A Proposal to Improve Network Throughput Using a QoS Building Blocks Approach at Central University of Venezuela

María E. Villapol
Central University of Venezuela
School of Computer Science
Caracas
58-212-6051023
mvillap@strix.ciens.ucv.ve

Eric A. Gamess
Central University of Venezuela
School of Computer Science
Caracas
58-212-6051061
egamess@ciens.ucv.ve

Neudith Morales
Central University of Venezuela
School of Computer Science
Caracas
58-212-6054910
moralesn@ucv.ve

ABSTRACT
Central University of Venezuela is the main university of Venezuela and has approximately 60,000 students and 16,000 staff members. The backbone network connects 11 colleges and many non-academic dependencies; some of them are located outside the main campus and even in other regions of the country. The Internet access is centralized and supported by private links at an aggregated data rate of 14.336 Mbps. Users of the institution network can access most of the Internet services with few or non restrictions. In the last few years, there has been an increasing demand on the use of the Internet services provided by the institution. The consequences of the above are degradation of the Internet access services provided by the institution to users, which claim for better response times and throughput. In this paper, we propose a solution to this problem based on the use of congestion control mechanisms and Internet service approaches proposed by the IETF. So, we start the paper given a brief introduction of such mechanisms and service models; then we detail the problem of the campus network and finally we describe the proposed solution. We conclude that the problem of network congestion may be reduced using existing congestion control mechanisms, which can complement a solution based on network over-provision.

Categories and Subject Descriptors
C.2.3 [**Computer-Communication Networks**]: Network Operations – *Network monitoring*.

General Terms
Measurement, Performance, Experimentation, Theory.

Keywords
Congestion Control, Quality of Service (QoS), Service Models.

LANC 2005, October 10–11, 2005, Cali, Colombia.

1. INTRODUCTION
Central University of Venezuela (In Spanish: Universidad Central de Venezuela, UCV) is the main high level education institution of Venezuela. It has about 60,000 students and 16,000 staff members. The main campus is located at Los Chaguaramos in Caracas, however there are some smaller campuses, research institutes and offices distributed around the city and other cities in the country.

The network topology of the University is an extended star; where the central node is located in the main campus, at Los Chaguaramos. Thus, networks of secondary campuses, research institutes and offices situated outside the main campus are attached to the central node via frame relay links. The Internet access is provided by two *Internet Service Providers* (ISPs) called Reacciun and CANTV as show in Figure 1. Reacciun is an office that heads the research institutes of the national universities. CANTV is the oldest and the most important telecommunication company of the country. The connection with Reacciun is through a unique E1 link and through six E1 links with CANTV. Since CANTV has many routers, one of the links is attached to a router and the other five links are attached to another router. The aggregated transmission rate is about 7*2.048=14.336 Mbps.

In the last few years, the University network has been expanded and currently about 8,000 concurrent users can utilize the network services with few or non restrictions. As a consequent of this growing demand, the network service is degraded during office hours and users experiment high response times and low throughput, in particularly, when they try to access the Internet.

The main efforts of the institution have been focused on buying more bandwidth to the ISPs. The capacity of the links has been increased gradually from 2.048 Mbps about two years ago to 14.336 Mbps recently. Since UCV is a public university with limited annual budget, it can not afford for more bandwidth. However, there has being few efforts for trying to find alternative solutions to improve the network service. Thus the main objectives of this paper are to describe the UCV´s network traffic problem and to present some solutions, which can be implemented to control the problem. These solutions could be also applied to other networks of the region having similar problems.

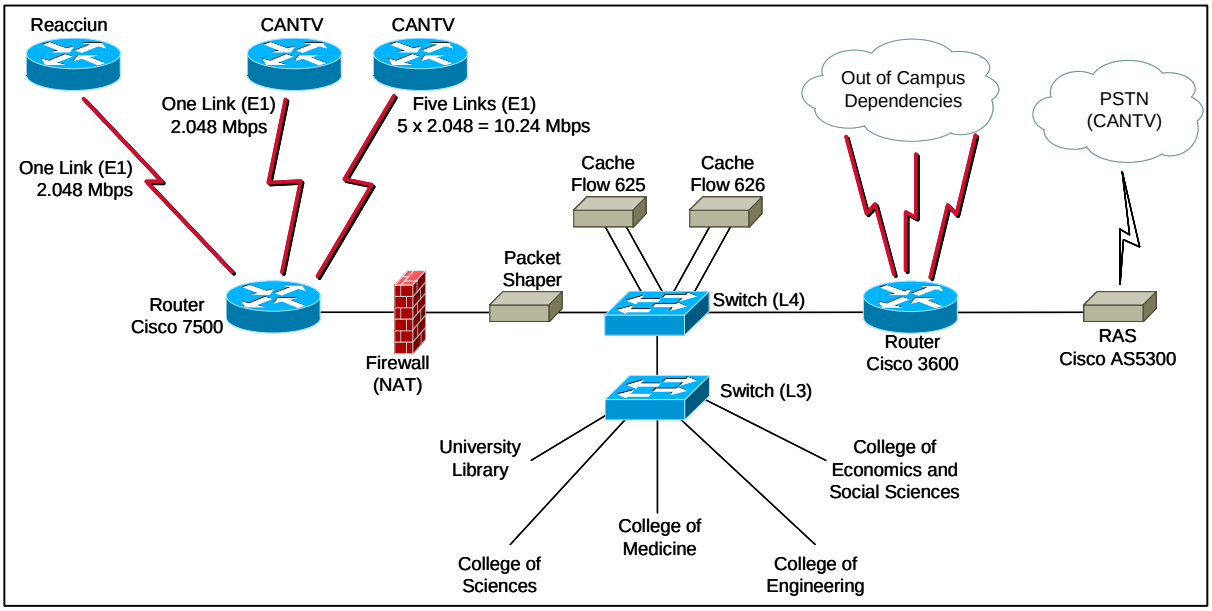


Figure 1: University Network.

The paper is organized as follows. Section 2 presents a description of Internet Services Models and congestion control mechanisms. Section 3 describes de main problems of the UCV network, while Section 4 presents some solutions to the problems. Finally, Section 5 concludes this paper.

2. CONGESTION CONTROL MECHANISMS AND SERVICE MODELS FOR TCP/IP NETWORKS

It is well known, the growth of the network traffic on the Internet as a consequence of the growing number of users and applications. Thus, some points of the network can become congested during the office hours. Network congestion may occur when the network resources (bandwidth, buffers) are not sufficient to support the aggregated demand of users. The consequences of network congestion are low throughput, high delay in delivering data packets, wasting of network resources because of dropped packets and possible network collapse, in which all communications in the entire network stop [10].

The *Internet Engineering Task Force* (IETF), a volunteer organisation that sets the standards for the Internet, and other Internet researchers have worked on improving extending the TCP/IP model to Internet congestion control and Quality of Service (QoS) support. These efforts may be broadly divided into the development and revision of the Internet protocols, the definition of the service models [15] and development of congestion control mechanisms [7][10].

2.1 Service Models

A service model includes a set of mechanisms and protocols for managing network resources in order to avoid network congestion conditions which can degrade the agreed service performance level of applications. The IETF has proposed two Internet Service models. The former is called the *Integrated Services* (IntServ) model [2], and the latest proposal is the *Differentiated Services* (DiffServ) model [1].

2.2 Internet Integrated Services Model

The IntServ model [2] is intended to support real-time and non-real time Internet services. Users are able to explicitly request some quantitative QoS guarantees, so their applications can operate in an acceptable way over a certain period of time. The model provides both a mechanism which conveys users' QoS requirements (reservation protocol) and one which decides if the network can meet those requirements (traffic control). Traffic control functions are performed by the admission control, packet scheduler, and classifier.

The components of the IntServ model interact in order to control the traffic in the network and reserve and negotiate different service classes along the communication path. It comprises a host communicating with an Internet router. The host and router systems are the same except that the application block in the host is replaced by a routing block in the router. Each of those blocks is described below:

a) **Applications**: request specific QoS from the network.

b) **Reservation process**: a set of procedures to reserve resources (eg bandwidth and buffer space) along the path of the data flows. The *Resource Reservation Protocol* (RSVP) has been adopted by the IETF for the IntServ model [3].

c) **Classifier**: classifies IP packets according to a set of service classes and assigns them to different queues.

d) **Packet scheduler**: determines which of the set of IP packets will be served next.

e) **Admission Control**: decides whether there are sufficient resources available to grant the requested QoS for a data flow. A *data flow* is a distinguishable packet stream which

results from a single user/application activity and requires the same QoS.

f) **Policy Control**: decides if the user requesting a reservation is permitted to do so. Policy control mechanisms may involve, for example, the identity of the user and application, traffic and data rate requirements, and security considerations [6].

g) **Routing process**: determines the route along which the packets will be forwarded.

2.2.1 Classes of Service

The *Integrated Services Work Group* has defined several classes of service [2], which are described as follows:

a) **Guaranteed service [6]**: is for guaranteed delay-bound real-time applications. It provides guaranteed data rate and delay. Also, data packets conforming to their traffic specifications will not be discarded because of queue overflow. The guaranteed service only controls the maximum queuing delay. Other delays which are fixed delays such as transmission delay and propagation delays may be determined by the setup mechanisms. This service is intended for applications which have firm time constraints, such as telephony and medical images.

b) **Controlled load service [6]**: corresponds to the predictive real-time service. Nodes (eg routers) which have committed to providing a controlled-load service should offer a service which approximates that provided by a best-effort service under lightly loaded conditions. A high percentage of delivered packets should not exceed a minimum transit delay and should arrive at their destination successfully (ie there is a low probability of packet loss). Controlled-load service may be used for applications such as video conferencing and Internet real-audio.

c) **Best effort service**: corresponds to "elastic" applications and is the current service provided by the Internet.

2.2.2 Resource Reservation Protocol

RSVP is designed to be run on network routers and in end hosts to support a QoS application. It reserves resources for a data flow from the sender to one or more destinations (i.e. multicast destination). Unlike other signalling protocols, RSVP destinations (receivers) request resource reservations. Those requests travel on the reverse path of the data flow by following the pre-established route setup by RSVP [3]. RSVP is also responsible for maintaining reservations on each node associated with the data flow. RSVP uses a *soft-state* approach where the reservation states must be refreshed periodically; otherwise they are automatically removed. The approach accommodates dynamic route changes, dynamic multicast group membership and dynamic QoS changes [3]. RSVP reserves resources for a *session*. A session includes all data flows from one or more senders to the same unicast (one receiver) or multicast destination (multiple receivers).

RSVP reservation requests are defined in terms of a *filter specification (filter spec)* and a *flow specification (flow spec)* [3][6]. A filter spec is used to identify the data flow that is to receive the QoS specified in a flow specification. A flow spec defines the desired QoS in terms of a service class, which comprises a *Reservation Specification (RSpec)*, and a *Traffic Specification (TSpec)*. A RSpec defines the reservation (i.e. desired QoS) characteristics of the flow, for example, the service rate (i.e. the data rate that a data flow can use). A TSpec defines the traffic characteristics of the flow, for example, the peak data rate (i.e. the maximum rate at which the sender is intended to send packets).

RSVP uses several messages in order to create, maintain, and release state information for a session between one or more senders and one or more receivers. Figure 2 shows RSVP operation over a multicast network.

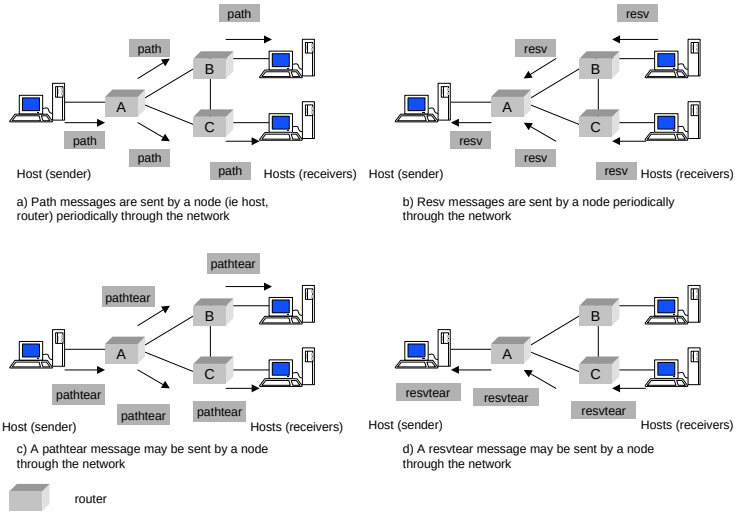


Figure 2: RSVP messages.

In general, sequences of packets traveling in opposite directions may follow different routes In RSVP, reservation requests travel from receivers to the sender(s), in the opposite direction to the user data flow for which such as reservation is being requested. *Path Messages* are used to set up a route for the reservation requests along the same path of the corresponding data flow (Figure 2 (a)). They set up and maintain path information (eg the IP address of the previous host and traffic characteristics of a data flow).

The path is refreshed as a result of either a state refresh timeout or the modification of a path state (as mentioned before). Once a path is established, a node periodically (ie every refresh timeout period) sends path refresh messages (ie Path messages) downstream (Figure 2 (a)).

Resv messages travel upstream from the receiver(s) to the sender (Figure 2 (b)). They carry reservation requests (e.g. for bandwidth and buffers) used to set up reservation state information along the route of a data flow. At any intermediate node, a reservation request may be rejected by Admission Control because there are not sufficient resources to guarantee the requested QoS. Also, reservation requests which arrive at a router are merged. The aim of merging is to control the overhead of reservation messages by making them carry more than one flow and filter specification [3][17]. Thus, the effective filter and flow specifications, which are carried in a reservation message, are the result of merging

reservations from several requests. Merging is a complex process [3] which will not be described further here.

The reservation is refreshed as a result of either a state refresh timeout or the modification of a reservation state (as mentioned before). Like path states, reservation states need to be refreshed. Thus, a receiver periodically sends reservation refresh messages (ie Resv messages) to the sender (Figure 2 (b)).

RSVP tear down messages are intended to speed up the removal of path and reservation state information from the nodes. They may be triggered because a state timeout occurs (as explained before) or an application wishes to finish a session (i.e. service preemption). A *PathTear message* travels downstream from a sender to the receiver(s) and deletes any path state information and dependent reservation state associated with the session and sender (Figure 2 (c)). A *ResvTear message* travels from a receiver to a sender and removes any reservation information state associated with one or more data flows (Figure 2 (d)).

In addition, there are two error messages, *Path Error* and *Resv Error*, which are used to report problems associated with processing or installing Path/Resv information or to report administratively defined constraints imposed on the setup of a reservation state. They travel hop-by-hop from the point where the error was found.

Optionally, a receiver may ask for a confirmation for its reservation by including a RESV conformation object in the Resv message (ie reservation request). A *ResvConf message* is used to notify the receiver that the reservation request was successful.

2.3 Differentiated Services Model

The main problem of the IntServ proposal is that it is not scalable across large networks. Thus, another working group developed a service model called *Differentiated Services* (DiffServ). The DiffServ model is intended to solve the scalability problem by aggregating traffic. Large flows with similar service requirements are aggregated. Traffic entering a network is classified and marked in order to receive a specific quantitative or qualitative QoS.

DiffServ redefines the IPv4 *TOS* (Type Of Service) octet [5] and the IPv6 *Traffic Class* octet [9]. The new defined field is called, *Differentiated Service field* (DS field). The 8-bit DS field is divided into a DS codepoint and a currently unused (CU) fields. Packets that enter the DiffServ network are marked with a DS codepoint (DSCP). The CU field is reserved. A collection of packets which have the same DS codepoint (DSCP), travel in the same direction and traverse the same link are referred as a *behaviour aggregate* (or traffic aggregate) [1].

The Diffserv architecture comprises a number of functional elements known as per-hop behaviours, packet classifiers and traffic conditioners. They are implemented in several nodes (eg routers) along the network. *A per-hop behaviour (PHB)* is the mean by which a sequence of packets obtains some level of service. It may be seen as the differential treatment which a packet will receive. It may be defined in terms of network resources (ie buffer), traffic characteristics (eg delay, loss), etc ..[1], and it is implemented in nodes through several queue management and packet scheduling mechanisms.

Figure 3: Differentiated services network.

A *packet classifier* starts by selecting the packets in a input traffic stream by using either the DS codepoint of the packet header or a combination of one or more header fields, such as IP destination address, IP source address, DS field, and IPv6 flow ID and/or other packet attributes. After that, it forwards them to an element of traffic conditioner for further processing. Thus, a classifier splits an input traffic stream into one or more output streams.

A *traffic conditioner* is an entity which performs control functions intended to enforce traffic rules. It may contain meters, markers, shapers, and droppers. These components are described briefly as follows:

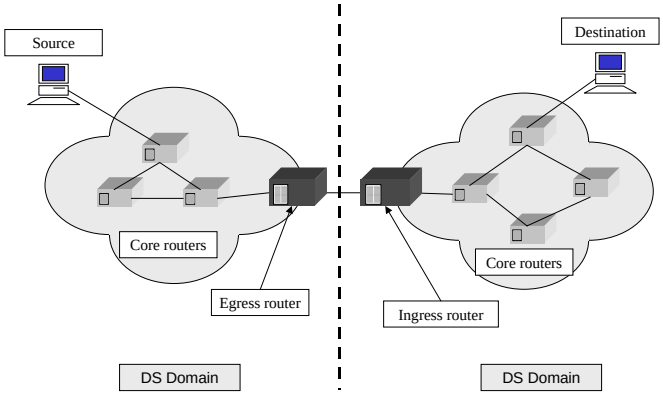
a) **Meters**: are used to monitor the arrival time of packets in order to verify that they are conforming to their traffic characteristics in the traffic characteristic agreement (ie traffic profile). The meter provides the resulting information to the other components of the traffic conditioner.

b) **Markers**: set the DS codepoint field in the IP packet to a particular value. For example, it may mark packets which have been classified by the classifier as a member of a particular flow. It also may re-mark previously marked packets which, for example, are not conforming to their traffic profile (see meters).

c) **Shapers**: delay packets in a traffic stream by using buffers, so the traffic conforms to its traffic profile.

d) **Droppers**: discard some or all the packets in a traffic stream so that the traffic stream conforms to its traffic profile.

The functional elements of the DiffServ architecture may be implemented in different nodes in a network; they are shown in Figure 3. A *node* (eg a router) which is enabled to support differentiated services functions is called a DS node. A DiffServ specification classifies the nodes according to their location in a DiffServ region and the functions they perform. The following terminology applies to a DiffServ network.



A *DS domain* includes a set of DS nodes which operate with a common set of differentiated service provisioning policies and share the same boundary nodes. A differentiated service provisioning policy defines how traffic handling mechanisms are configured in core and edge nodes to provide a range of services. DS boundary nodes, also called edge nodes, interconnect a DS domain with either another DS domain or a non-DS domain. Traffic enters a domain through a DS edge ingress node and leaves the domain from a DS edge egress node. DS nodes in a domain which may be connected to boundary nodes are called interior nodes or core nodes. For example a campus or corporate network may be a DS domain.

The *core nodes* implement limited differentiated services functions. They apply the appropriate PHB to packets in a traffic stream based on their DSCP. Edge nodes, in addition, perform traffic classification and conditioner functions.

Providers (DS domain) and customers (eg local users and adjacent networks) must negotiate agreements with respect to the level of service which will be given to customers. Such agreements are called *Service Level Agreements* (SLA). A SLA is a complex contract which includes overall service features such as network availability guarantees, payment models, billing mechanisms, etc…[15]. The *Service Level Specification* (SLS) is part of a SLA. The SLS comprises the technical specification of the service. It includes, for example, *Traffic Conditioning Agreement* (TCA) parameters, encryption services, routing constraints, and pricing and billing mechanisms. A TCA specifies classifier and conditioning rules as well as traffic stream characteristics (ie traffic profile) such as rate and burst size.

The DiffServ Working Group has defined several classes of services so far [15]. They are defined in terms of PHBs and include Expedited Forwarding, Assured Forwarding, and Best-Effort Forwarding.

a) **Expedited Forwarding**: provides a virtual leased line end-to-end service, which is characterised by low loss, low latency, low jitter, and assured bandwidth. It is also called "premium service". It may suit applications such as video broadcast, voice-over-IP, and virtual private networks.

b) **Assured Forwarding**: provides a service based on an "expected" usage profile. This profile indicates the level of performance (service assurance) uncertainty the user may tolerate (user expectation), more than a strict guarantee (like RSVP may provide). During periods of congestion some packets may still be dropped, but it may be acceptable for the user. Heinanen et al [8] define several assured forwarding classes, and within each class also define several "drop precedence" values. The drop precedence values determine which packets are likely to be dropped during periods of congestion. In order to provide a level of forwarding assurance, a certain amount of resources (bandwidth and buffer space) are allocated for an assured forwarding class, and each IP packet must be marked with a drop precedence value.

c) **Best-Effort Forwarding**: is the default service given when there is no other agreement in place. It corresponds to the common best-effort service with no QoS guarantees.

2.4 Congestion Control Mechanisms

The service models described in the last sections represent the major efforts of the IETF for improving the Internet services; they main objective was to offer a better service to emerging real-time and multimedia applications. However, the vast majority of Internet traffic is best effort [7]. Also, it has been demonstrated that many real-time and multimedia applications such as packet audio and video conference are able to adapt to the network load changes using different mechanisms [7]. Thus in the following sections, we describe a set of congestion control mechanisms which may be used for best effort traffic. We use the congestion control mechanism classification presented by Grevos et al [7]; they divide these mechanisms based on where they are implemented (hosts or routers) as follows: *congestion control mechanism based on hosts and based on routers.*

2.4.1 Congestion Control Mechanisms Based on Hosts

Traditionally, the congestion control mechanisms have been implemented at hosts. A source node may respond to network congestion conditions by decreasing the data rate at which the traffic is injected to the network. Thus, this mechanism is known as control flow. Feedback information from the network is usually required by the source to adjust its traffic data rate. The control flow mechanisms are divided in: *open loop* and *closed loop* [7][16]. The open loop control flow mechanism do not use any feedback information from the network, otherwise the source traffic is described in terms of some parameters such as, the burst size and average data rate. This information is provided to the network, which may reserve the resources according to the information provided by the source using admission o policy control mechanisms. The open loop control flow is used in Intserv (see Section 2.2).

The closed loop control flow mechanisms use feedback information provided by the network. The source node then may adapt its traffic to the network load changes. The slow start algorithm incorporated in TCP is an example of a closed loop control. It uses windows, whose sizes are changed based on the feedback information provided by the *Round-Trip-Time* (RTT) of the TCP ACK segments [12].

2.4.2 Congestion Control Mechanisms Based on Routers

The routers can know how congested they are and then run some resource management mechanisms to handle this situation. They are classified in: *scheduling mechanisms* and *buffer/queues management mechanisms*.

Scheduling Mechanisms

The scheduling mechanisms determine which packet in a queue will be served next. The simplest scheduling mechanism is the *First-in-First-Out* (FIFO) and is also the default service discipline used on the Internet routers. FIFO does not protect any packet traffic, does not provide fairness and does not provide any packet priority. A service discipline which can improve FIFO by providing fair allocation is called *Generalized Processor Sharing* (GPS). In GPS, packets are served as they were located in different logical queues. Each non empty queue is visited in turn and an infinitesimally small amount of data from each queue is sent. Each queue may have a weight which determines the amount of data from the queue which is transmitted. GPS can not be implemented in the practice, so some approximation has been proposed.

Weighted Fair Queuing (WFQ) [13] is an approximation of GPS bit by bit. The next packet to be transmitted is the packet which has the smallest finish time. The finish time is defined as the time when the packet would complete service if it would have been served by GPS. Since WFQ is computational expensive to implement, some variations have been proposed, such as *Self-Clocked Fair Queuing* (SCFQ) and *Worst-Case WFQ* (W2FQ) [7].

Unlike the other schemes, the *Class-Based Queuing* (CBQ) does not serve an individual flow, but rather a complete class of flows, having similar QoS requirements. This scheme provides a hierarchical way to treat the queues, which determines how the queues at the top level will be attended, then how the flows of the same class will be served, and so on. The flows of the same class may be served using a *Fair Queuing* (FQ) scheme, where the packets of each flow are queued in a separated queue. The queues are attended in a round robin order. Unlike round-robin scheduling, it takes into account the count of bytes serviced from each flow. A *Class-Based WFQ* has also been proposed in [12].

Buffer Management

The main objective of a buffer management is to determine how a buffer is shared among different packets going through the router. The most common buffer management mechanisms are *shared buffer* and *per-flow allocation* [7]. In the former (the simplest one) the buffers are used under the basis on first come first use. The consequent of using this mechanism is that a data flow can occupy all the buffers by sending faster than other flows.

In per flow allocation, buffers are managed by flows. The system keeps track of the utilization of the buffer and the packets are dropped based on the occupancy level of each flow. Per flow allocation is expensive and no well scalable in terms of processing requirements.

Queue Management

Queue management mechanism control the length of the queues and which flows occupy them. There are two approximations for queue management: *queue management for congestion recovery and active queue management for congestion avoidance*. The former acts when the queues are full. For example, Tail Drop is usually used on the Internet. It drops the last packets of a full queue. It reacts too late, so it keeps the queue filled. In addition, it introduces the global synchronization in the network, that is, the sources reduce simultaneously theirs data rates when they realize that their packets are dropped, so their actions get synchronize. The consequents are that the link utilization is decreased and some sources may monopolize the queue space.

Drop From Front is another queue management mechanism which consists in dropping packets from the front of the queue when a new packet arrives. It improves the fair packet dropping and avoids the problem of monopolization of the queue space. Finally, *Random Drop* chooses randomly the packet from the queue, which will be dropped when a new packet arrives. It does not have the global synchronization problem and avoid the monopolization of the queue space.

The active queue management for congestion avoidance acts to avoid congestion conditions. For example, *Early Random Drop* (ERD) drops packets chosen randomly and uniformly when the network congestion is anticipated. A congestion detection mechanism is needed. A simple mechanism is to use a threshold in queue length. When this limit is reached, the packets can be dropped. Although ERD improves mechanism such as tail drop, ERD routers are biased against bursty traffic, that is, the dropping packet algorithm is influenced against the bursty traffic source as compared to other sources which generate the same average load to the network [7][13].

Random Early Detection (RED) marks the packets early in order to control the size of the queue. When the algorithm detects incipient congestion by looking the average size of the queue, the packets are marked or dropped based on a dropped probability. The calculation of the probability is based on the average size of the queue. RED is based on two algorithms: one which calculates the average size of the queue, and so it determines the burst size allowed. The second algorithm defines how frequently the packets are dropped (or marked) based on the congestion level at the moment of congestion. It is difficult to set the RED parameters for the different traffic conditions, so some approximations have been formulated (see [10]).

3. DESCRIPTION OF THE MAIN PROBLEMS OF THE UCV NETWORK

3.1 Network Configuration

Figure 1 shows an overview of the network of the UCV. The network topology is an extended star, where the edge router (Cisco 7500) is located in the main Campus, at Los Chaguaramos. As we mentioned in Section 1, the Internet access is provided by Reacciun and CANTV at an aggregated data rate of 14.336 Mbps. The backbone network ties together *Intermediate Distribution Facilities* (IDFs) to the *Main Distribution Facility* (MDF). Fiber optic cables were used with a bandwidth of 100 Mbps. IDFs were placed in four colleges (Sciences, Engineering, Medicine and,

Economics and Social Sciences), as well as in the University Library. The purpose of the backbone network is to facilitate fast and easy information exchange and, resources sharing among all the interconnected dependencies (colleges and out of campus dependencies) of the University and to provide good connectivity to the outside world. In each college, the network at the different schools is connected to a layer 3 switch in the IDF. This layer 3 switch is connected to another layer 3 switch in the MDF. So traffic from college to college passes uniquely through the layer 3 switch of the MDF. Traffic from college to the Internet passes through the layer 3 switch before arriving to the layer 4 switch (layer 4 switches can forward traffic based on protocols). If it is an HTTP request (TCP port 80), the layer 4 switch sends it to the cache flows. If one of the cache flows has the requested Web page in memory, then it is sent to the HTTP client (Netscape Navigator o Internet Explorer). Otherwise, one of the cache flows looks for the Web page in the Internet and sends it to the HTTP client. If the traffic is not Web based, then the layer 4 switch sends it to the packet shaper (a packet shaper is a traffic management system that monitors and controls traffic). Our packet shaper has some basic rules of QoS. For example, HTTP traffic has a biggest bandwidth than other protocols. After the packet shaper, packets will be received by the firewall which can drop them. Finally, they will arrive to the edge router.

3.2 Network Load

To measure the links performance, we used *Multi Router Traffic Grapher*[1] (MRTG) which is a tool to monitor the traffic load on network-links. MRTG used *Simple Network Management Protocol* (SNMP) [4] to get some variables of the *Management Information Base* (MIB) related to traffic, from manageable switches and routers to generate HTML pages containing GIF images which provide a live visual representation of this traffic. MRTG shows a detailed daily view of the network traffic, as well as the last seven days view, and the last five weeks view. Figure 4 shows the network traffic of one of the E1 links between the University and CANTV during two working days. The other E1 links have a similar behaviour. As we can observe, the link is full duplex. The outbound traffic is about 500 Kbps during office hours, which is low, compared to the capacity of the link (2.048 Mbps). The inbound traffic reaches its top early in the morning and start to decrease in the late evening. During office hours, the inbound traffic average is about 1.897 Mbps which seems to show that there is network congestion for incoming packets.

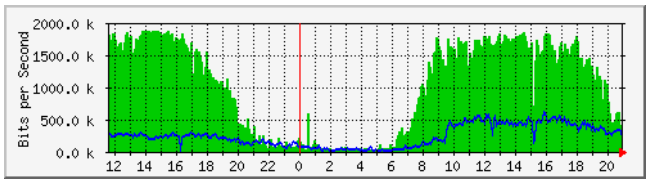


Figure 4: Network traffic obtained during two consecutive working days.

[1] http://www.mrtg.org

Figure 5 shows the network traffic of one of the E1 links between the University and CANTV during a week. The outbound traffic during office hours is about 500 Kbps from Monday to Friday and, about 300 Kbps for the weekend. As we can observe, the E1 link is congested by inbound traffic during office hours from Monday to Friday, while it is almost idle during weekend.

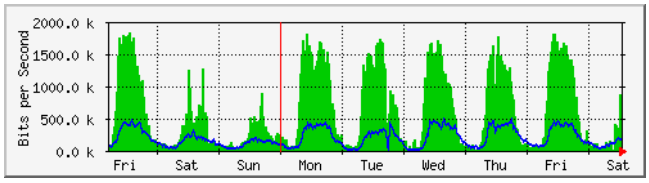


Figure 5: Network traffic obtained during a week.

To have a more specific idea about congestion, we download a file using *File Transfer Protocol* (FTP) from different sites of the Internet. We choose two sites in the United States, two sites in Europe and one site in Japan. The idea of choosing sites from all over the world is to minimize the effect of traffic congestion that is not due to the links between the University and CANTV, over the results. The chosen file has a size of 5 MB and we measured the download time at different hours. Table 1 shows the results that we obtained. Grey rows contain the download time for working days, while white rows have the results of weekends. As we can see, the download time is more than 40 minutes before 5:00 PM during working days and falls after 5:00 PM. It is noticeable that the download time is never over 3 minutes during weekends. These measurements seem to strengthen our hypothesis that we have a problem of congestion between the University and CANTV during office hours from Monday to Friday.

Table 1: Download time of a 5 MB file at different hours.

	10:00 AM	12:00 M	3:00 PM	5:00 PM	9:00 PM
jungle.metalab.unc.edu	58:34	56:22	1:10:22	17:16	8:11
	1:34	1:38	1:20	1:27	1:19
fedora.cat.pdx.edu	41:34	40:09	42:31	13:33	8:47
	1:59	1:41	1:42	1:38	1:45
ftp.lip6.fr	48:57	52:21	1:15:02	18:12	10:04
	2:32	2:48	2:17	2:46	2:05
ftp.nluug.nl	52:26	50:41	1:08:23	13:39	9:58
	2:42	1:57	2:21	2:00	2:07
ftp.nara.wide.ad.jp	50:27	1:02:41	55:51	17:16	10:48
	2:33	2:38	2:25	2:50	2:43

To show that the congestion is situated in the E1 links which communicate the router of the University to the routers of CANTV, and not in the local area network of the University, we took the *Round Trip Time* (RTT) and the percent of packet lost between a computer of the College of Sciences and the different interfaces of the routers of CANTV. The size of the IP packets than were sent and received was exactly of 1500 bytes (The Path MTU), so we used the maximum size allowed by the path before fragmentation. We measured the data at different hours (See Table 2). For each hour, the first column contains the RTT in milliseconds, while the second has the percent of packet lost.

Grey rows indicate working days, while white rows represent weekends. We also took these data between the same computer of the College of Sciences and the router of the University. Those latter data, relative to the local area network, are in the last two rows.

Table 2: RTT Time (in milliseconds) and packet lost (in percent).

	10:00 AM		12:00 M		3:00 PM		5:00 PM		9:00 PM	
Link 1	486	36%	449	43%	477	40%	420	17%	67	0%
CANTV	57	0%	71	0%	84	0%	57	0%	54	0%
Link 2	436	33%	463	34%	491	37%	431	15%	71	0%
CANTV	62	0%	59	1%	88	0%	60	0%	52	0%
Link 3	472	40%	468	33%	475	35%	412	16%	63	0%
CANTV	58	0%	64	0%	62	0%	54	1%	57	0%
Link 4	435	38%	447	40%	480	37%	435	18%	80	0%
CANTV	72	0%	61	0%	75	0%	55	0%	56	1%
Link 5	430	41%	462	37%	471	39%	428	15%	64	0%
CANTV	58	0%	73	0%	91	0%	50	0%	65	0%
Link 6	457	40%	480	36%	466	38%	440	18%	69	0%
CANTV	64	0%	72	0%	97	0%	68	0%	70	0%
Router of the University	22	0%	20	0%	21	0%	18	0%	20	0%
	21	0%	20	0%	17	0%	21	0%	18	0%

From this experiment, we can see that there is almost no packet lost from the College of Sciences to the router of the University (last two rows); that is, there is no packet lost in the local area network, even in working days. Similar results were obtained from the other colleges. The packet lost is very important for the packets that go from the College of Sciences to the different interfaces of the routers of CANTV, during working days before 5:00 PM. It is over 33%. So, we can concluded than the routers of CANTV drop a lot of inbound packets during office hours since there is not enough resources to send them to the router of the University. The RTT of a packet from the College of Sciences to the routers of CANTV is around 470 ms, during working days before 5:00 PM, while it is about 60 ms when there is no much traffic. The difference is significant and it is because packets have to stay a long time in the egress buffers of the routers of CANTV before being sent to the router of the University.

3.3 Quality of Service

Any communication device of the UCV network does not provide quality of service. However, recently, a packet shaper device has been installed as shown in Figure 1. It allows controlling the share of the links by defining per-protocol quotas to the bandwidth available, having the HTTP traffic the highest quota. The packet shaper only controls the Internet traffic at the egress point of the network (see Figure 1).

3.4 Fault Tolerance

Since the network topology is an extended star (see Figure 1), any failure of a link between an IDF and the main switch (see switch L3 in Figure 1) will stop providing services to the users connected through such link. Also, the main switch is a bottleneck, that is, all the traffic going to the Internet goes through this switch. Communications between users located in different colleges go through this switch also.

3.5 Middle Boxes

In the last years, several components have been incorporated to the data network. They have different functions. Figure 1 shows these components. The firewall is intended to protect the network against some networks instructions. The packet shaper controls link sharing. The cache flow is intended to improve the Internet response time by caching the most frequently visited web pages. We measured the delay which can experiment a packet passing through the middle boxes, with the aid of the ping command. The average RTT between an IDF and the switch (L3) is 3 ms; the average RTT between an IDF and the router of the University (Cisco 7500) is 20 ms; Thus, the middle boxes introduce a packet delay of about 8.5 ms (i.e. (20-3) ms/2), which is considerable.

3.6 Use of the Network

Users of the data network of the UCV can use services with few or non restrictions. Recently, the University has approved a document called *Policies for the Rational Use of the Network Resources and Internet Access* [14]. The main objectives of these policies are to establish the use of the Internet access and claim for a better use of the network resources. Since, these polices are general, the University is also working on developing some regulations based on the document.

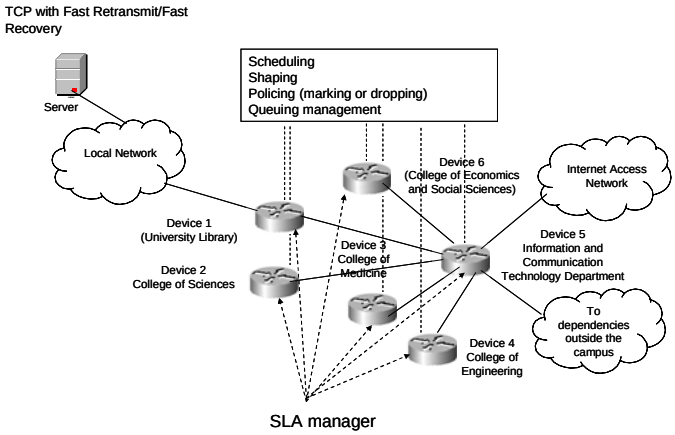
Figure 6: Proposed QoS network design.

4. PROPOSED SOLUTION

As mentioned before, over-provisioning has been the main mechanism to reduce the network congestion problems of the UCV's network. Over-provisioning presents several disadvantages. The main one is that the University is a public institution with a limited annual budget, so it can not afford for more bandwidth due to its high cost in Venezuela. Also, over-provisioning neither ensures the necessary QoS; the lack of some QoS guaranties has lead to poor application performance (in particular for real time and multimedia applications), as shown in Section 3.

In this section, we propose a complementary solution to the network congestion problems. It is described following the *QoS building blocks framework* proposed in [11]. In this framework, the traffic and congestion control function such as, shaping, classification, marking, scheduling (see Section 2) constitute buildings blocks for achieving QoS at the level packet. Unlike the Intserv and Diffserv models (see Section 2.2 and Section 2.3), the QoS building blocks framework provides a more practical QoS network design option, since the network engineers and designers can select among a pool of building blocks and associated mechanisms that best meet the requirements of the organization (such as architecture, resources and technologies required by users and applications, capabilities of the network devices and, the complexity and cost of the targeted solution).

Figure 6 shows the mechanisms which should be deployed and how they may be distributed in the network. For simplicity we have omitted some the components of Figure 1 and only shown one server (there are several servers in most of the Colleges). The mechanisms and components are described in the following paragraphs. We initially classify the existing and coming applications of the institution and then describe the building blocks which may be considered in the design of the University's QoS network. Finally, we present other issues which may be relevant in order to improve network performance and fault tolerance.



4.1 Services and Applications

Currently, the distributed applications of the University are treated as best effort. We propose to classify the existing and coming applications (in *italics*) as shown in Table 3. They have been categorized as guaranteed, controlled load and best effort services (see Section 2.2.1).

Table 3: Classification of the existing and coming applications.

Guaranteed service
Video Conferencing *VoIP*
Controlled-load service
Corporative applications (student management, accounting system)
Best effort service
HTTP Browsing File transfer, etc.

4.2 Policy Management

Given the number of users of the UCV's network and the different user network requirements, a policy management block such as SLA and SLS (see Section 2.3) should be deployed. The SLA defines the characteristics of the service offering and the responsibilities of the parties involved for using the offered service. The technical characteristics of the offered service are given in the SLS, such as scope of SLS, flow identification (e.g. DSCP), traffic conformance and characteristics, service guaranties. The SLA should describe the characteristics of the service which best adapts to the colleges and dependencies connected to the network (see Figure 1). For example, some colleges may want to put a per-user quota on the bandwidth available for Web services based on the function of the user in the institution (e.g. lecturer, student, and administrative staff).

4.3 Queuing and Scheduling

For the University network to supports the applications shown in Table 3: the guaranteed service applications should be allocated a guaranteed portion of the network resources, the controlled-load service applications should be allocated minimum amount of resource so they can achieve their functionality, and the rest of the applications should be treated a best effort. Thus, a scheduling block should be deployed in the network with a class-based queuing mechanism, so the application traffic is classified and serviced by separate queues to which the scheduling algorithm assigns the desired priority and resources according to the service level agreements (see Section 4.2). Figure 6 shows where the scheduling mechanism should be implemented.

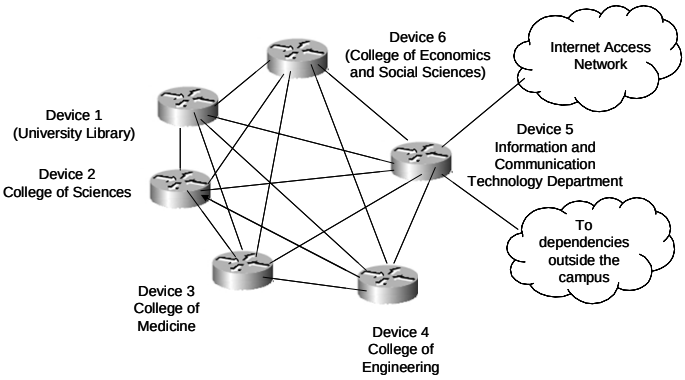


Figure 7: Proposed network topology.

Since queue management achieves the desired results only when applied to traffic responsive to congestion control (e.g TCP), it should not be applied to queue servicing VoIP traffic [11]. A queue management block with RED or any other AQM mechanism (see Section 2.4) should be deployed in the network for traffic responsive to congestion control.

4.4 Shaping and Policing

Shaping and policing blocks should be set up in order to control the rate of outgoing traffic toward enforcing a particular traffic according to the established SLA and monitor, forward or drop (or marking) packets to ensure that the traffic entering the network remains complaint to a predefined profile (according to the SLA). Figure 6 shows where the shaping and policing mechanisms should be employed.

4.5 Signalling and Admission Control

End users of applications such as VoIP and Video conferencing need to signal their request for call establishment. Thus, a signalling building block needs to be deployed. The implementation of this block depends on the implementation of the application (e.g. some signalling protocols which may be use are RSVP, SIP, H.323). A call request which can carry some QoS parameters may be admitted or rejected depending on the availability of network resources, so an admission control block (see Section 2.2) also needs to be set up in the network.

4.6 Congestion Control

In addition to the congestion control mechanisms mentioned before, end-to-end congestion control mechanisms should be set

up. The most common end-to-end congestion control mechanisms have been implemented in TCP. For instance Fast Retransmit and Fast Recovery mechanisms have been incorporated in the Reno and NewReno improvements to TCP [12]. It complements the queue management scheme. Thus, the devices at the end side of the network such as University's servers should incorporate these mechanisms (see Figure 6).

4.7 Other Issues

In order to improve fault tolerance of the backbone network, we propose to modify the current network topology as shown in Figure 7. The topology is a mesh which connects each device (IDFs) to each other. The current fibre optical network can easily support such topology. A routing algorithm such as RIP or OSPF [12] should be deployed in order to establish the routing policies. This solves the bottleneck problem of the current network (see Section 3.4) and improves the exchange of packets in the Intranet.

5. CONCLUSIONS

In this paper, we described the Internet service models as well as the congestion control mechanisms in some details. Then, we present the traffic problems of the UCV's network. As the consequent of these problems the packet experience long delay and the rate of packet loss is high, so the users usually suffer long response time during office hours. In addition, the network topology is an extended star; the layer 3 switch in the MDF which connects the IDFs is a bottleneck. In trying to solve the problem, network engineers have included several components, located between this switch and the edge router of the University. They introduce a considerable packet delay. The problems get worts because the University has established none or few restrictions to users.

Over-provisioning is the main mechanisms used to reduce the traffic problems so far (the capacity of the Internet access links has been increased gradually from 2.048 Mbps about two years ago to 14.336 Mbps recently). This is a costly solution. Thus, we propose a solution based on a QoS building blocks framework, which allows the network engineering and designers to select among several blocks such as shaping, admission control, signalling, which best suit the necessities of the UCV's network users. Although some of these mechanisms are identified in the Internet Service models, the QoS building blocks framework provides a more practical framework since it allows several combinations of building blocks (different from the ones suggested by the standard service models) according to the architecture, the network resources and technologies, the users requirements and, the capabilities of network devices with respect to QoS. We also propose some changes to the network topology which may improve the fault tolerance of the network when a network link fails.

Future work may include the characterization of the UCV's network traffic and the simulation of several scenarios based on the solution proposed in this paper. In addition, a SLA which will include the requirements of the network users and the services offered by the institution and ISPs should be developed.

6. REFERENCES

[1] Blake S. et al. An Architecture for Differentiated Services, *RFC 2475, IETF*, December 1998.

[2] Braden R., Clark D., and Shenker S. Integrated Services in the Internet Architecture: an Overview. *RFC 1633, IETF*, June 1994.

[3] Braden R. et al. Resource Reservation Protocol (RSVP) - Version 1: Functional Specification. *RFC 2205, IETF*, September 1997.

[4] Case J., Fedor M., Schoffstall M., and Davin J. A Simple Network Management Protocol (SNMP). *RFC 1157, IETF*, May 1990.

[5] Defense Advanced Research Projects Agency. Internet Protocol, DARPA Internet Program, Protocol Specification, *RFC 791, IETF*, September 1981.

[6] Durham D. and Yavatkar R. Inside the Internet's Resource Reservation Protocol. *Wiley*, USA, 1999.

[7] Gevros P., Crowcroft J., Kirstein P., and Bhatti S. Congestion Control Mechanisms and the Best Effort Service Model. *IEEE Network*. May/June 2001. pp 16-26.

[8] Heinanen J., Baker F., Weiss W., Wroclawski J. Assured Forwarding PHB Group, *RFC 2597, IETF*, June 1999.

[9] Nichols K., Blake S., Baker F., and Black D. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, *RFC 2474, IETF*, December 1998.

[10] Ryu S., Rump C., and Qiao C. Advances In Internet Congestion Control. *IEEE Communications Surveys & Tutorials*. Third Quarter 2003, Volume 5, No. 1. pp 28-39.

[11] Soldatos J, Vavias E and Kormentzas G. On the Building Blocks of Quality of Service in Heterogeneous IP networks. *IEEE Communications Surveys and Tutorials*, First Quarter 2005, pp 70-89.

[12] Stevens R. TCP/IP Illustrated, Volume 1. *Addison Wesley*. 1994.

[13] Stallings W. High-Speed Networks and Internets: Performance and Quality of Service. Second edition. *Prentice Hall*. 2002.

[14] Universidad Central de Venezuela. Políticas para el Uso Racional de los Recursos de la Red Corporativa de Datos y del Acceso a Internet. Noviembre 2003.

[15] Villapol M.E. and Billington J. Internet Service Quality: A Survey and Comparison of the IETF Approaches, *Telecommunication Journal of Australia*, 2000, Volume 50, No. 2, pp 57-69.

[16] Yang C. and Reddy A. A Taxonomy for Congestion Control Algoritms in Packet Switching Networks. *IEEE Network*. July/August 1995.

[17] Zhang L., Estrin D., and Zappala D. RSVP: A New Resource Reservation Protocol, *IEEE Network Magazine*, September/Octuber, 1993, Volume 7, pp 8-18.