Naval Research Laboratory
Washington, DC 20375-5320

Evaluation of OpenROUTE Networks Bandwidth Reservation System and Priority Queueing

VINCENT D. PARK
JOSEPH P. MACKER
STEVE J. SOLLON

Protocol Engineering and Advanced Networking (Protean) Research Group
Communication Systems Branch
Information Technology Division

August 31, 1999

Executive Summary

INTRODUCTION

We performed a series of network router tests to evaluate the Bandwidth Reservation System (BRS) and priority queueing currently available in OpenROUTE Networks routers. The evaluation was intended to assess the potential of these router capabilities to support improved network bandwidth management in the Navy's Advanced Digital Network System (ADNS) architecture. Internet Protocol (IP) router products from OpenROUTE Networks with software versions 3.0 and above include optional BRS and priority queueing services. BRS and priority queueing services promise to provide the capability to better manage which packets are dropped at a router's outbound interface under congestion. Within the existing NRL network router testbed, we were able to control network loading and congestion characteristics in a very predictable and detailed manner using NRL-developed network traffic test tools (e.g., MGEN). Using these traffic generation and analysis test tools we performed a series of experiments using the OpenROUTE Networks router. In general, the experiments focused on the use of this technology across moderate rate wireless or satellite communications links (e.g., 64 kbps).

FINDINGS AND RECOMMENDATIONS

In a first series of experiments, we evaluated the router's ability to allocate and guarantee minimum percentages of the transmission capacity to specified classes of traffic. This can be thought of as a guaranteed minimum bandwidth service. The first series of tests yielded largely positive results with some notable exceptions. Managed bandwidth percentages for separate network traffic classes were effectively demonstrated under dynamically stressed traffic conditions. In addition, unused bandwidth allocations for classes of traffic were shown to be dynamically shared by network traffic from other classes. Under testing with unequal percentage bandwidth allocations (e.g., 25% and 75%), our initial test results indicate that traffic in the higher percentage class receives slightly more bandwidth than the configured allocation, while traffic in the lower percentage class receives slightly less. This percentage discrepancy appears to be within a few percentile points of the expected value (i.e., less than approximately 5% was observable). Follow-on tests included variation of the percentage allocations, traffic loads, packet sizes (100 - 1000 bytes) and link rates (64 - 500 kbps) to verify that this was not an artifact from our testing scenario. We conclude that the guaranteed minimum bandwidth service within OpenROUTE 3.0 works reasonable well with the exception of some small allocation discrepancies noted in our trials.

In a second series of experiments, we evaluated the router's ability to perform priority queueing services. In these experiments, different network traffic flows were assigned to different priority levels (i.e., LOW, NORMAL, HIGH and URGENT) without minimum bandwidth allocation distinctions. By definition, the servicing of higher priority packets should be absolute. Higher priority packets should always gain precedence for transmission over lower priority packets. Some initial simple tests of priority queueing, using two different priority levels simultaneously, revealed expected results; higher priority traffic received absolute precedence over lower priority traffic for all single combination pairings of the different priority levels. However, further testing of priority queueing, using more than two priority levels simultaneously, yielded some unexpected results. As an example, one test, using LOW, NORMAL, and HIGH priority levels simultaneously, revealed that NORMAL and HIGH priority traffic achieved about the same throughput, with HIGH priority traffic receiving much less servicing than expected. In another test, using all four priority levels simultaneously, the URGENT, HIGH and NORMAL priority traffic appeared to share the link capacity equally, while the throughput of the LOW priority traffic was reduced to near zero. Once again, the expected behavior would be for the highest priority traffic to dominate and for traffic with lower priorities to receive leftover bandwidth in an absolute descending service order fashion. Thus, our experiments have revealed a potential flaw within the priority queueing model of the OpenROUTE 3.0 software.

Finally, a third series of experiments evaluated the combination of minimum bandwidth allocations and priority queueing. Several classes of guaranteed minimum bandwidth were defined and traffic flows were assigned to these classes. Within each guaranteed bandwidth class, traffic flows were further subdivided by priority. While the aggregate traffic classes roughly received their expected bandwidth allocations as predicted and indicated by the first test series, combinations of priority services once again revealed problems as discovered in the second set of experiments discussed above. We conclude that the observations of test series 1 and 2 carry over into the hybrid operation as examined in test series 3.

In conclusion, we have extensively tested a number of features and operational modes of the OpenROUTE 3.0 BRS and priority queueing. Our findings indicate that under dynamic conditions and moderate link data rates, BRS appears to function reasonably well as a pure guaranteed minimum bandwidth service for aggregate traffic classes. The software appears robust and mature enough to deploy safely within the ADNS architecture (with the noted discrepancies discovered). Further, we discovered problems with the priority queueing service, which indicate it does not function as anticipated when servicing more than two simultaneous priority levels. Until these problems are fixed, we do not recommend reliance on the pure priority traffic handling features of OpenROUTE 3.0. These problems can be likely fixed and/or improved in future releases of software.

Evaluation of OpenROUTE Networks Bandwidth Reservation System and Priority Queueing

INTRODUCTION

This report outlines a series of tests performed to evaluate the Bandwidth Reservation System (BRS) and priority queueing currently available in OpenROUTE Networks routers. The evaluation was intended to assess the potential of these router capabilities to enhance Quality of Service (QoS) support and network traffic management in the Navy's Advanced Digital Network System (ADNS) architecture. The tests were conducted using a testbed, constructed at the Naval Research Laboratory (NRL), consisting of available COTS network hardware components and computer workstations. Data production, collection, and reduction were performed using NRL-developed specialized network testing software. Initial test results are presented in detail and performance issues are discussed.

OVERVIEW OF THE BANDWIDTH RESERVATION SYSTEM

Internet Protocol (IP) router products from OpenROUTE Networks with software versions 3.0 and above include BRS and priority queueing. BRS and priority queueing provide the capability to manage which packets are dropped at an outbound interface when the offered load exceeds the throughput (i.e., when the interface becomes congested). Based on the available documentation, the BRS queueing mechanism appears to be a single-level implementation of class-based queueing with the ability to designate the priority levels of traffic within a given class, Fig. 1. Thus, the outbound traffic on a given interface can be segregated into different classes of traffic, and each such class can be allocated a minimum percentage of the transmission capacity. When the offered load of traffic from a given class is less than the minimum percentage allocated for that class, the available capacity can be dynamically shared by traffic from other classes. Within a given class, traffic can be further segregated into four priority levels (i.e., low, normal, high, and urgent). The servicing of packets within a class is described as classic priority queueing, where packets of the highest priority level are always serviced first. If there are no packets of the highest priority level, packets of the next highest priority level are serviced, and so on. It is unclear whether the priority levels also play a roll in the dynamic sharing of unused allocations across aggregate traffic classes.

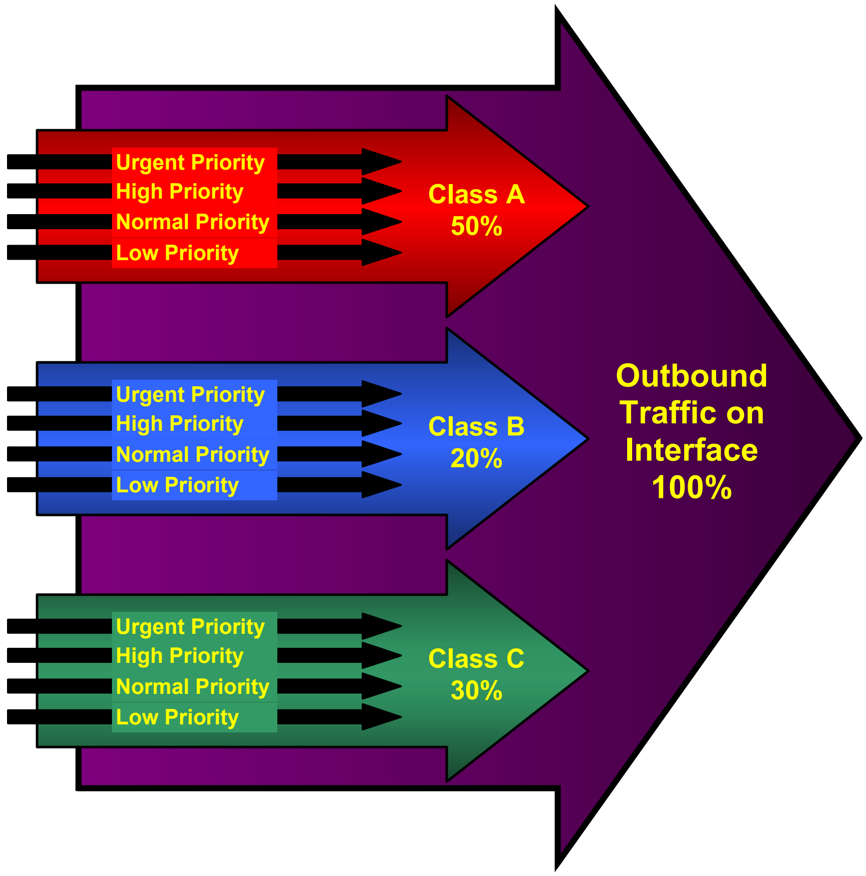

Fig. 1 — Conceptual illustration of transmission capacity percentage allocations and priority queueing in the Bandwidth Reservation System

TEST METHODS

A testbed was established at NRL for experimentation and evaluation of Internet technology. The testbed was designed to facilitate rapid reconfiguration and allow for future growth. In order to evaluate BRS and priority queueing in OpenROUTE Networks router products, the testbed was augmented with the addition of an OpenROUTE Networks GTX-1000 router. While this is not the exact router model presently being deployed as part of the ADNS architecture, it utilizes the same software and thus provides a good representation of the capabilities under evaluation.

Testbed Configuration

The testbed configuration was essentially designed to model an internetwork where multiple high-speed LANs are connected via a moderate-rate (bottleneck) link. As shown in Fig. 2, the configuration consisted of two Ethernet segments connected through two IP routers and a moderate-rate link (64 kbps) using the Point-to-Point Protocol (PPP). End systems (i.e. computer workstations) on the Ethernet segments were used to generate and log network traffic. During many of the test runs the traffic source produced congestion levels well above the capacity of the bottleneck link to test the queueing mechanisms under stressed conditions.

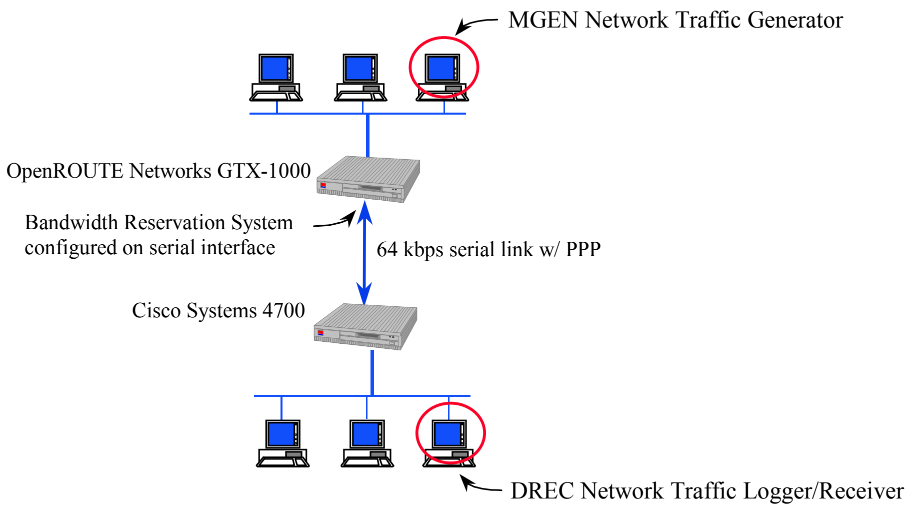

Fig. 2 — Testbed configuration

Test Tools

While the TCP transport mechanism provides the majority of internetwork communications at present, multicast and unicast UDP traffic are anticipated to be important military network transport mechanisms. The reason behind this is the increasing demand for group data dissemination/collaboration tools, multimedia applications, and support for relatively short self-contained messages more suitable to UDP datagram delivery. For test purposes, the use of UDP traffic generators facilitates the analysis of test results by eliminating the statistical complexity of the TCP flow/congestion control mechanism. Therefore, this preliminary study focused primarily on the effects of the various queueing mechanisms on UDP data flows.

The NRL MGEN/DREC toolkit was used to generate UDP data flows and log end-to-end statistics. The MGEN/DREC toolkit provided the ability to produce accurate time scripted traffic loading from multiple traffic sources and capture the data from the multiple data flows at the receivers. In using these tools, we were able to control network loading and congestion characteristics in a very predictable and detailed manner.

Bandwidth Reservation System Configuration

BRS and priority queueing in the OpenROUTE Networks router were evaluated in three different configurations. The first configuration was designed to allow evaluation of the capability to allocate minimum percentages of the transmission capacity to specified classes of traffic, without the added complexity of priority queueing. Thus, BRS was configured with multiple classes and IP filters were defined to tag certain UDP data flows and assign them to the various classes, but all of the data flows were assigned the same priority level. The second configuration was intended to allow evaluation of the priority queueing mechanism. In this configuration, IP filters were defined to tag certain UDP data flows and assign them to different priority levels, but within this test series all of the separate data flows were assigned to a single class. To gain additional insight into how these mechanisms work together we conducted a test with a third configuration that combined minimum percentage class allocations and priority queueing within each defined class. BRS was configured with multiple classes and IP filters were defined to tag certain UDP data flows and assign them to a particular minimum bandwidth class and to indicate various priority levels within each class.

RESULTS AND DISCUSSION

The actual router settings of BRS and IP Filtering for these three test configurations are presented in Appendix A. For each of the router configurations, specifically designed MGEN test scripts were used to provide insight into the operation of queueing mechanisms. Some of the MGEN test scripts step through a series of different traffic conditions using two or more data flows. The individual data flows were generated by MGEN with different UDP port numbers to allow for identification and segregation within the router. The actual MGEN test scripts used for testing are presented in Appendix B.

Bandwidth Reservation

The first set of tests investigated the capability to allocate minimum percentages of the transmission capacity to specified classes of network traffic. To simplify interpretation of the results, the configuration did not include priority queueing within the specified classes. The original configuration of BRS when enabled on an interface includes two classes—LOCAL and DEFAULT. Although not well documented, the LOCAL class appeared to be reserved for traffic addressed to/from the router (e.g., traffic from a telnet session to the router to modify the configuration). While the percentage allocation could be modified, the LOCAL class could not be deleted from the configuration. The DEFAULT class could be either modified or deleted. For the subsequent tests an additional class was defined and labeled CRITICAL. The percentage allocation for the LOCAL class was reduced to 10%, while the allocations for the CRITICAL and DEFAULT classes were set to 60% and 30%, respectively. Based on the 64 kbps data rate of the bottleneck link, this corresponds to a minimum guarantee of approximately 38.4 kbps for traffic in the CRITICAL class and 19.2 kbps for traffic in the DEFAULT class. The actual throughput of IP traffic should be slightly less due to the overhead of lower-layer protocols (e.g., PPP).

Initially, two IP filters were defined—*test.critical* to mark (i.e., TAG=6) and pass all UDP packets with destination port numbers from 6000 to 6005, and *test.other* to simply pass any other packets. Packets marked with TAG=6 were explicitly assigned to the CRITICAL class in the BRS configuration. While the BRS configuration was such that all other packets should have been assigned to the DEFAULT class, initial tests showed that UDP packets with destination port 5000 were also being assigned to the CRITICAL class. Use of the Event Logging System (ELS) revealed that the UDP packets with destination port 5000 were correctly matched to the *test.other* IP filter, but the BRS packet counters indicated that the packets were incorrectly assigned to the CRITICAL class. A simple workaround was developed to allow further testing. The workaround entailed marking the packets matched to the *test.other* IP filter with TAG=5 and explicitly assigning packets with TAG=5 to the DEFAULT class in the BRS configuration.

Five different MGEN test scripts were used to evaluate BRS with this router configuration. Some tests were designed to investigate the basic functionality, while others were designed to test performance under specific conditions.

Test 1

Test 1 was designed to investigate the ability to allocate a minimum percentage of the transmission capacity and the ability to dynamically share unused capacity from another class. The test comprised two data flows—flow 1 packets were sent to port 6000 and assigned to the CRITICAL class, while flow 2 packets were sent to port 5000 and assigned to the DEFAULT class. The MGEN test script is best described as a series of distinct one-minute intervals during each of which the offered load of the individual data flows were held constant. During each one-minute interval, the offered load of a data flow can be described as either off, below its allocation or above its allocation and the link can be described as either congested or not. The MGEN test script proceeded through all combinations of the above conditions and also tested whether each data flow could acquire the entire link in the absence of competing traffic. A visual representation of the MGEN test script illustrating the offer load of the two data flows in test 1 is depicted in Fig 3, while the measured IP throughput is depicted in Fig 4.

The first and last minute of the test show that each data flow could acquire the entire link capacity (approximately 60 kbps) in the absence of competing traffic. During the sixth minute of the test (300-360 seconds), when both data flows were providing sufficient traffic to congest the link, the percentage allocations can be clearly seen. Flow 1 achieved a throughput of approximately 45 kbps, while flow 2 achieved a throughput of approximately 15 kbps. These throughputs differ slightly from the expected results. If we apply the configured percentages to the 60 kbps achievable throughput, the expected minimum percentage allocations would be 6 kbps for LOCAL, 36 kbps for CRITICAL (i.e., flow 1) and 18 kbps for DEFAULT (i.e., flow 2). Based on the available documentation, it is unclear how the unused LOCAL allocation should have been shared by flows 1 and 2—but even it is assumed that all of the LOCAL allocation was used for CRITICAL traffic (i.e., flow 1), the DEFAULT traffic (i.e., flow 2) failed to achieve its minimum guarantee by approximately 5%. Other intervals of the test illustrate the capability to share unused capacity from another class, both when the link is congested and when not. With the exception of the inaccuracy of the percentage allocations, this test successfully shows the ability to allocate a minimum percentage of the transmission capacity and the ability to dynamically share unused capacity from another class.

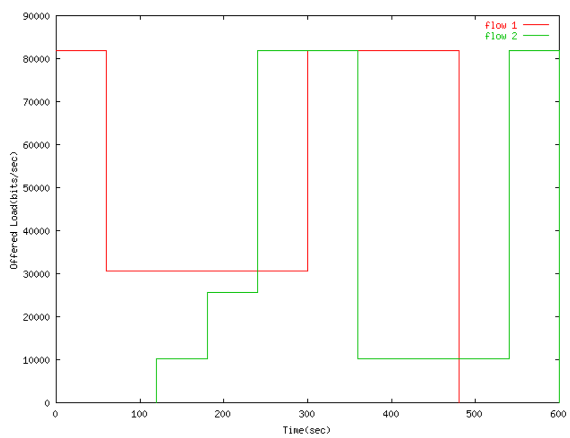

Fig. 3 — Test 1 offered load

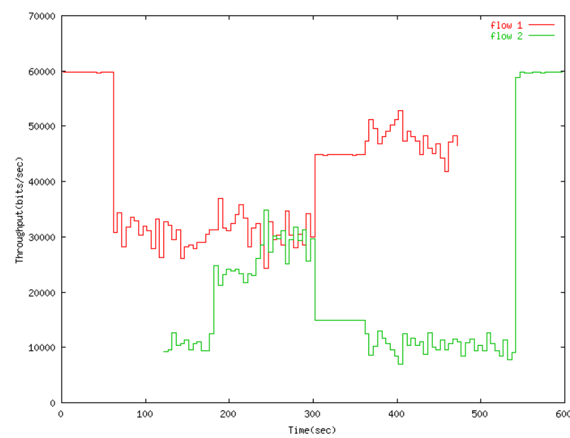

Fig. 4 — Test 1 throughput

Several additional tests were conducted to further investigate the apparent discrepancy in the percentage allocations. Follow-on tests included variation of the percentage allocations, traffic loads, packet sizes (100 - 1000 bytes) and link rates (64 - 500 kbps) to verify that this was not an artifact from our testing scenario. Under testing with unequal percentage bandwidth allocations (e.g., 25% and 75%), our initial results indicate that traffic in the higher percentage class receives slightly more bandwidth than the configured allocation, while traffic in the lower percentage class receives slightly less. This percentage discrepancy appears to be within a few percentile points of the expected value (i.e., less than approximately 5% was observable).

Test 2

Test 2 increased the complexity of the evaluation by including multiple data flows within each class. The intent was to provide insight into how multiple data flows within the same class share capacity when priority queueing is not used. Flow 1 packets and flow 2 packets were sent to ports 6000 and 6001, respectively, and thus were both assigned to the CRITICAL class, while flow 3 packets and flow 4 packets were sent to ports 5000 and 5001, respectively, and were assigned to the DEFAULT class. Again, the MGEN test script is best described as a series of distinct one-minute intervals during each of which the offered load of the individual data flows were held constant. The offered load and throughput of the four data flows in test 2 are depicted in Figs. 5 and 6.

During the first minute of the test, flows 1 and 2 provided traffic with a collective offered load that exceeded the capacity of the link. As expected, the throughputs achieved by flows 1 and 2 were proportional to their respective offered loads, and collectively they acquired the entire link capacity. During the second minute interval (60-120 seconds), flow 3 was added with an offered load that exceeded the capacity of the link. The throughput of flow 3 (DEFAULT class) and the collective throughput of flows 1 and 2 (CRITICAL class) were limited to the allocations documented in test 1. While the collective throughput of flows 1 and 2 was reduced, the individual throughputs remained proportional to the offered loads of the data flows. During the third minute interval, the offered load of flow 1 was reduced to be equivalent to the offered load of flow 2. Accordingly, the throughput of flow 1 was reduced and remained proportional to the offered load. These results are consistent with the use of first-in-first-out (FIFO) queueing for packets of the same priority within a given class; however, the details of the queueing mechanism are not publicly documented. Note that traffic for a given data flow was generated by MGEN using a Poisson distribution to avoid synchronization effects between the individual data flows. The later half of the test essentially repeats the first half, but with multiple data flows in the DEFAULT class.

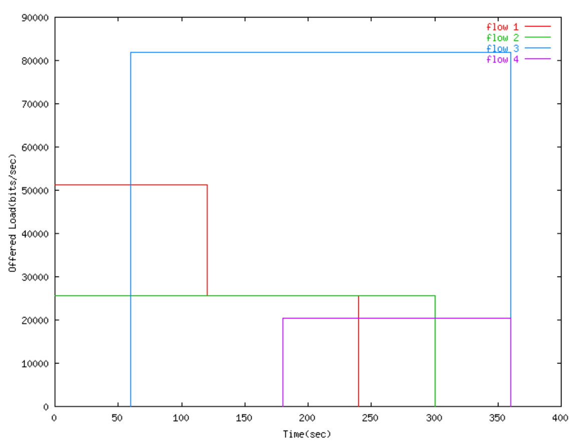

Fig. 5 — Test 2 offered load

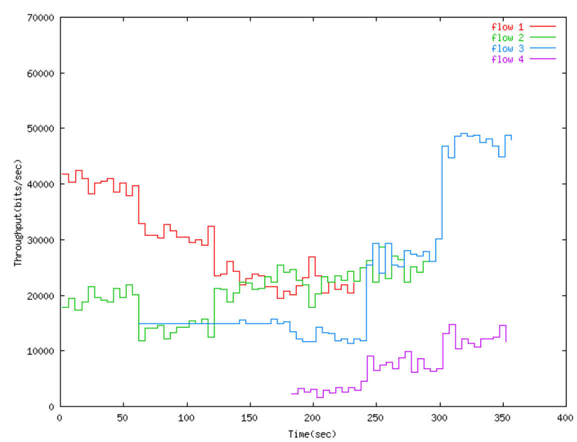

Fig. 6 — Test 2 throughput

Test 3

Test 3 was specifically designed to investigate whether the offered load of one class had any effect on the minimum percentage allocation (i.e., achievable throughput) of another class. To accomplish this the MGEN test script included two data flows—flow 1 assigned to the CRITICAL class and flow 2 assigned to the DEFAULT class. The offered load of flow 1 was set slightly *above* its minimum percentage allocation and held constant throughout the test. The offered load of flow 2 was initially set to approximately 25.6 kbps and then doubled in each successive one-minute interval. The offered load and throughput plots for test 3 are depicted in Figs. 7 and 8. When the offered load of flow 2 is the highest (i.e., during the last minute of the test) there appears to be a slight decrease in the throughput of flow 1 and a corresponding increase in the throughput of flow 2. Although the effect does not appear very significant, further investigation with greater offered loads for flow 2 may be warranted.

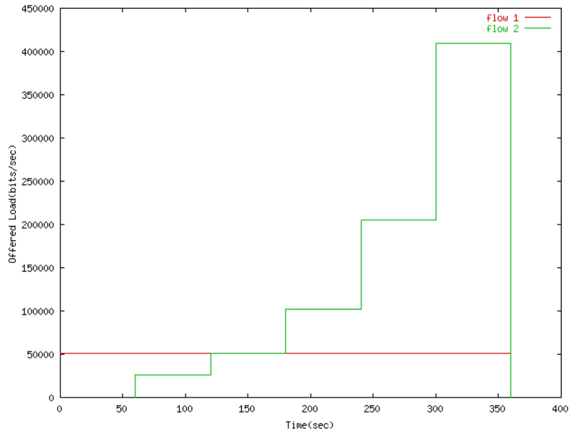

Fig. 7 — Test 3 offered load

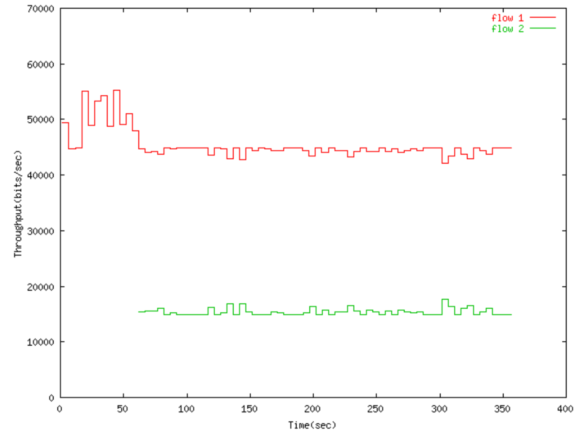

Fig. 8 — Test 3 throughput

Test 4

Test 4 was specifically designed to investigate whether the size of packets in one class had any effect on the minimum percentage allocation (i.e., achievable throughput) of another class. To accomplish this the MGEN test script included two data flows—flow 1 assigned to the CRITICAL class and flow 2 assigned to the DEFAULT class. In previous and subsequent tests, all packets were of the same size (i.e., 100 byte payload + 28 byte UDP/IP header). The payload size of packets in flow 1 was held constant at 100 bytes and the offered load was set slightly *below* its minimum percentage allocation (based on the results of prior tests). The payload size of packets in flow 2 was initially set to 100 bytes and then increased in each successive one-minute interval. The offered load and throughput plots for test 4 are depicted in Figs. 9 and 10.

As the size of packets in flow 2 increases, there appears to be a slight decrease in the throughput of flow 1 and a corresponding increase in the throughput of flow 2. Although the effect does not appear very significant, further investigation may be warranted. Note that as the size of packets in flow 2 is increased, so is the offered load. Thus, the noted effect may be due to the increasing offered load (as in test 3). Any follow-on testing should attempt to investigate the effect of varying packet size without changing the offered load (i.e., reduce packet rate as packet size is increased).

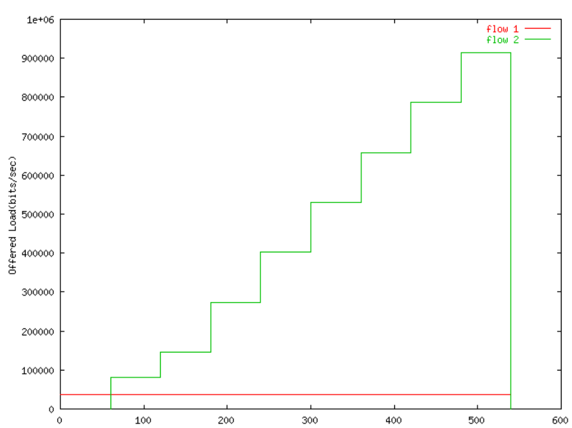

Fig. 9 — Test 4 offered load

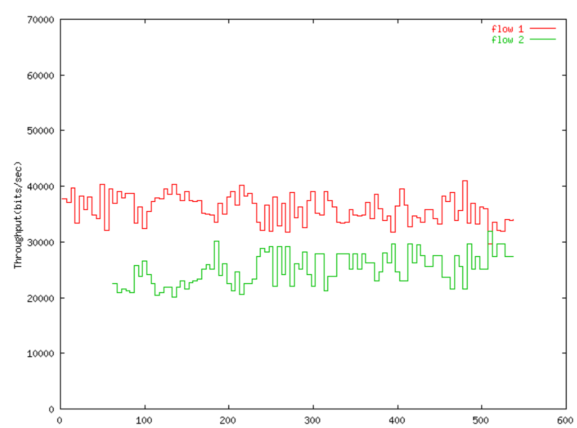

Fig. 10 — Test 4 throughput

Test 5

Test 5 was designed to determine if there is any reduction in the achievable throughput for an individual data flow when BRS is enabled. While the MGEN test script included two data flows—flow 1 assigned to the CRITICAL class and flow 2 assigned to the DEFAULT class—only one data flow was turned on during a given time interval. During the first two minutes of the test only flow 1 sent packets and during the last two minutes of the test only flow 2 sent packets. Each data flow sent packets with an offered load that exceeded the capacity of the link during its two-minute interval. The test was conducted once with BRS enable and once with BRS disabled. There was no significant degradation in the achievable throughput for an individual data flow when BRS was enabled. However, during this evaluation the rate of the bottleneck link on which BRS was configured was only 64 kbps. It is expected that there is an upper limit to the link rate at which BRS can operate; therefore, there may be performance degradation if used with higher data rate links. If this is of concern, further investigation may be warranted prior to use with higher data rate links.

Priority Queueing

The second set of tests investigated the use of priority queueing within a given BRS class. The configuration of BRS was simplified to include only the two original classes—LOCAL and DEFAULT. The percentage allocations for the LOCAL and DEFAULT classes were set to 15% and 85%, respectively. IP filters were defined to mark packets with a TAG from 1 to 5, based on the protocol (i.e., UDP) and the destination port of the packet. In the BRS configuration, the TAG value was used to assign packets a priority level within the DEFAULT class. The details regarding the tagging of packets and assignment of priority levels are outline in Table 1 below.

Table 1 — Filter Definitions and BRS Priority Assignments for Tests 6 and 7

Filter Name	Filter Match Criteria	Filter Action	BRS Class/Priority
test.urgent	UDP destination port 5001	TAG=1, Pass	DEFAULT/URGENT
test.high	UDP destination port 5002	TAG=2, Pass	DEFAULT/HIGH
test.normal	UDP destination port 5003	TAG=3, Pass	DEFAULT/NORMAL
test.low	UDP destination port 5004	TAG=4, Pass	DEFAULT/LOW
test.other	All other traffic	TAG=5, Pass	DEFAULT/NORMAL

Two MGEN test scripts were used to investigate the basic functionality of priority queueing with this router configuration. The test scripts were intentionally designed to be uncomplicated and facilitate interpretation of the results.

Test 6

Test 6 was designed to evaluate priority queueing with all possible combinations of two data flows of different priority levels. The MGEN test script included four separate data flows that were assigned to different priority levels—flow 1 was assigned LOW priority, flow 2 was assigned NORMAL priority, flow 3 was assigned HIGH priority, and flow 4 was assigned URGENT priority. The basic structure of the test was as follows. One of the data flows (e.g., LOW priority) was turned on with an offered load of approximately 48.1 kbps for a period of 2.5 minutes. During that 2.5-minute period, each of the other priority level data flows (e.g. NORMAL, HIGH, and URGENT) was individually turned (during separate non-overlapping one-minute intervals) with an offered load of approximately 35.8 kbps. This structure was repeated with each priority level data flow turned on for the longer duration with the higher offered load. Only two data flows were turned on during any given interval of time throughout the test and the combined offered load of two such data flows exceeded the capacity of the link. The offered load and throughput plots of the four data flows in test 6 are depicted in Figs. 11 and 12.

The throughput plot illustrates the expected behavior of priority queueing under the given test conditions. During each interval in which two data flows are on, the data flow with the higher relative priority achieves a throughput that is essentially equivalent to its offered load, while the throughput of the data flow with the lower relative priority is reduced to the remaining available capacity.

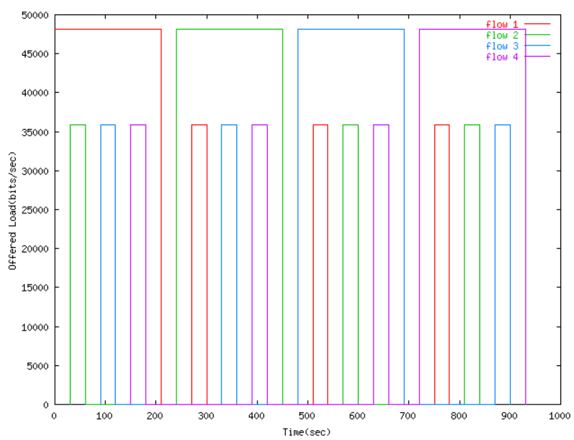

Fig. 11 — Test 6 offered load

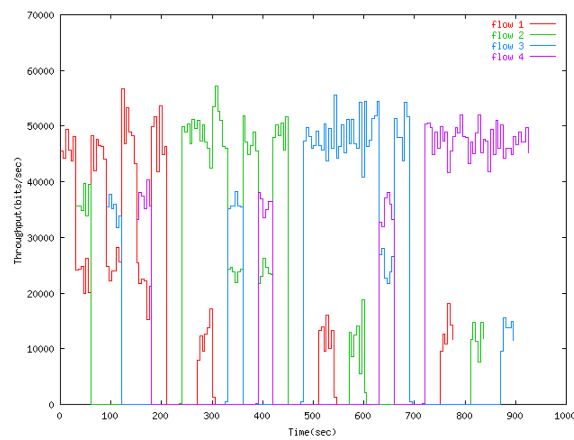

Fig. 12 — Test 6 throughput

Test 7

Test 7 was designed to investigate the interaction between four data flows with different priority levels and illustrate the potential for lower priority data flows to be locked out. The MGEN test script included four separate data flows assigned to different priority levels—flow 1 was assigned LOW priority, flow 2 was assigned NORMAL priority, flow 3 was assigned HIGH priority, and flow 4 was assigned URGENT priority. Initially the low priority data flow was turned on with an offered load of approximately 35.8 kbps. After each subsequent one-minute interval, the next highest priority data flow was turned on (with the same offered load) until all four data flows were on simultaneously. All four data flows remained on for a period of one minute, at which point the data flows were turned off, at one-minute intervals, in reverse order. Note that when two or more data flows were on simultaneously, their combined offered load exceeded the capacity of the bottleneck link. The offered load and throughput plots of the four data flows in test 7 are depicted in Figs. 13 and 14.

The results of this test indicate a potential problem with the priority queueing implementation. The throughput plot exhibits the expected behavior during the first two minutes and final two minutes of the test, but during the other portions of the test (i.e., when more than two different priority level data flows are turned on) the behavior appears incorrect.

When only the LOW priority data flow is turned on, it achieves a throughput equivalent to its offered load, as expected. Upon addition of the NORMAL priority data flow, the NORMAL priority data flow achieves a throughput equivalent to its offered load and the throughput of the LOW priority data flow is reduced to the remaining fraction of the link capacity. Again, this is the expected behavior for priority queueing. However, when the LOW, NORMAL and HIGH priority data flows are turned on simultaneously, it appears that the HIGH and NORMAL priority data flows share the link capacity equally, while the throughput of the LOW priority data flow is reduced to near zero. Thus, the throughput of the HIGH priority data flow is *less* than its offered load. Typically, with priority queueing, the expected behavior would be for the HIGH priority data flow to achieve a throughput equivalent to its offered load, while the throughput of the NORMAL priority data flow is reduced to the remaining fraction of the link capacity and the throughput of the LOW priority data flow is reduced to zero. Again, when all four data flows are on simultaneously, it appears that the URGENT, HIGH and NORMAL priority data flows share the link capacity equally, while the throughput of the LOW priority data flow is reduced to near zero. The expected behavior would be for the URGENT priority data flow to achieve a throughput equivalent to its offered load, while the throughput of the HIGH priority data flow is reduced to the remaining fraction of the link capacity and the throughput of the NORMAL and LOW priority data flows are reduced to zero.

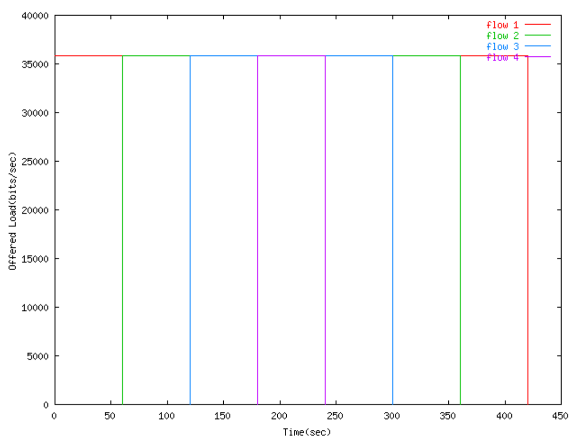

Fig. 13 — Test 7 offered load

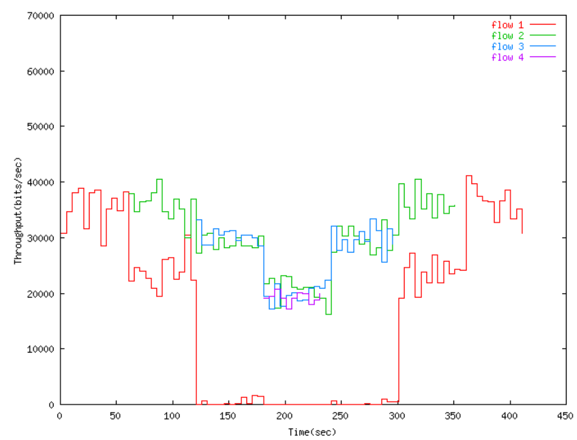

Fig. 14 — Test 7 throughput

Bandwidth Reservation with Priority Queueing

A final test was conducted to investigate the interaction of minimum percentage class allocations and priority queueing. The BRS configuration for the test included four classes—LOCAL, CLASS A, CLASS B and DEFAULT. The percentage allocations for LOCAL, CLASS A, CLASS B and DEFAULT were set to 10%, 20%, 30% and 40%, respectively. IP filters were defined to mark packets with a TAG from 1 to 13, based on the protocol (i.e., UDP) and the destination port of the packet. In the BRS configuration, the TAG value was used to assign packets a priority level within one of the classes. The details regarding the tagging of packets and assignment of priority levels are outline in Table 2.

Table 2 — Filter Definitions and BRS Priority Assignments for Test 8

Filter Name	Filter Match Criteria	Filter Action	BRS Class/Priority
test.urgent	UDP destination port 5001	TAG=1, Pass	DEFAULT/URGENT
test.high	UDP destination port 5002	TAG=2, Pass	DEFAULT/HIGH
test.normal	UDP destination port 5003	TAG=3, Pass	DEFAULT/NORMAL
test.low	UDP destination port 5004	TAG=4, Pass	DEFAULT/LOW
test.urgent(A)	UDP destination port 6001	TAG=5, Pass	CLASS A/URGENT
test.high(A)	UDP destination port 6002	TAG=6, Pass	CLASS A/HIGH
test.normal(A)	UDP destination port 6003	TAG=7, Pass	CLASS A/NORMAL
test.low(A)	UDP destination port 6004	TAG=8, Pass	CLASS A/LOW
test.urgent(B)	UDP destination port 7001	TAG=9, Pass	CLASS B/URGENT
test.high(B)	UDP destination port 7002	TAG=10, Pass	CLASS B/HIGH
test.normal(B)	UDP destination port 7003	TAG=11, Pass	CLASS B/NORMAL
test.low(B)	UDP destination port 7004	TAG=12, Pass	CLASS B/LOW
test.other	All other traffic	TAG=13, Pass	DEFAULT/NORMAL

Test 8

There are two specific areas of interest that test 8 was designed the further investigate— the roll of priority levels in the dynamic sharing of unused class allocations and the interaction of multiple data flows with different priority levels in different classes. Again, the MGEN test script is best described as a series of distinct one-minute intervals during each of which the offered load of the individual data flows were held constant. The MGEN script included a total of five different data flows—flow 1 was assigned to CLASS A with LOW priority, flow 2 was assigned to CLASS B with HIGH priority, flow 3 was assigned to DEFAULT with NORMAL priority, flow 4 was assigned to CLASS A with URGENT priority and flow 5 was assigned to DEFAULT with URGENT priority. The offered load and throughput plots of the four data flows in test 8 are depicted in Figs. 15 and 16.

During the first minute of the test, flows 1, 2 and 3 are all turned on with the same offered load to illustrate the percentage allocations. Flow 1 achieves a throughput of approximately 16 kbps (CLASS A), while flow 2 achieves a throughput of approximately 20 kbps (CLASS B) and flow 3 achieves a throughput of approximately 24 kbps (DEFAULT). Based on application of the configured percentages to the 60 kbps achievable throughput, the expected minimum percentage allocations would be 6 kbps for LOCAL, 12 kbps for CLASS A, 18 kbps for CLASS B and 24 kbps for DEFAULT. Thus, in this configuration it appears that each data flow received its minimum allocation, and the unused LOCAL allocation was shared by the CLASS A and CLASS B traffic. During the second minute of the test (60-120 seconds), the offered load of flow 3 was reduced to well below its minimum percentage allocation to provide insight into how the unused allocation would be shared by flows 1 and 2. While the throughput of flow 1 (LOW priority) shows a slight increase, the throughput of flow 2 (HIGH priority) increases dramatically. This implies that the priority level may be a significant factor for sharing unused allocations from other classes, as flow 2 has the higher relative priority.

When flow 4 (which is in the same class as flow 1, but has a higher relative priority level) is turned on during the third minute of the test (120-180 seconds), the throughput of flow 1 is reduced to zero. This is the expected behavior based on application of priority queueing to the packets within a common class. Through the third and forth minutes of the test (120-240 seconds), flows 2 and 4 achieve approximately the same throughput. Due to the lack of information regarding in mechanism for the sharing unused allocations, it is unclear whether this is a coincidence or a breakdown of the queueing mechanism as previously seen in test 7. However, during the fifth minute of the test (240-300 seconds), a problem with the queueing mechanism is more clearly evident. When flow 5 (which is in the same class as flow 3, but has a higher relative priority level) is turned on, the throughput of flow 3 is *not* reduced to zero. This indicates priority queueing is not being correctly applied to the packets within the DEFAULT class. While the results are much more complex to analyze and describe, it is apparent that the problems revealed in test 7 are also evident in test 8.

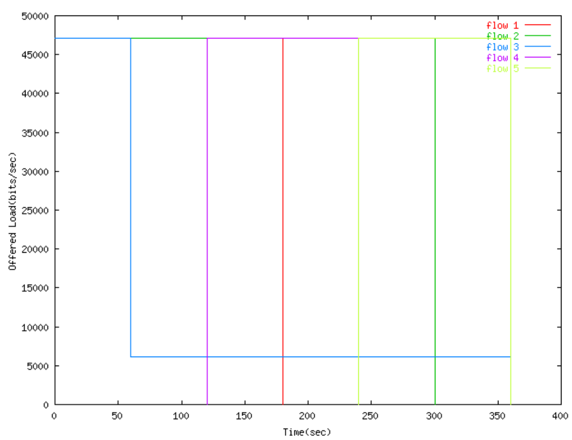

Fig. 15 — Test 8 offered load

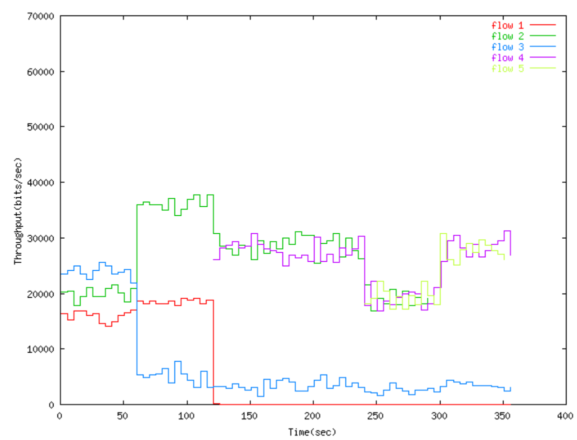

Fig. 16 — Test 8 throughput

Summary of Major Test Results

Managed bandwidth percentages for separate network traffic classes were effectively demonstrated under dynamically stressed traffic conditions. In addition, unused bandwidth allocations for classes of traffic were shown to be dynamically shared by network traffic from other classes. Under testing with unequal percentage bandwidth allocations (e.g., 25% and 75%), our initial test results indicate that traffic in the higher percentage class receives slightly more bandwidth than the configured allocation, while traffic in the lower percentage class receives slightly less. This percentage discrepancy appears to be within a few percentile points of the expected value (i.e., less than approximately 5% was observable). We conclude that the guaranteed minimum bandwidth service within OpenROUTE 3.0 works reasonable well with the exception of some small allocation discrepancies noted in our trials.

Initial tests of priority queueing, using two different priority levels simultaneously, revealed expected results; higher priority traffic received absolute precedence over lower priority traffic for all single combination pairings of the different priority levels. However, further testing of priority queueing, using more than two priority levels simultaneously, yielded some unexpected results. For example, using all four priority levels simultaneously, the URGENT, HIGH and NORMAL priority traffic appeared to share the link capacity equally, while the throughput of the LOW priority traffic was reduced to near zero. The expected behavior would be for the highest priority traffic to dominate and for traffic with lower priorities to receive leftover bandwidth in an absolute descending service order fashion. Thus, our experiments have revealed a potential flaw within the priority queueing model of the OpenROUTE 3.0 software.

RECOMMENDATIONS

In conclusion, we have extensively tested a number of features and operational modes of the OpenROUTE 3.0 BRS and priority queueing. Our findings indicate that under dynamic conditions and moderate link data rates, BRS appears to function reasonably well as a pure guaranteed minimum bandwidth service for aggregate traffic classes. The software appears robust and mature enough to deploy safely within the ADNS architecture (with the noted discrepancies discovered). Further, we discovered problems with the priority queueing service, which indicate it does not function as anticipated when servicing more than two simultaneous priority levels. Until these problems are fixed, we do not recommend reliance on the pure priority traffic handling features of OpenROUTE 3.0. These problems can be likely fixed and/or improved in future releases of software.

Appendix A
Router Settings for the Bandwidth Reservation System and IP Filtering

BANDWIDTH RESERVATION TESTS

The following router configurations were used to evaluate bandwidth reservation without priority queueing (tests 1-5)[1].

Bandwidth Reservation System

```
BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 2
maximum queue length 10 minimum queue length 3
total bandwidth allocated 100%
total classes defined (counting one local and one default) 3


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 30% bandwidth allocated
  the following protocols and filters are assigned:
    protocol IP with priority NORMAL
    protocol ARP with default priority
    protocol IPX with default priority
    protocol AP2 with default priority
    protocol BRIDGE with default priority
    filter TAG5 with priority NORMAL


class CRITICAL has 60% bandwidth allocated
  the following protocols and filters are assigned:
    filter TAG6 with priority NORMAL


default class is DEFAULT with priority NORMAL
```

IP Filtering

```
Listing Filters

Name                     Dir  Address            Port          Protocol
Idle
                                                                    Action
------------------------------------------------------------------------------
--
test.critical            Out                     dp=6000-6005  UDP  Tag=6  Off
                                                                    Pass
test.other               Both                                  Any  Tag=5  Off
                                                                    Pass
```

---

[1] bandwidth reservation was not enabled during half of test 5

PRIORITY QUEUEING TESTS

The following router configurations were used to evaluate priority queueing within a given class (tests 6 and 7).

Bandwidth Reservation System

```
BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 2
maximum queue length 10 minimum queue length 3
total bandwidth allocated 100%
total classes defined (counting one local and one default) 2


class LOCAL has 15% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 85% bandwidth allocated
  the following protocols and filters are assigned:
    protocol IP with priority NORMAL
    protocol ARP with default priority
    protocol IPX with default priority
    protocol AP2 with default priority
    protocol BRIDGE with default priority
    filter TAG1 with priority URGENT
    filter TAG2 with priority HIGH
    filter TAG3 with priority NORMAL
    filter TAG4 with priority LOW
    filter TAG5 with priority NORMAL


default class is DEFAULT with priority NORMAL
```

IP Filtering

```
Listing Filters

Name                      Dir  Address            Port          Protocol
Idle
                                                                     Action
-------------------------------------------------------------------------------
--
test.urgent               Out                     dp=5001       UDP  Tag=1  Off
                                                                     Pass
test.high                 Out                     dp=5002       UDP  Tag=2  Off
                                                                     Pass
test.normal               Out                     dp=5003       UDP  Tag=3  Off
                                                                     Pass
test.low                  Out                     dp=5004       UDP  Tag=4  Off
                                                                     Pass
test.other                Both                                  Any  Tag=5  Off
                                                                     Pass
```

BANDWIDTH RESERVATION WITH PRIORITY QUEUEING TESTS

The following router configurations were used to evaluate bandwidth reservation in combination with priority queueing (test 8).

Bandwidth Reservation System

```
BRS Config <NET-2> LIST

BANDWIDTH RESERVATION listing from SRAM
bandwidth reservation is enabled
interface number 2
maximum queue length 10 minimum queue length 3
total bandwidth allocated 100%
total classes defined (counting one local and one default) 4


class LOCAL has 10% bandwidth allocated
  protocols and filters cannot be assigned to this class.


class DEFAULT has 40% bandwidth allocated
  the following protocols and filters are assigned:
    protocol IP with priority NORMAL
    protocol ARP with default priority
    protocol IPX with default priority
    protocol AP2 with default priority
    protocol BRIDGE with default priority
    filter TAG1 with priority URGENT
    filter TAG2 with priority HIGH
    filter TAG3 with priority NORMAL
    filter TAG4 with priority LOW
    filter TAG13 with priority NORMAL


class ClassA has 20% bandwidth allocated
  the following protocols and filters are assigned:
    filter TAG5 with priority URGENT
    filter TAG6 with priority HIGH
    filter TAG7 with priority NORMAL
    filter TAG8 with priority LOW


class ClassB has 30% bandwidth allocated
  the following protocols and filters are assigned:
    filter TAG9 with priority URGENT
    filter TAG10 with priority HIGH
    filter TAG11 with priority NORMAL
    filter TAG12 with priority LOW


default class is DEFAULT with priority NORMAL
```

IP Filtering

```
Listing Filters

Name                     Dir  Address            Port          Protocol
Idle
                                                                    Action
------------------------------------------------------------------------------
--
test.urgent              Out                     dp=5001       UDP  Tag=1  Off
                                                                    Pass
test.high                Out                     dp=5002       UDP  Tag=2  Off
                                                                    Pass
test.normal              Out                     dp=5003       UDP  Tag=3  Off
                                                                    Pass
test.low                 Out                     dp=5004       UDP  Tag=4  Off
                                                                    Pass
test.urgent(A)           Out                     dp=6001       UDP  Tag=5  Off
                                                                    Pass
test.high(A)             Out                     dp=6002       UDP  Tag=6  Off
                                                                    Pass
test.normal(A)           Out                     dp=6003       UDP  Tag=7  Off
                                                                    Pass
test.low(A)              Out                     dp=6004       UDP  Tag=8  Off
                                                                    Pass
test.urgent(B)           Out                     dp=7001       UDP  Tag=9  Off
                                                                    Pass
test.high(B)             Out                     dp=7002       UDP  Tag=10 Off
                                                                    Pass
test.normal(B)           Out                     dp=7003       UDP  Tag=11 Off
                                                                    Pass
test.low(B)              Out                     dp=7004       UDP  Tag=12 Off
                                                                    Pass
test.other               Both                                  Any  Tag=13 Off
                                                                    Pass
```

Appendix B

MGEN Test Scripts

BANDWIDTH RESERVATION TESTS

Test 1

```
START 18:20:00GMT
# Protected Flow
00000       1     ON    132.250.68.20:6000      POISSON     80    100
60000       1     MOD   132.250.68.20:6000      POISSON     30    100
300000      1     MOD   132.250.68.20:6000      POISSON     80    100
420000      1     MOD   132.250.68.20:6000      POISSON     45    100
480000      1     OFF
# Unprotected Flow
120000      2     ON    132.250.68.20:5000      POISSON     10    100
180000      2     MOD   132.250.68.20:5000      POISSON     25    100
240000      2     MOD   132.250.68.20:5000      POISSON     80    100
360000      2     MOD   132.250.68.20:5000      POISSON     10    100
540000      2     MOD   132.250.68.20:5000      POISSON     80    100
600000      2     OFF
```

Test 2

```
START 19:55:00GMT
#Protected Flow One
00000       1     ON    132.250.68.20:6000      POISSON     50    100
120000      1     MOD   132.250.68.20:6000      POISSON     25    100
240000      1     OFF
#Protected Flow Two
00000       2     ON    132.250.68.20:6001      POISSON     25    100
300000      2     OFF
#Unprotected Flow One
60000       3     ON    132.250.68.20:5000      POISSON     80    100
360000      3     OFF
#Unprotected Flow Two
180000      4     ON    132.250.68.20:5001      POISSON     20    100
360000      4     OFF
```

Test 3

```
START 20:35:00GMT
# Protected Flow
00000       1     ON    132.250.68.20:6000      POISSON     50    100
360000      1     OFF
# Unprotected Flow
60000       2     ON    132.250.68.20:5000      POISSON     25    100
120000      2     MOD   132.250.68.20:5000      POISSON     50    100
180000      2     MOD   132.250.68.20:5000      POISSON     100   100
240000      2     MOD   132.250.68.20:5000      POISSON     200   100
300000      2     MOD   132.250.68.20:5000      POISSON     400   100
360000      2     OFF
```

Test 4

```
START 12:35:00GMT
# Protected Flow
00000       1     ON    132.250.68.20:6000      POISSON     36    100
540000      1     OFF
# Unprotected Flow
60000       2     ON    132.250.68.20:5000      POISSON     80    100
120000      2     MOD   132.250.68.20:5000      POISSON     80    200
180000      2     MOD   132.250.68.20:5000      POISSON     80    400
240000      2     MOD   132.250.68.20:5000      POISSON     80    600
300000      2     MOD   132.250.68.20:5000      POISSON     80    800
360000      2     MOD   132.250.68.20:5000      POISSON     80    1000
420000      2     MOD   132.250.68.20:5000      POISSON     80    1200
480000      2     MOD   132.250.68.20:5000      POISSON     80    1400
540000      2     OFF
```

Test 5

```
START 13:25:00GMT
# Protected
00000       1     ON    132.250.68.20:6000      POISSON     80    100
120000      1     OFF
# Unprotected
120000      2     ON    132.250.68.20:5000      POISSON     80    100
240000      2     OFF
```

PRIORITY QUEUEING TESTS

Test 6

```
START 14:30:00GMT
#Low Priority-add other priorities
00000       1     ON    132.250.68.20:5004      POISSON     47    100
30000       2     ON    132.250.68.20:5003      POISSON     35    100
60000       2     OFF
90000       3     ON    132.250.68.20:5002      POISSON     35    100
120000      3     OFF
150000      4     ON    132.250.68.20:5001      POISSON     35    100
180000      4     OFF
210000      1     OFF
#Normal Priority-add other priorities
240000      2     ON    132.250.68.20:5003      POISSON     47    100
270000      1     ON    132.250.68.20:5004      POISSON     35    100
300000      1     OFF
330000      3     ON    132.250.68.20:5002      POISSON     35    100
360000      3     OFF
390000      4     ON    132.250.68.20:5001      POISSON     35    100
420000      4     OFF
450000      2     OFF
#High Priority-add other priorities
480000      3     ON    132.250.68.20:5002      POISSON     47    100
510000      1     ON    132.250.68.20:5004      POISSON     35    100
540000      1     OFF
570000      2     ON    132.250.68.20:5003      POISSON     35    100
600000      2     OFF
630000      4     ON    132.250.68.20:5001      POISSON     35    100
660000      4     OFF
690000      3     OFF
#Urgent Priority-add other priorities
720000      4     ON    132.250.68.20:5001      POISSON     47    100
750000      1     ON    132.250.68.20:5004      POISSON     35    100
780000      1     OFF
810000      2     ON    132.250.68.20:5003      POISSON     35    100
840000      2     OFF
870000      3     ON    132.250.68.20:5002      POISSON     35    100
900000      3     OFF
930000      4     OFF
```

Test 7

```
START 15:20:00GMT
00000       1     ON    132.250.68.20:5004      POISSON     35    100
60000       2     ON    132.250.68.20:5003      POISSON     35    100
120000      3     ON    132.250.68.20:5002      POISSON     35    100
180000      4     ON    132.250.68.20:5001      POISSON     35    100
240000      4     OFF
300000      3     OFF
360000      2     OFF
420000      1     OFF
```

BANDWIDTH RESERVATION WITH PRIORITY QUEUEING TESTS

Test 8

```
START 09:43:00GMT
#Class A Low
00000       1     ON    132.250.68.20:6004      POISSON     46    100
180000      1     OFF
#Class B High
00000       2     ON    132.250.68.20:7002      POISSON     46    100
300000      2     OFF
#Default Class Normal
00000       3     ON    132.250.68.20:5003      POISSON     46    100
60000       3     MOD   132.250.68.20:5003      POISSON     6     100
360000      3     OFF
#Class A Urgent
120000      4     ON    132.250.68.20:6001      POISSON     46    100
360000      4     OFF
#Default Class Urgent
240000      5     ON    132.250.68.20:5001      POISSON     46    100
360000      5     OFF
```