Study of QoS and Traffic Control Mechanisms in IP Networks

Filipe Oliveira, João Rua, Miguel Zenha
Computer Science Department
Universidade do Minho
Email: a57816@alunos.uminho.pt, a41841@alunos.uminho.pt, a66551@alunos.uminho.pt,

***Abstract*—In traditional networks, all connections and services get the same treatment. However, since network resources are limited, and the overall Internet only offers a "Best-Effort" approach, it is important to differentiate between connection classes, and to be able to treat them accordingly to standardised and well documented parameters.**

This exploratory essay focus on developing a comparative study of traffic control mechanisms in IP networks and corresponding parametrisation, using the Network Simulator NS-2. In order to do so, a test platform will be presented and several Diffserv parameters will be discussed.

1. Network topology to be used

The network topology to be used as test platform is illustrated in figure 1. The network topology includes six clients (from Cli1 to Cli6), two edge routers (E1 and E2), and a core router (C0). The clients' access links have a capacity of 5Mbps and a delay of 5ms, and the core network links have a capacity of 5Mbps and a delay of 10ms.

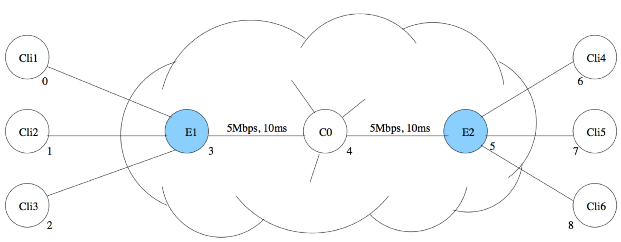

Figure 1. ISP network topology

The topology is deliberately symmetric to simplify traffic analysis. During this exploratory essay several changes will be made regarding the services/applications that every Client holds, however, the topology remains unchanged.

In most of the cases, it will be enough to analyse flow in one way, however, in the last analyse on chapter **??**, flows in both ways will be analysed due to the bigger complexity of the simulation.

As the topology evidences, if all clients use the link capacity simultaneously then congestion will occur in the network backbone, and the service provider will not be able to guarantee proper traffic delivery. To minimise or solve this effect, several traffic control mechanisms will be used in order to promote quality of service (QoS) in the domain.

Simulations for all scenarios were 15 seconds long. This was a very short simulation time, but enabled us to achieve a confidence interval, producing a stable final state.

2. Applications/Services to be used

- **CBR over UDP** - generates Constant Bit Rate (CBR) traffic over UDP. This may correspond to the transmission of audio or video traffic at a regular/periodic rate.
  - Parameters: rate (bits/sec) e packet size (Bytes);
- **FTP** - transfer of large files over TCP;
- **Voice over UDP** - simulates a voice call over UDP; This traffic is characterised by having a constant rate, alternating between talk and silence time periods.
  - Parameters: rate (bits/sec) and burst size (in seconds).

3. Tools and evaluation metrics

In order to infer the network quality of service we will take in consideration the following parameters Metrics to use in the simulations:

- **Loss rate** (total and per flow), in packets/sec.
- **Bandwidth** in use (total and per flow), in bits/sec.

4. A - Simulating the "Best-Effort" scenario

By default, routers handle packets based on a simple FIFO queueing system, trying to forward them in the best possible way according to the available resources (memory and CPU).

This well-known model is called Best-Effort as there are QoS guarantees on packet delivery (in terms of bounded delays, loss and/or bandwidth utilisation).

For a first approach we considered similar clients with CBR applications, generating each one a rate of 3Mbps, for a total of six flows ($0 \rightarrow 8$, $1 \rightarrow 7$, $2 \rightarrow 6$, $8 \rightarrow 0$, $7 \rightarrow 1$, $6 \rightarrow 2$).

4.1. Identification of the links under congestion

Producing the graphs illustrating the levels of loss and bandwidth utilisation along the time, we can infer that the core network links ($3 \leftrightarrow 4$, $4 \leftrightarrow 5$), since the full bandwidth is being used, as stated in figures 2 and 3.

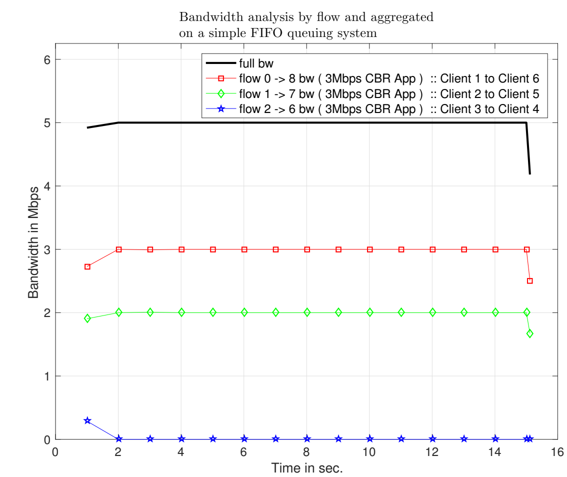

Figure 2. Bandwidth analysis by flow and aggregated on a simple FIFO queueing system, simulation a "best effort" scenario

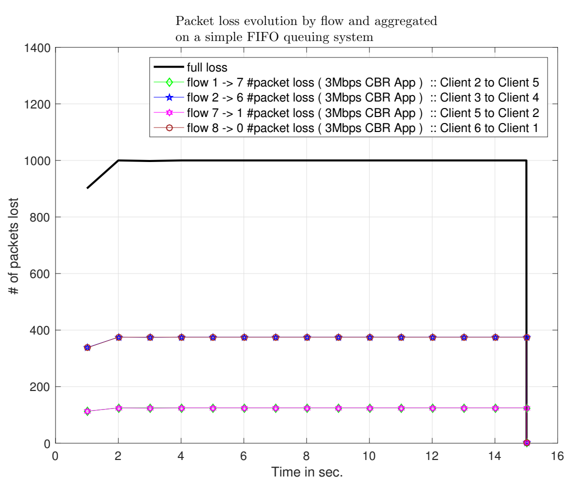

Figure 3. Packet loss evolution by flow and aggregated on a simple FIFO queueing system, simulation a "best effort" scenario

As stated before, the Internet's "best-effort" scenario produces an undesired non-equitable bandwidth distribution. Denote that this simple simulation only deals with one type of service simulation. The inclusion of other, "more sensible" to network congestion, services like for example VOIP, would result in an unacceptable QoS.

Changing the queues associated with the links under congestion from DropTail to RED would theoretically result into in a better service. The corresponding results are shown in figures 4 and 5.

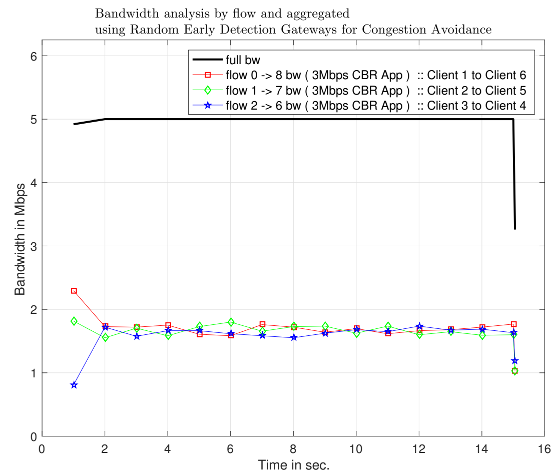

Figure 4. Bandwidth analysis by flow and aggregated on a simple FIFO queueing system, simulation a "best effort" scenario, using Random Early Detection Gateways for Congestion Avoidance

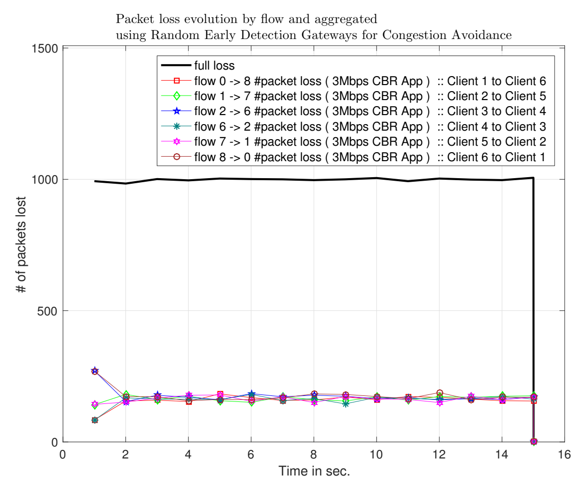

Figure 5. Packet loss evolution by flow and aggregated on a simple FIFO queueing system, simulation a "best effort" scenario, using Random Early Detection Gateways for Congestion Avoidance

Notice that this "solution" only improves the equitable bandwidth distribution across flow because they all are produced with the same service/traffic type. If we included for exemplo some TCP over IP service, since it behaves in order to prevent/diminish congestion, it would suffer more from bandwidth "starvation" than any service using UDP over IP.

In figures 6 and 7, simulated results are show if one client would generate more CBR traffic than the others, on a simple FIFO queuing system, simulating a "best effort" scenario.

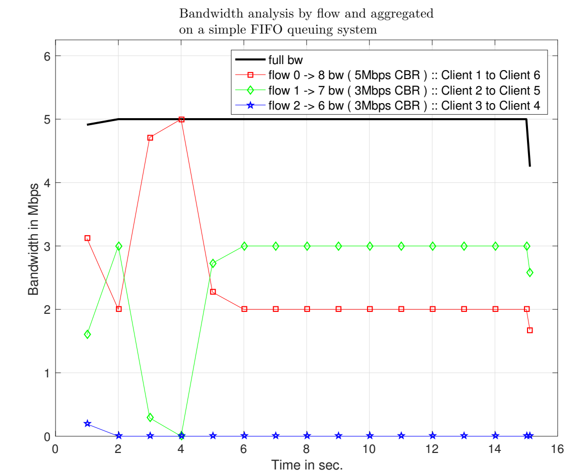

Figure 6. Bandwidth analysis by flow and aggregated on a simple FIFO queueing system, simulation a "best effort" scenario, in which one client would generate more CBR traffic than the others.

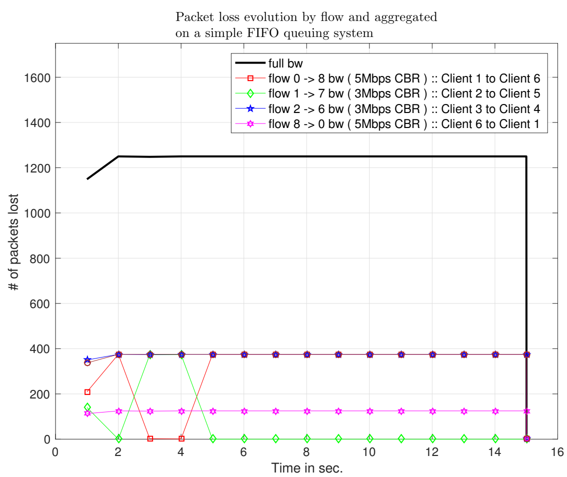

Figure 7. Packet loss evolution by flow and aggregated on a simple FIFO queueing system, simulation a "best effort" scenario, in which one client would generate more CBR traffic than the others.

5. B - Simulating a multi-service network in the "Best-Effort" scenario

In a more realistic scenario, it would be expectable to have both UDP and TCP traffic with other characteristics (FTP, HTTP, etc.). Using the procedures already included in the simulation script, several changes were made in order to obtain the following scenario:

- a **CBR application** sending 4Mbps from client 1 to client 6, and other from client 6 to client 1;
- a **FTP connection** from client 2 to client 5, and other from client 4 to client 2;
- a **voice connection over UDP** from client 3 to client 4, and vice-versa. Since VOIP Bandwidth consumption naturally depends on the codec used, we selected G.711 - 64 Kbps Bitrate and 87.2 Kbps Nominal Ethernet Bandwidth, and simulated a maximum of 30 calls at any given simulation time. The presented graphic results for VOIP are an aggregation of all the 30 calls.

The corresponding results are shown in figures 8 and 9.

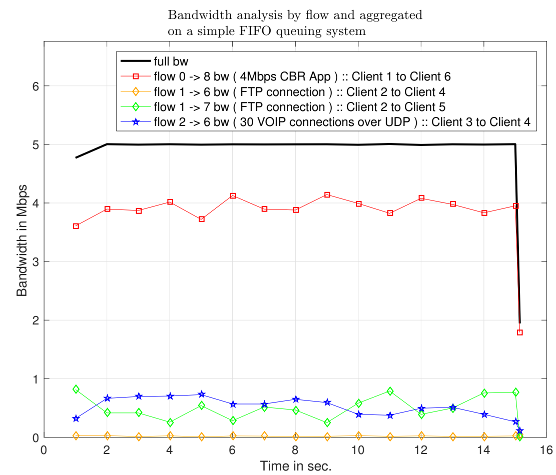

Figure 8. Bandwidth analysis by flow and aggregated on a simple FIFO queueing system, simulating a multi-service network in a "best effort" scenario.

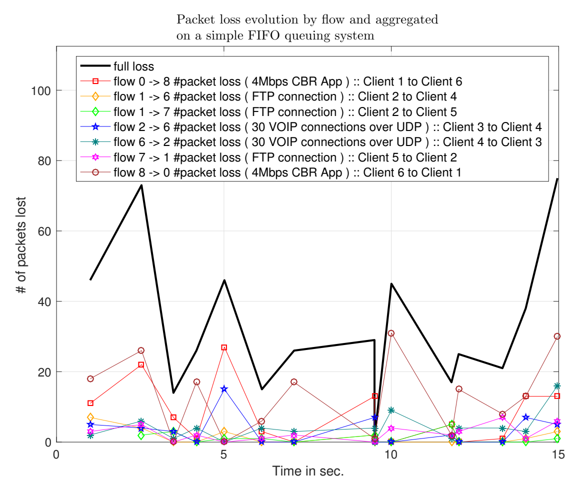

Figure 9. Packet loss evolution by flow and aggregated on a simple FIFO queueing system, simulating a multi-service network in a "best effort" scenario.

As you can state in figure 9, services like VOIP connections over UDP and FTP connections suffer the most when the network is fully congested, being the flows $1 \rightarrow 7$ (FTP connection), $2 \rightarrow 6$ (30 VOIP connections over UDP), and $6 \rightarrow 2$ (30 VOIP connections over UDP), the ones that are most affected.

The relation between flow, total number of packets lost, and percentage of loss/sent packages, is presented in table

1, and lets us fully understand the harm of treating all traffic with the same priority.

TABLE 1. RELATION BETWEEN FLOW, TOTAL NUMBER OF PACKETS LOST, TOTAL NUMBER OF PACKETS SENT, AND PERCENTAGE OF LOSS/SENT PACKAGES, ON A SIMPLE FIFO QUEUEING SYSTEM, SIMULATING A MULTI-SERVICE NETWORK IN A "BEST EFFORT" SCENARIO.

Flow	#packets loss	#packets received	% loss/received
0 → 8 (4 Mbps CBR App)	116	29652	0.3912 %
1 → 6 (FTP connection)	20	3720	0.5376 %
1 → 7 (FTP connection)	15	3743	0.4007 %
2 → 6 (30 VOIP connections over UDP)	48	4016	1.1952 %
6 → 2 (30 VOIP connections over UDP)	61	4349	1.4026 %
7 → 1 (FTP connection)	36	3620	0.9945 %
8 → 0 (4 Mbps CBR App)	185	29445	0.6283 %

Please denote that despite the loss percentage doesn't seem to high for all flows, those values are presented as a mean value, giving the possibility of loss increase in certain time intervals, and decrease in others. We should therefore analyse the percentage of loss per flow by connection time. The corresponding results are shown in figure 10.

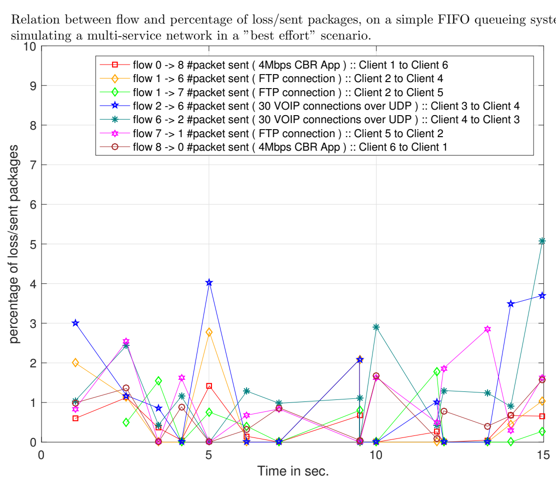

Figure 10. Packet loss/sent percentage by flow evolution, on a simple FIFO queueing system, simulating a multi-service network in a "best effort" scenario.

As stated before, it is in flows 2 → 6 (30 VOIP connections over UDP) and 6 → 2 (30 VOIP connections over UDP) that we observe a bigger loss percentage over time (5%). The service that should be prioritised and treated as the most volatile to delays (real time traffic necessity) is the one suffering the most from congestion.

6. Simulating Differentiated Services :: protecting vulnerable packets

In some flows, the loss of some segments has more impact than others on the perfomance of the service/application – as FTP (over TCP) and VOIP (over UDP). We call the packets from these services, our vulnerable packets in our simulation scenario. By "marking" these segments/packets with a higher priority and implementing the priority using a diffserv architecture, the overall perfomance and QoS considerably improves.

For a first approach we considered similar clients with CBR applications, generating each one a rate of 3Mbps, for a total of six flows (0 → 8, 1 → 7, 2 → 6, 8 →0, 7 →1, 6 →2).

We considered the simple network topology defined in figure 1, and before any further change we analysed the links under congestion, and identified the queue which suffers higher packet loss.

6.1. Identification of the queueing management parameters for an Edge → Core Configuration and an Core → Edge Configuration

The presented parameters for an Edge → Core Configuration and an Core → Edge Configuration, were not chosen to necessarily obtain an optimal initial performance, but rather to create conditions that allow us to study the effect of diffserv on diminishing the loss probabilities of vulnerable flows, and the impact of the further changes versus this initial simple configuration.

Consider this simulation scenario and its initial traffic model as a base comparison model for the "best-effort" model presented in section 4 on page 1.

6.1.1. E1 - C0 (Edge → Core Configuration).

- **The number of existing queues and the traffic scheduler in use**
  1 physical queue, implementing 2 virtual queues;

- **Policy Entry**
  - **Client1 → Client6** – TokenBucket:
    - **Committed Information Rate:** 2 Mbits/sec;
    - **Committed Burst Size:** 5 KBytes;
    - **Policer Table** has initial (green) code point 10, and downgraded (yellow) code point 11;
  - **Every remaining initial and end station** – Dumb:
    - **Policer Table** has always downgraded (yellow) code point 11;

- **The queueing discipline in use and the configuration of each queue:**
  Round Robin scheduling and RIO-C Active Queue Management:
  - queue 0:
    - **minimum threshold**: 20 Packets;

- **maximum threshold**: 40 Packets;
- **maximum dropping probability**: $2 * 10^{-2}$;

– queue 1:

- **minimum threshold**: 10 Packets;
- **maximum threshold**: 20 Packets;
- **maximum dropping probability**: $1 * 10^{-1}$;

- **the amount of memory allocated to the queues:** Default queue buffer size is 20 packets (Packet size 1 KB) : 20KB per queue;

- **the queues which handle data flows:** Code point 10 mapped to physical queue 0 and virtual queue 0, Code point 11 mapped to physical queue 0 and virtual queue 1;

6.1.2. C0 - E2 (Core → Edge Configuration).

- **The number of existing queues and the traffic scheduler in use** 1 physical queue, implementing 2 virtual queues;

- **The queueing discipline in use and the configuration of each queue:** Round Robin scheduling and RIO-C Active Queue Management:

  – queue 0:

  - **minimum threshold**: 20 Packets;
  - **maximum threshold**: 40 Packets;
  - **maximum dropping probability**: $2 * 10^{-2}$;

  – queue 1:

  - **minimum threshold**: 10 Packets;
  - **maximum threshold**: 20 Packets;
  - **maximum dropping probability**: $1 * 10^{-1}$;

- **the amount of memory allocated to the queues:** Default queue buffer size is 20 packets (Packet size 1 KB) : 20KB per queue;

- **the queues which handle data flows:** Code point 10 mapped to physical queue 0 and virtual queue 0, Code point 11 mapped to physical queue 0 and virtual queue 1;

6.2. Identification of the queue which suffers higher packet loss

Taking into account the simulation results/statistics, presented on tables 2 to 7, we can identify the queue which suffers higher packet loss – physical queue 0 and virtual queue 1, from E1 to C0 (Edge → Core Configuration). This behaviour is easily explained since every traffic with initial station that is not Client1, and end station that is not Client6, is always downgrade to code point 11 (yellow tag). In congestion, these are the first packets to be dropped (wether late dropped or early dropped).

6.2.1. Statistics for time = 5s. .

TABLE 2. STATISTICS FOR THE QUEUE FROM E1 TO C0 (EDGE → CORE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	5619	3136	2097	386
10	1252	1252	0	0
11	4367	1884	2097	386

TABLE 3. STATISTICS FOR THE QUEUE FROM C0 TO E2 (CORE → EDGE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	3111	3111	0	0
10	1242	1242	0	0
11	1869	1869	0	0

6.2.2. Statistics for time = 10s. .

TABLE 4. STATISTICS FOR THE QUEUE FROM E1 TO C0 (EDGE → CORE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	11244	6261	4197	786
10	2502	2502	0	0
11	8742	3759	4197	786

TABLE 5. STATISTICS FOR THE QUEUE FROM C0 TO E2 (CORE → EDGE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	6236	6236	0	0
10	2492	2492	0	0
11	3744	3744	0	0

6.2.3. Statistics for time = 15s. .

TABLE 6. STATISTICS FOR THE QUEUE FROM E1 TO C0 (EDGE → CORE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	16869	9386	6298	1185
10	3752	3752	0	0
11	13117	5634	6298	1185

TABLE 7. STATISTICS FOR THE QUEUE FROM C0 TO E2 (CORE → EDGE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	9361	9361	0	0
10	3742	3742	0	0
11	5619	5619	0	0

6.3. A visual interpretation of the packet loss and bandwidth utilisation along the time

Given the simulation scenario and the initial traffic model presented on section 6 on page 4, the corresponding results illustration the levels of loss and bandwidth utilisation along time are show in figures 11 and 12.

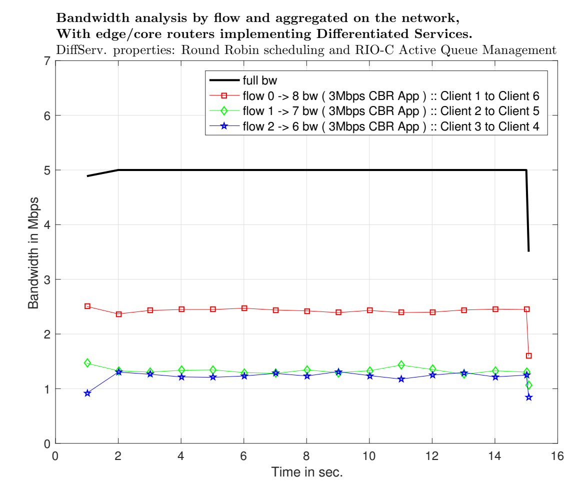

Figure 11. Bandwidth analysis by flow and aggregated on a simple FIFO queueing system, using Random Early Detection Gateways for congestion avoidance.

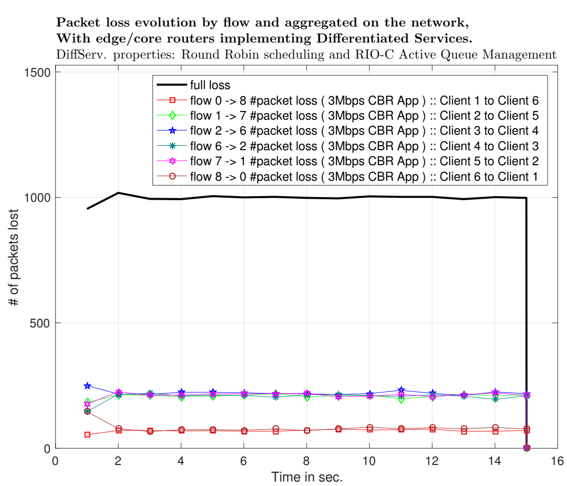

Figure 12. Packet loss evolution by flow and aggregated on a simple FIFO queueing system, using Random Early Detection Gateways for congestion avoidance.

When comparing the simulation results to the "best-effort" scenario we conclude that, despite the "final solution" being unacceptable, every flow obtains some quota of the available total bandwidth. We can observe that we also achieved a better packet loss maximum for all the simulation flows (approximately 250 vs 400 for the "best-effort" scenario). We have achived differentiated services for th e same type of traffic, based on the initial station and end station of every traffic.

Given that, the next step is to assume that traffic from all clients is marked as belonging to the same class of service, and verify if the option for a single traffic class bring any added value to network QoS when compared with the best-effort scenario. The corresponding results are shown in figures 13 and 14.

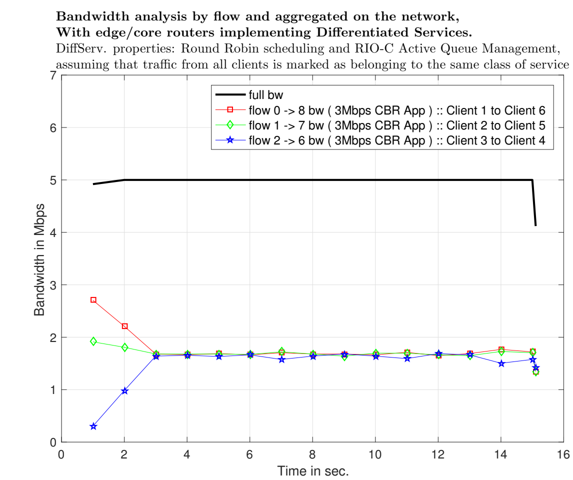

Figure 13. Bandwidth analysis by flow and aggregated on a simple FIFO queueing system, using Random Early Detection Gateways for congestion avoidance.

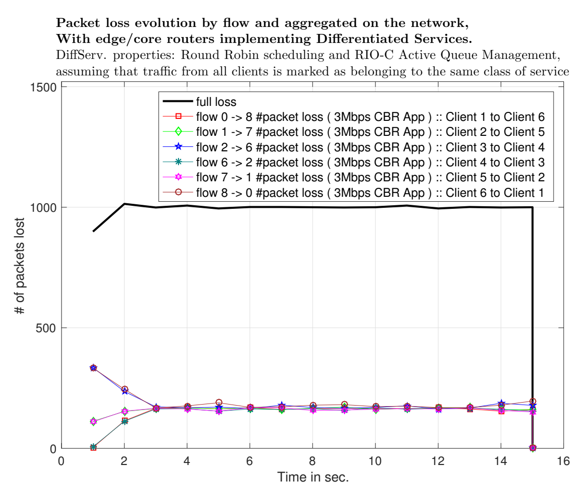

Figure 14. Packet loss evolution by flow and aggregated on a simple FIFO queueing system, using Random Early Detection Gateways for congestion avoidance.

Notice that this "solution" only improves the equitable bandwidth distribution across flow because they all are produced with the same service/traffic type, and, when compared to the "best-effort" scenario with Random Early Detection for Congestion Avoidance, the only improvement is the possibility of limiting the bandwidth percentage to be used, since we can "become" harsher to the yellow tagged traffic increasing its maximum dropping probability to 1, and therefore dropping every traffic out the Committed Information Rate limit.

But the problem of the multi-services in the network maintains. In this simple simulation scenario there is only the possibility of tagging traffic to be ”in-bounds” or ”out-of-bounds”. If we added both UDP and TCP traffic with other characteristics, the problems identified in section 5 would still be observed.

6.4. Simulating a multi-service network in a Diff-Serv scenario

Suppose that the service provider intends to implement the following policy:

- To assure a 30 % of capacity to clients with identical characteristics (the full capacity may be used if available).
- Traffic exceeding the negotiated rate must be downgraded, i.e., forwarded with lower priority.

Using the procedures already included in the simulation script, several changes were made in order to obtain the following traffic model:

- a **CBR application** sending 4Mbps from client 1 to client 6, starting at 2.5s;
- a **CBR application** sending 4Mbps from client 2 to client 4, starting at 5s;
- a **CBR application** sending 4Mbps from client 5 to client 2, starting at 10s;
- a **CBR application** sending 4Mbps from client 6 to client 1, starting at 0s;
- a **FTP connection** from client 2 to client 5, and other from client 4 to client 2;
- a **voice connection over UDP** from client 3 to client 4, and vice-versa. Since VOIP Bandwidth consumption naturally depends on the codec used, we selected G.711 - 64 Kbps Bitrate and 87.2 Kbps Nominal Ethernet Bandwidth, and simulated a maximum of 30 calls at any given simulation time. The presented graphic results for VOIP are an aggregation of all the 30 calls.

In order to correctly simulate Differentiated Services, giving special attention in protecting vulnerable real-time packets, several changes were made to the parameters for Edge → Core Configurations and an Core → Edge Configurations:

6.4.1. Edge → Core Configurations.

- **The number of existing queues and the traffic scheduler in use**
  3 physical queues, each implementing 3 virtual queues;

- **Policy Entry**
  - **Client1 → Client6 (4 Mbps CBR App)** – Two Rate Three Color Marker:
    - **Committed Information Rate:** 1 Mbit-s/sec;
    - **Committed Burst Size:** 5 KBytes;
    - **Peak Information Rate:** 3 Mbits/sec;
    - **Peak Burst Size:** 5 KBytes;
    - **Policer Table** has initial (green) code point 10, and downgraded (yellow) code point 11, and further downgraded (red) code point 12;
  - **Client2 → Client4 (4 Mbps CBR App)** – Two Rate Three Color Marker:
    - **Committed Information Rate:** 1 Mbit-s/sec;
    - **Committed Burst Size:** 5 KBytes;
    - **Peak Information Rate:** 3 Mbits/sec;
    - **Peak Burst Size:** 5 KBytes;
    - **Policer Table** has initial (green) code point 10, and downgraded (yellow) code point 11, and further downgraded (red) code point 12;
  - **Client2 → Client5 (FTP connection)** – Time Sliding Window Three Color Marker:
    - **Committed Information Rate:** 1 Mbit-s/sec;
    - **Peak Information Rate:** 3 Mbits/sec;
    - **Policer Table** has initial (green) code point 7, and downgraded (yellow) code point 8, and further downgraded (red) code point 9;
  - **Client3 → Client4 (30 VOIP connections over UDP)** – TokenBucket:
    - **Committed Information Rate:** 1 Mbit-s/sec;
    - **Committed Burst Size:** 5 KBytes;
    - **Policer Table** has initial (green) code point 4, and downgraded (yellow) code point 5;
  - **Client4 → Client2 (FTP connection)** – Time Sliding Window Three Color Marker:
    - **Committed Information Rate:** 1 Mbit-s/sec;
    - **Peak Information Rate:** 3 Mbits/sec;
    - **Policer Table** has initial (green) code point 7, and downgraded (yellow) code point 8, and further downgraded (red) code point 9;
  - **Client4 → Client3 (30 VOIP connections over UDP)** – TokenBucket:
    - **Committed Information Rate:** 1 Mbit-s/sec;

    * **Committed Burst Size:** 5 KBytes;
    * **Policer Table** has initial (green) code point 4, and downgraded (yellow) code point 5;
  - **Client5 $\rightarrow$ Client2 (4 Mbps CBR App)** – Two Rate Three Color Marker:
    * **Committed Information Rate:** 1 Mbits/sec;
    * **Committed Burst Size:** 5 KBytes;
    * **Peak Information Rate:** 3 Mbits/sec;
    * **Peak Burst Size:** 5 KBytes;
    * **Policer Table** has initial (green) code point 10, and downgraded (yellow) code point 11, and further downgraded (red) code point 12;
  - **Client6 $\rightarrow$ Client1 (4 Mbps CBR App)** – Two Rate Three Color Marker:
    * **Committed Information Rate:** 1 Mbits/sec;
    * **Committed Burst Size:** 5 KBytes;
    * **Peak Information Rate:** 3 Mbits/sec;
    * **Peak Burst Size:** 5 KBytes;
    * **Policer Table** has initial (green) code point 10, and downgraded (yellow) code point 11, and further downgraded (red) code point 12;
  - **Every remaining initial and end station** – Dumb:
    * **Policer Table** has always downgraded (yellow) code point 11;

- **The queueing discipline in use and the configuration of each queue:**
  Weighted Round Robin scheduling and RIO-C Active Queue Management:
  - physical queue 0:
    * virtual queue 0:
      · **minimum threshold**: 20 Packets;
      · **maximum threshold**: 40 Packets;
      · **maximum dropping probability**: $2 * 10^{-2}$;
    * virtual queue 1:
      · **minimum threshold**: 10 Packets;
      · **maximum threshold**: 20 Packets;
      · **maximum dropping probability**: $1 * 10^{-1}$;
    * virtual queue 2:
      · **minimum threshold**: 5 Packets;
      · **maximum threshold**: 10 Packets;
      · **maximum dropping probability**: $5 * 10^{-1}$;
  - physical queue 1:
    * virtual queue 0:
      · **minimum threshold**: 10 Packets;
      · **maximum threshold**: 20 Packets;
      · **maximum dropping probability**: $2 * 10^{-2}$;
    * virtual queue 1:
      · **minimum threshold**: 5 Packets;
      · **maximum threshold**: 10 Packets;
      · **maximum dropping probability**: $1 * 10^{-1}$;
    * virtual queue 2:
      · **minimum threshold**: 1 Packets;
      · **maximum threshold**: 5 Packets;
      · **maximum dropping probability**: $5 * 10^{-1}$;
  - physical queue 2:
    * virtual queue 0:
      · **minimum threshold**: 10 Packets;
      · **maximum threshold**: 20 Packets;
      · **maximum dropping probability**: $2 * 10^{-2}$;
    * virtual queue 1:
      · **minimum threshold**: 5 Packets;
      · **maximum threshold**: 10 Packets;
      · **maximum dropping probability**: $1 * 10^{-1}$;
    * virtual queue 2:
      · **minimum threshold**: 1 Packets;
      · **maximum threshold**: 5 Packets;
      · **maximum dropping probability**: 1;

- **the amount of memory allocated to the queues:** Default queue buffer size is 20 packets (Packet size 1 KB) : 20KB per queue;

- **the queues which handle data flows:**
  - **Code point 4**: mapped to physical queue 0 and virtual queue 0;
  - **Code point 5**: mapped to physical queue 0 and virtual queue 1;
  - **Code point 7**: mapped to physical queue 1 and virtual queue 0;
  - **Code point 8**: mapped to physical queue 1 and virtual queue 1;
  - **Code point 9**: mapped to physical queue 1 and virtual queue 2;
  - **Code point 10**: mapped to physical queue 2 and virtual queue 0;
  - **Code point 11**: mapped to physical queue 2 and virtual queue 1;
  - **Code point 12**: mapped to physical queue 2 and virtual queue 2;

6.4.2. Core $\rightarrow$ Edge Configurations.

- **The number of existing queues and the traffic scheduler in use**
  3 physical queues, each implementing 3 virtual queues;

- **The queueing discipline in use and the configuration of each queue:**
  Weighted Round Robin scheduling and RIO-C Active Queue Management:
  - physical queue 0:
    - virtual queue 0:
      - **minimum threshold**: 20 Packets;
      - **maximum threshold**: 40 Packets;
      - **maximum dropping probability**: $2*10^{-2}$;
    - virtual queue 1:
      - **minimum threshold**: 10 Packets;
      - **maximum threshold**: 20 Packets;
      - **maximum dropping probability**: $1*10^{-1}$;
    - virtual queue 2:
      - **minimum threshold**: 5 Packets;
      - **maximum threshold**: 10 Packets;
      - **maximum dropping probability**: $5*10^{-1}$;
  - physical queue 1:
    - virtual queue 0:
      - **minimum threshold**: 10 Packets;
      - **maximum threshold**: 20 Packets;
      - **maximum dropping probability**: $2*10^{-2}$;
    - virtual queue 1:
      - **minimum threshold**: 5 Packets;
      - **maximum threshold**: 10 Packets;
      - **maximum dropping probability**: $1*10^{-1}$;
    - virtual queue 2:
      - **minimum threshold**: 1 Packets;
      - **maximum threshold**: 5 Packets;
      - **maximum dropping probability**: $5*10^{-1}$;
  - physical queue 2:
    - virtual queue 0:
      - **minimum threshold**: 10 Packets;
      - **maximum threshold**: 20 Packets;
      - **maximum dropping probability**: $2*10^{-2}$;
    - virtual queue 1:
      - **minimum threshold**: 5 Packets;
      - **maximum threshold**: 10 Packets;
      - **maximum dropping probability**: $1*10^{-1}$;
    - virtual queue 2:
      - **minimum threshold**: 1 Packets;
      - **maximum threshold**: 5 Packets;
      - **maximum dropping probability**: 1;

- **the amount of memory allocated to the queues:**
  Default queue buffer size is 20 packets (Packet size 1 KB) : 20KB per queue;

- **the queues which handle data flows:**
  - **Code point 4**: mapped to physical queue 0 and virtual queue 0;
  - **Code point 5**: mapped to physical queue 0 and virtual queue 1;
  - **Code point 7**: mapped to physical queue 1 and virtual queue 0;
  - **Code point 8**: mapped to physical queue 1 and virtual queue 1;
  - **Code point 9**: mapped to physical queue 1 and virtual queue 2;
  - **Code point 10**: mapped to physical queue 2 and virtual queue 0;
  - **Code point 11**: mapped to physical queue 2 and virtual queue 1;
  - **Code point 12**: mapped to physical queue 2 and virtual queue 2;

As you can state in **Edge $\rightarrow$ Core Configurations**, the policer configuration and the queue management discipline are much more harsher to CBR Apps generated traffic. By completely dropping all packets further downgraded to that specific type of traffic we are assuring a minimum QoS to the other services present on the network, keeping the requested 30% of capacity to clients with identical characteristics (we considered the maximum CBR bandwidth as 3Mbps), since as you can see the 1Mbps CIR is higher than 30% of 3Mbps. The full capacity delegated to CBR traffic is used by a flow if no other similar client is using its delegated bandwidth, since traffic between 1Mbps and 3Mbps is tagged with yellow code point 11, as requested.

To fully proof the requested capabilities, the simulation results are shown in figures 15, 16, and 17. Denote that, since the simulation scenario is not fully symmetric we included also an bandwidth analysis for **E2 $\rightarrow$ C0** traffic flows on figure 16.

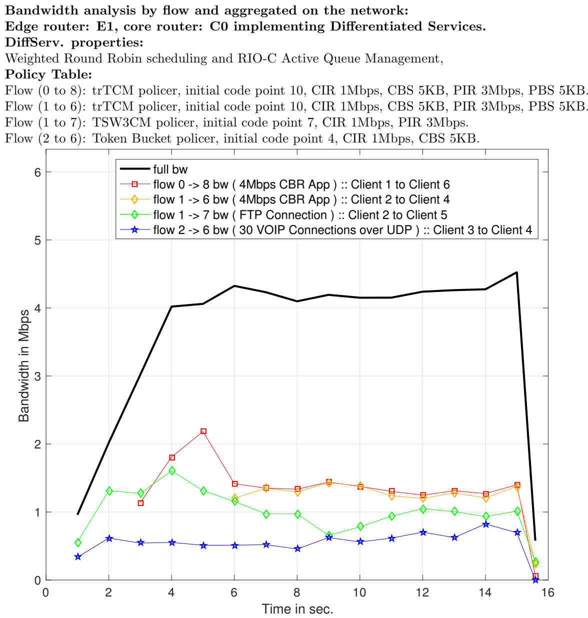

Figure 15. Bandwidth analysis by flow and aggregated, for edge router E1 and core router C0, implementing Differentiated Services.

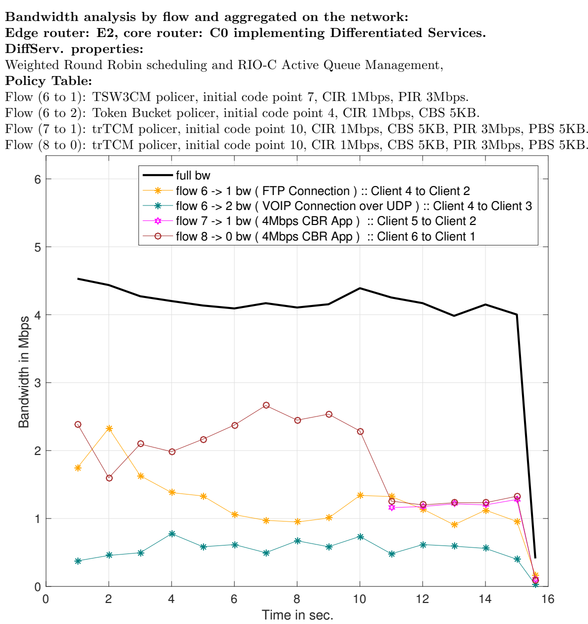

Figure 16. Bandwidth analysis by flow and aggregated, for edge router E2 and core router C0, implementing Differentiated Services.

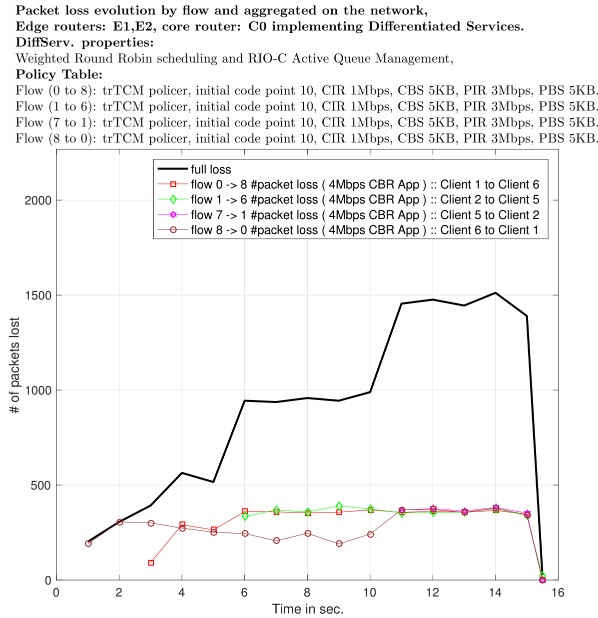

Figure 17. Packet loss evolution by flow and aggregated, with edge/core routers implementing Differentiated Services.

As you can stated by analysing figures 15 and 16, despite the late start, all traffic from the same class has the same equitable bandwidth distribution – as desired. We can also stated, by analysing figure 17, that an increase in the total packets sent by flows only influences the packet drop of its traffic class.

6.4.3. Identification of the queue which suffers higher packet loss. .

Taking into account the simulation results/statistics, presented on tables 8 to 13, we can identify the queue which suffers higher packet loss – physical queue 2 and virtual queue 1 (CBR traffic downgrade (yellow)), from E1 to C0 (Edge $\rightarrow$ Core Configuration). The network behaviour is the expected one. As you can state all UDP and FTP traffic is delivered, and only the CBR traffic exceeding CIR is dropped in congestion.

6.4.4. Statistics for time = 5s. .

TABLE 8. STATISTICS FOR THE QUEUE FROM E1 TO C0 (EDGE $\rightarrow$ CORE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	4250	3918	196	136
4	362	362	0	0
5	1	1	0	0
7	664	664	0	0
8	334	334	0	0
10	1644	1644	0	0
11	1245	913	196	136

TABLE 9. STATISTICS FOR THE QUEUE FROM C0 TO E2 (CORE → EDGE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	3890	3890	0	0
4	362	362	0	0
5	1	1	0	0
7	664	664	0	0
8	333	333	0	0
10	1625	1625	0	0
11	905	905	0	0

6.4.5. Statistics for time = 10s. .

TABLE 10. STATISTICS FOR THE QUEUE FROM E1 TO C0 (EDGE → CORE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	8599	7957	312	330
4	726	726	0	0
5	1	1	0	0
7	1266	1266	0	0
8	792	792	0	0
10	3319	3319	0	0
11	2495	1853	312	330

TABLE 11. STATISTICS FOR THE QUEUE FROM C0 TO E2 (CORE → EDGE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	7938	7938	0	0
4	724	724	0	0
5	1	1	0	0
7	1266	1266	0	0
8	792	792	0	0
10	3308	3308	0	0
11	1847	1847	0	0

6.4.6. Statistics for time = 15s. .

TABLE 12. STATISTICS FOR THE QUEUE FROM E1 TO C0 (EDGE → CORE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	12911	12027	357	527
4	1021	1021	0	0
5	1	1	0	0
7	1864	1864	0	0
8	1264	1264	0	0
10	5016	5016	0	0
11	3745	2861	357	527

TABLE 13. STATISTICS FOR THE QUEUE FROM C0 TO E2 (CORE → EDGE CONFIGURATION)

CP	TotPkts	TxPkts	ldrops	edrops
All	12006	12006	0	0
4	1020	1020	0	0
5	1	1	0	0
7	1861	1861	0	0
8	1258	1258	0	0
10	5012	5012	0	0
11	2854	2854	0	0

7. Conclusion

In today's multi-service networks, more and more complex applications are deployed and QoS-sensitive data is growing in it. In this exploratory essay, we reduced link congestion and end-to-end delays that multi-service networks suffers from.

Our proposed solution implied an full understanding of the network topology (something unfeasible in the real world). This ensured that high-priority traffic had the less loss percentage, and minimised the average percentage of packets lost during packet transmission. This resulted in improved network throughput and better QoS.

However, as stated, this essay has low real-world scenario applications, being considered as an introductory mean to QoS terms and applications in multi-service networks to the authors.

References

[1] network simulator—ns (version 2). [Online]. Available: https://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/

[2] The ns Manual [Online]. Available: http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf

[3] Altman E, and Jimenez T., (2003). NS Simulator for Beginners. Lecture notes. Univ.de Los Andes, Merida, Venezuela and ESSI.Sophia-Antipolis, France.

[4] T. Issariyakul, Introduction to Network Simulator NS2, Springer, 2008.