

Summary:

End-to-end ΔQ Budget:	Delay: 1s, Jitter: 250ms, Loss: 1%
Number of Hops:	1
Per-hop ΔQ Budget:	Delay: 1s, Jitter: 250ms, Loss: 1%

2.6 Web Application

Users of the network have access to an HTTP-style web-based application. This application makes RPC style requests to a server located no more than 3 hops away. Each request is no bigger than 500 bytes in size, and the responses are broadly negatively exponentially distributed with a mean size of 10k bytes. Clients can request information in one of two modes, normal, or priority, the selection of which changes the time to complete.

Figure 3 shows the arithmetic mean and 95th percentile time to complete the RPC transaction. These results were again generated using simulations with ns2, and are therefore subject to some error.

For the normal priority traffic, the requirement is to complete 50% of the transfers within 5s and 95% within 8s. This requires an end-to-end loss and delay of 2% and 300ms respectively. For a single hop, this equates to a loss of 0.6% and a delay of 100ms.

For the high priority traffic, the requirement is to complete 50% of the transfers in 2s, and 95% within 4s. This requires an end-to-end loss and delay of 1% and 200ms respectively. For a single hop this equates to a loss of 0.3% and a delay of 65ms.

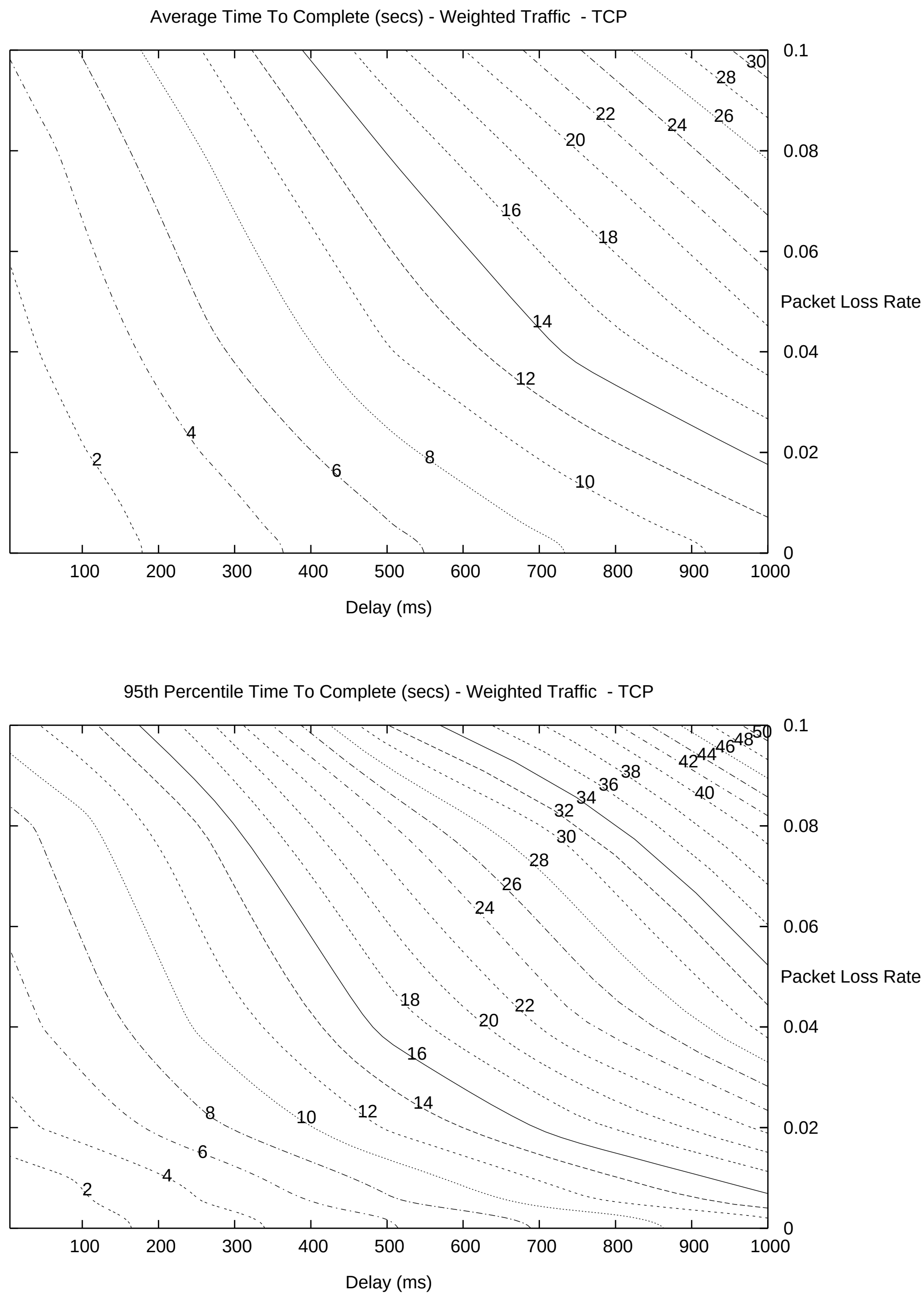


Figure 3: RPC time to complete

Summaries

For the normal priority RPC:

End-to-end ΔQ Budget:	**Delay: 300ms, Loss: 2%**
Number of Hops:	**3**
Per-hop ΔQ Budget:	**Delay: 100ms, Loss: 0.6%**

For the high priority RPC:

End-to-end ΔQ Budget:	**Delay: 200ms, Loss: 1%**
Number of Hops:	**3**
Per-hop ΔQ Budget:	**Delay: 65ms, Loss: 0.3%**

3 Requirements

Table 1 shows a summary of the requirements, and figure 4 shows the applications laid out on a two dimensional loss/delay grid. Table 2 shows the average number of instances of each application, and the load that this creates on the network.

We have assumed that the frames will be encoded using HDLC, with an overhead of 7 bytes, which has been Incorporated in the offered load calculation. It is assumed that this link is the primary contention point for the network. The offered load consumes approximately 50% of the capacity of the link at 2Mbps, the remainder being consumed by background traffic. At 1Mbps it is expected that the background traffic will shrink, and that most applications will work. As the capacity shrinks the applications should fail in the following order (top fails first):

1. Video
2. Normal RPC
3. VoIP
4. Photos
5. Priority RPC
6. OSPF
7. Alarm

Application	Loss	Delay	Jitter
Alarm	6%	75ms	-
OSPF	5%	10s	-
VoIP	1.5%	50ms	10ms
Video	1%	1s	250ms
Photos	8%	100ms	-
Normal RPC	0.6%	100ms	-
Priority RPC	0.3%	65ms	-

Table 1: Summary of traffic requirements

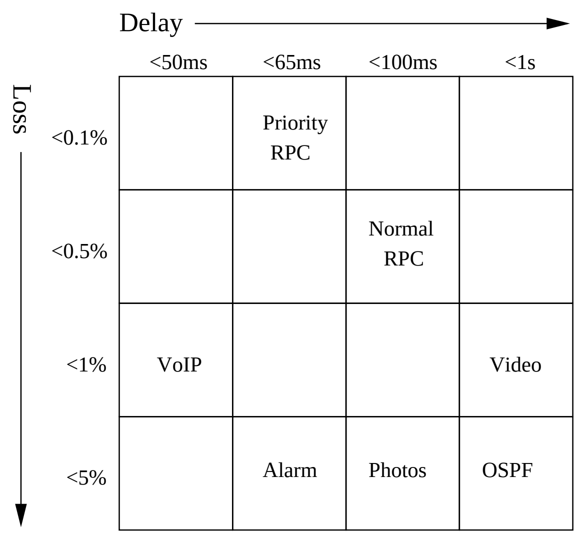


Figure 4: Loss/Delay Grid

Application Name	Packet Size (bytes)	Packet Rate (pps)	No. Instances	Data Rate (bps)
Alarm	256	10.0	1	21K
OSPF	64	0.1	1	57
VoIP	170	12.5	8	142K
Video	1500	50.0	1	603K
Photos	1300	1.3	5	68K
Normal RPC	1500	1.4	8	135K
Priority RPC	1500	3.5	1	42K

Table 2: Offered Load

References

[1] Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, Ya Xu, and Haobo Yu. Advances in network simulation. *IEEE Computer*, 33(5):59–67, May 2000.

[2] ITU-T Recommendation G.729. Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP), March 1996.

[3] Steven Gringeri, Roman Egorov, Khaled Shuaib, Arianne Lewis, and Bert Basch. Robust compression and transmission of mpeg-4 video. In *ACM Multimedia (1)*, pages 113–120, 1999.

[4] Mihai Ivanovici Razvan Beuran. User-perceived quality assessment for VoIP applications. Technical report, CERN, January 2004.