Controlling Access to Preserve QoS in a Self-Aware Network

Erol Gelenbe
Imperial College London
Intelligent Systems and Networks Group
Electrical & Electronic Engineering Dept.
SW7 2BT London, UK
e.gelenbe@imperial.ac.uk

Georgia Sakellari
Imperial College London
Intelligent Systems and Networks Group
Electrical & Electronic Engineering Dept.
SW7 2BT London, UK
g.sakellari@imperial.ac.uk

Maurizio D'Arienzo
Seconda Università degli studi di Napoli
Via del setificio 15
San Leucio (CE)
81020, Italy
maudarie@unina.it

Abstract

Multimedia traffic and real-time applications created a need for network Quality of Service (QoS). This demand led to the development of autonomous networks that use adaptive packet routing in order to provide the best possible QoS. Admission Control (AC) is a mechanism which takes those networks a step further in guaranteeing packet delivery even under strict QoS constraints. This paper describes a measurement-based admission control algorithm which decides whether a new connection can be served without affecting the existing users of the network, based on the multiple QoS metrics that the users of a Self-Aware Network have specified. Our algorithm promises QoS throughout the lifetime of all accepted connections in the network. The impact that the new call will have, on the QoS of both the new and the existing users, is estimated by sending probe packets and monitoring the networks by exploiting its self-awareness. The decision of whether to accept a new call is made using a novel algebra of QoS metrics, inspired by Warshall's algorithm, which looks for a path with acceptable QoS values that can accommodate the new flow. In this paper we describe the underlying mathematical principles and present experimental results obtained by evaluating the method in a large laboratory test-bed operating the Self-Aware Cognitive Packet Network (CPN) protocol.

1. Introduction

The current "best effort" Internet architecture does not secure the Quality of Service (QoS) that multimedia traffic and real-time applications, such as video on demand, Internet telephony (Voice-over-IP), remote medical diagnosis and treatment, and online trading systems require in order to function properly. This need led to the development of Self-Aware Networks [13] that use adaptive packet routing protocols, like the Cognitive Packet Network (CPN) [18], to address QoS and provide reliable service to their users. But even in these networks, congestion is a factor that can lead to unstable and unreliable situations which will affect the service quality. Admission Control (AC) provides a solution to traffic congestion and promises a stable and reliable network which will guarantee packet delivery under QoS constraints. A proposal of a measurement-based AC, which exploits the ability of CPN to collect QoS information on all links, is the purpose of the work presented in this paper. Our scheme bases its decision of whether to accept a new connection in the network by estimating both the resources it will need and the impact it will have on the ongoing connections.

2. AC Algorithms - Types and History

According to whether the traffic parameters are specified a priori or whether the admission decisions rely on measurements of the actual traffic load, Admission Control Algorithms can be classified in two categories: parameter-based and measurement-based. The parameter-based AC algorithms can be analysed by formal methods, while the measurement-based ones can only be analysed through experimentation on real networks or with simulation or emulation.

2.1. Parameter-based AC

These algorithms compute the amount of network resources required to support new flow by given a priori flow characteristics. They can be further classified as non-statistical and statistical allocation algorithms. The **Non-statistical allocation or deterministic** (also called Peak Bandwidth Allocation) is the simplest form among all admission control algorithms. The only knowledge it uses is the peak rate or worst-case parameter to compare against the network's available bandwidth and make a decision on whether to accept the new connection request or not. The algorithm ensures that the sum of requested resources and the existing connections is bounded by the physical link capacity. The most significant disadvantage of this type is that they assume that connections transmit at their peak rates all the time, thus they allocate more bandwidth than it is required to provide QoS guarantees for the existing connections, and the network resources are under-utilised. An other drawback is that there is no multiplexing gain among the sessions admitted into the network, since it works on a flow by flow basis and does not consider sharing bandwidth resources between connections. **Statistical allocation** is a group of much more complicated admission control algorithms. It does not admit new connection requests on the basis of their peak rates; rather the bandwidth for a connection is assigned at less than the peak bandwidth of the connection, depending on the statistical distribution of the arriving cells in the connection [25]. Statistical allocation results in statistical multiplexing gain, when dealing with sources that arrive in "bursts", since it assumes sharing bandwidth resource with other connections and thus the sum of all peak rates may be greater than the capacity of the output link. This type is difficult to be implemented effectively because of the uncertainty in the distribution of the incoming traffic and the inaccurate and difficult-to-calculate statistical information of the traffic arrival process.

Following are the most well-known parameter based algorithms.

Rate or Simple Sum is the simplest parameter-based algorithm, and the most widely implemented by switch and router vendors [24]. It ensures that the sum of requested resources does not exceed the link capacity. The Simple Sum algorithm does not take into account QoS metrics other than bandwidth, and it assumes that every user will use all of its reserved bandwidth.

Acceptance Region schemes decide whether to admit a new flow according to the current state of the system and whether the state lies within the "acceptance" or "rejection" region. The acceptance region is calculated in order to maximize the line utilisation for a nominal packet loss, given a set of flows with a given declaration of peak and mean rates. The calculation in both [20] and [27] assumes that calls arrive according to a Poisson process, that the calls admitted are independent and stay in the network for exponentially distributed time, that they have identical bandwidth requirement statistics and they are described by a Markov fluid process. The model in [20] tries to deal with high offered load situations by not considering for acceptance any call, every time an arriving call is rejected, until after a call currently in progress has ended.

Though the acceptance region algorithms are quite simple, this simplicity comes as a result of simplifications of the network model, which results in limitations in such algorithms. For instance, they perform quite poorly at low link capacity. Another limitation is related to the often made assumption of homogeneous on-off sources. Thus it may not be clear whether such algorithms are still applicable when the traffic sources do not fit this model.

Equivalent Bandwidth (also known as equivalent capacity or efficient bandwidth) is the minimum bandwidth that is needed to carry the traffic that is generated by a source in isolation, without violating the QoS requirements. In this approach, each source is assigned an equivalent bandwidth and a new call is accepted if the sum of these equivalent bandwidths is less than the capacity of the links.

The schemes of [21] and [22] uses a fluid-flow model for the source and a bandwidth allocation process to calculate the equivalent bandwidth by taking into account the impact of source characteristics (the duration of the burst period) either when the impact of individual connection characteristics is critical or when the effect of statistical multiplexing is of significance. The idle periods and burst lengths are assumed exponentially distributed and independent from each other, so that the source statistics can be fully characterised by the peak rate of a connection the mean rate of a connection, and the average duration of a burst period. The equivalent capacity is the minimum value of two parameters. The first relies on a Gaussian approximation for the aggregate bit rate of the network connections routed over a link and is representative of bandwidth requirements when the effect of statistical multiplexing is of significance and the second represents the impact of source characteristics (the duration of the burst period) on the required bandwidth and estimates the equivalent capacity when the impact of individual connection characteristics is critical.

[6] proposes an admission control algorithm, based on [21] scheme, which estimates the equivalent capacity of a class using the Hoeffding Bound, a looser bound of the sum of N independent, random variables. The Hoeffding

Bound does not assume normal distribution of the aggregate traffic, so this model does not assume a normal approximation of the arrival rate distribution, and thus is preferable for classes with either a moderate number of admitted connections or for traffic from heterogeneous sources with wide range in the peak rates.

Equivalent bandwidth schemes are characterised by their simplicity, since determining whether a given set of traffic sources can be accommodated without any QoS violation comes down to comparing the sum of the equivalent bandwidths of individual sources to the link capacity. This approach does not consider the effect of buffering which increases the effective capacity of a system. Of course, since Equivalent Capacity AC algorithms are parameter-based, they reserve the resources that are specified by the source traffic description, which could lead to low network utilisation since users could request more resources than they require.

The **Diffusion based statistical AC** uses statistical bandwidth based on closed-form expressions that use diffusion approximation models. It exploits information about buffer sizes and the multiplexed traffic that shares a common link to obtain a diffusion approximation based cell loss estimate and assumes that traffic follows an "On-Off" behaviour.

The scheme in [17] uses two diffusion models: one for a finite buffer (FB) ATM multiplexer and another for an infinite buffer (IB) ATM multiplexer. The cell loss probability is estimated by the overflow probability, which is the overall probability of exceeding the actual buffer capacity B.

A new connection is admitted if the statistical bandwidth on every intermediate link along the selected path is less than the link capacity. The use of diffusion-based techniques has been shown to be conservative with respect to cell loss, but more economical in bandwidth allocation, and has the disadvantages of the parameter-based algorithms.

2.2. Measurement-based AC (MBAC)

This approach relies on measurements of actual traffic load in making admission decisions. It uses these network measurements to estimate the current load of existing traffic, instead of computing the traffic characteristics out of the user specified connection's parameters. It has no prior knowledge of the traffic characteristics and makes the admission decisions based only on the current state of the network. The measurement-based schemes alleviate the burden on the users to accurately specify the parameters for their traffic flow, and thus is a more practical approach for achieving statistical multiplexing gain with variable-rate traffic.

Following are the most well-known MBAC mechanisms which have been proposed for conventional networks.

Measured Sum [24] is the measurement-based version of the Simple Sum algorithm. It tries to increase the network utilisation by measuring the actual network load and substituting the reserved rates of the existing users with the measured load. Again the admission decision is based only on the availability of bandwidth and does not consider any other QoS metric.

Measurement Based Admission Control with Delay and Bandwidth Constraint schemes do both delay and bandwidth checking and are used with predictive service for "tolerant" applications which allow a certain degree of QoS violations. When a new flow requests service the network must characterise its traffic. The algorithm described in [23], is designed for predictive service which approximates the maximum delay of predictive flows by replacing the worst-case parameters in the analytical models with measured quantities. The computation of worst-case queueing delay is different for guaranteed and predictive services. Such algorithms cannot be used in networks with strict QoS requirements and are only applicable in networks with predictive service. Also the schemes tend to exceed the needed bandwidth reservation, since they use worst-case delays, which is rarely the case since multiple sources will rarely simultaneously transmit packets at peak rate.

The **Endpoint Admission Control** [2, 1, 4, 19, 3] is a measurement-based scheme in which the end host (endpoint) probes the network by sending probe packets at the data rate it would like to reserve and records the resulting level of packet losses, specially marked packets, or other QoS criteria. The host then admits the flow only if the loss or marking percentage is below some threshold value. Endpoint admission control requires no explicit support from the routers that do not need to keep a per-flow state or process reservation requests. This approach simply uses the fact that routers may drop or mark packets in a normal manner. In some cases the probe packets are treated equally with the data packets and in others they are sent at a different priority level.

When a new user requests to enter the network, a probe is sent from the source to the destination. The destination counts the received packets until the probe time period expires and sends a measurement report, under high priority, to the source with the number of probe packets received. Based on that report, if the calculated probe loss probability is less than a threshold the source decides to accept the new call. If he probe does not succeed on reaching the destination, due to temporal network overload or opposing probe processes, the session establishment fails and the source must wait for a random time (back-off time) before issuing a new probe. The back-off time is calculated from a uniform distribution of some width, which is doubled for each consecutively blocked attempt to reach the same receiver.

[7] assumes that call requests arrive according to a Poisson process of rate λ. A new call sends w probe packets

and is admitted if and only if the number of marked probe packets is less than or equal to a threshold r.

End-point admission schemes suffer from the shortcomings of any measurement based scheme where estimates may not be in line with what will be observed when the real traffic is sent instead of the probe traffic.

Measurement-based AC algorithms are shown to achieve much higher utilisation than parameter-based [23], and the more accurate and up-to-date the measurements the better the algorithm. So the last few years the research is focused on accurate network monitoring tools and as admission control is concerned it is turned towards measurement-based approaches.

3. Self Aware Networks

Self Aware Networks (SAN) is a proposal of QoS enabled networks with enhanced monitoring and self improvement capabilities that use adaptive packet routing protocols, such as Cognitive Packet Network (CPN) ([18], [14], [15], [12], [11], [10], [9], [16]) and address QoS by using adaptive techniques based on on-line measurements. In CPN, users declare their QoS requirements (QoS Goals) such as minimum delay, maximum bandwidth, minimum cost, etc. It is designed to perform Self-Improvement by learning from the experience of smart packets, using random neural networks (RNN) [8] with reinforcement learning (RL), and genetic algorithms. RL is carried out using a QoS Goal defined by the user, who generated a request for the connection, or by the network itself. The decisional weights of a RNN are increased or decreased based on the observed success or failure of subsequent smart packets to achieve the Goal. Thus RL will tend to prefer better routing schemes, more reliable access paths to data objects, and better QoS.

More analytically, CPN makes use of three types of packets: smart packets (SP) for discovery, source routed dumb packets (DP) to carry payload, and acknowledgements (ACK) to bring back information that has been discovered by SPs which are used in nodes to train neural networks. Conventional IP packets may tunnel through CPN to seamlessly operate mixed IP and CPN networks. SPs are generated either by a user requesting to create a path to some CPN node, or by a user requesting to discover parts of network state, including location of certain fixed or mobile nodes, power levels at nodes, topology, paths and their QoS metrics.

SPs discover routes by using random neural networks (RNN) [8] with reinforcement learning (RL). RL is carried out using a QoS Goal (such as packet delay, loss, hop count, jitter, etc) which is defined by the user who generated a request for the connection, or by the network itself. The decisional weights of a RNN are increased or decreased based on the observed success or failure of subsequent SPs to achieve the Goal. Thus RL will tend to prefer better routing schemes, more reliable access paths to data objects, and better QoS.

When a Smart Packet arrives to its destination, an ACK is generated and heads back to the source of the request, following the reversed path of the SP. It updates mailboxes (MBs) in the CPN nodes it visits with the information which has discovered, and provides the source node with the successful path to the node. All packets have a life-time constraint based on the number of nodes visited, to avoid overburdening the system with unsuccessful requests or packets which are in effect lost. A node in the CPN acts as a storage area for packets and mailboxes (MBs). It also stores and executes the code used to route smart packets. It has an input buffer for packets arriving from the input links, a set of mailboxes, and a set of output buffers which are associated with output links. The route brought back by an ACK is used as a source route by subsequent DPs of the same QoS class having the same destination, until a newer and/or better route is brought back by another ACK. ACK messages also contain timestamp information that can be used to monitor the QoS metrics on a single link and/or partial or complete paths.

Each node stores a specific RNN for each active source-destination pair, and each QoS goal. The number of neurons in an RNN corresponds to the number of routing outputs of a node. Each output link of a node is represented by a neuron in the RNN. The arrival of Smart Packets(SPs) triggers the execution of RNN and the routing decision is the output link corresponding to the most excited neuron. CPN reinforcement learning changes neuron weights to reward or punish a neuron according to the level of goal satisfaction measured on the corresponding output.

The level of goal satisfaction is expressed by a reward. Given some goal G that a packet has to minimize, the reward R is formulated simply as $R = 1/G$. The state q_i of ith neuron in the network is the probability that it is excited. The q_i, $1 < i < n$ satisfy the following system of nonlinear equations:

$q_i = \frac{\lambda^+(i)}{r(i)+\lambda^-(i)}$

where $\lambda^+(i) = \sum_j q_j w^+_{ji} + \Lambda_i$ and $\lambda_-(i) = \sum_j q_j w^+_{ji} + \lambda_i$

w^+_{ji} is the rate at which neuron j sends "excitation spikes" to neuron i when j is excited, w^-_{ji} is the rate at which neuron j sends "inhibition spikes" to neuron i when j is excited, and $r(i)$ is the total firing rate from the neuron i. For an n neuron network, the network parameters are these n by n "weight matrices" $W^+ = \{w^+(i,j)\}$ and $W^- = \{w^-(i,j)\}$ which need to be "learned" from input data.

The RNN weights are updated based on a threshold T:

$T_k = \alpha T_{k-1} + (1-\alpha) R_k$

where R_k, k = 1, 2, ... are successive measured values of reward R and α is some constant ($0 < \alpha < 1$) that is used to tune the responsiveness of the algorithm: for instance $\alpha = 0.8$ means that on the average five past values of R are being taken into account. Neurons are rewarded or punished based on the difference between R_k, the current reward, and T_{k-1}, the last threshold.

4. Our proposed multiple criterion AC algorithm

The measurement-based AC algorithm we propose [26] is based on measurements of the QoS metrics on each link of the network before and after the transmission of probe packets. This does not require any special monitoring mechanism since the CPN already collects QoS information on all links and paths that the SPs have explored and on all paths that any user is using in the network. Furthermore, since it is the users that determine the QoS metrics that interest them, CPN collects data for the different QoS metrics that are relevant to the users themselves.

The proposed AC scheme consists of two stages. The fist one is called Probing Stage and is the stage where the impact of the new flow is estimated by probing the network. In the second stage (Decision Stage) the AC decides on whether to accept a new call into the network based on whether there is a feasible path which can accommodate the new call without affecting the quality of formerly accepted flows.

4.1. Probing Stage - Estimation of the impact of a new flow

Every QoS metric can be considered as a value which increases as the “traffic load” increases. A new connection will increase the load of the paths it may be using so it is assumed that the value taken by the QoS metrics will increase. For example, delay increases as the network traffic load increases.

Let as consider some link (i, j). A small increase x in the load that is obtained in a controlled manner, e.g. by sending probe packets at rate x, generates an estimate of the manner in which the QoS metric q varies around the current load point Y:

$$\hat{q}' = \frac{q(Y + x) - q(Y)}{x}. \qquad (1)$$

The impact of a new flow with total traffic rate X can then be evaluated by using the estimate and the measured derivative from (1):

$$\hat{q}(Y + X) = q(Y) + \hat{q}'X, \qquad (2)$$

without having to know the initial load Y. This estimate may be optimistic or pessimistic. However it is likely that

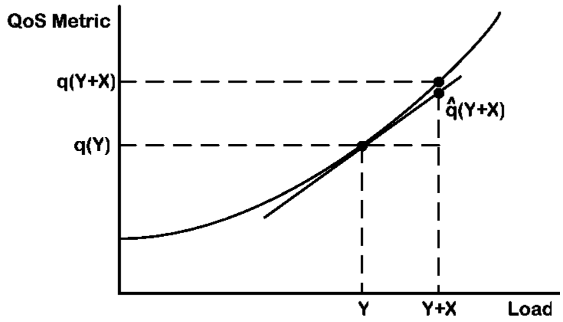

Figure 1. QoS metric vs Load

the path that CPN will select for the probe traffic, because it provides the most favorable impact on current flows and because it satisfies the QoS needs of the new flow. It is also likely that this path is also the best path in terms of actual observed QoS after the new user’s full traffic is inserted. Contrary to the existing measurement-based AC schemes that use probing, in our scheme, it is not required to send the probe packets at the same rate as the new call’s requested rate. Instead we can send them at a much lower rate and still have an accurate estimation. This is a major advantage since this way the probing process has no significant impact on the network’s congestion.

4.2. Decision Stage

Let us assume that the users may be concerned with m distinct QoS metrics $q_v \in R$, $v = 1, ...m$ that are specified in terms of QoS constraints $[q_v \in C_v(u)$ for each user $u]$, where $C_v(u) \subset R$ is typically an interval of acceptable values of the QoS metric v for user u. We will detail the AC algorithm in terms of forwarding packets from some source s to a destination d. However this approach can be generalised to the case where u is requesting some service S.

A network can be considered as a network graph $G(N, E)$ with nodes N, $n = |N|$, and a set E of directional links (i, j), where $i, j \in N$. The CPN algorithm explores $G(N, E)$ and collects QoS data about the parts of the network that are being currently used, or which have been explored by SPs. We assume that this data is available in one or more locations in the form of $n \times n$ *link QoS matrices* Q_v with elements:

- $Q_v(i, j) = r$ where $r \geq 0$ is a real number representing the QoS of link (i, j) which has been measured at some recent enough time, and
- $Q_v(i, j) = unknown$ if i and j are not directly connected or if either a SP has not explored the link for

QoS metric v or if this happened so long ago that the value could be inaccurate.

From the link matrices Q_v we can compute:

- The set of known (explored) paths $P(s,d)$ from s to d, and

- The *path QoS matrices* K_v, where $K_v(s,d)$ is the *known best value* of the QoS metric v for any *path* going from s to d if such a path exists and if the links on the path have known entries in the link QoS matrices. Other entries in K_v are set to the value "unknown".

By "best value" we mean that several paths may exist for the source-destination pair (s,d), but $K_v(s,d)$ will store, for instance, the smallest known delay for all paths going from s to d if q_v is the delay metric. We will discuss below how the path QoS matrices are computed from the link matrices.

4.3. The AC Algorithm

Let us assume that the network is currently carrying z users, any one of which will be generically represented by some QoS constraint $q_w(z)$ and a new user u requests admission for a connection from source s to destination d carrying a traffic rate X and with QoS constraint $q_v(u)$. The proposed AC algorithm proceeds as follows:

- Find the set $P(s,d)$. If it is empty, send SPs to discover paths. If unsuccessful, reject the request. Otherwise monitor the current network, create the $Q_w(i,j)$ matrices for all discovered links and all QoS metrics (including $w=v$), and then send probe traffic at rate x along the network.

- Use the probe traffic to obtain $\hat{q}'_w(i,j)$ for each QoS metric w of interest, including $w=v$, and for all links (i,j). Note that some links may not be concerned by the probe traffic so for that links we take $\hat{q}'_w(i,j)=0$. The path that the probe packets will follow, will be the one that the SPs have chosen as more appropriate so that it satisfy the QoS needs of the new flow, so, it is very likely to also be the path that will be followed after the new user's full traffic is inserted.

- Afterwards compute the estimation

  $$\hat{Q}_w(i,j) = Q_w(i,j) + X\hat{q}'_w(i,j) \quad (3)$$

  for all concerned links and all QoS metrics. For unconcerned links we take $\hat{Q}_w(i,j) = Q_w(i,j)$.

- Compute $\hat{K}_w$ from $\hat{Q}_w$ (to be detailed below) for all the QoS metrics of interest, including v.

- Finally, if $\hat{K}_v(s,d) \in C_v(u)$ AND $\hat{K}_w(s',d') \in C_w(z)$ for all other current users z with source-destination pair (s',d') and QoS metric $q_w \in C_w(z)$, then accept u; else reject the request.

4.4. Computing the QoS matrices

For each $i,j \in N$ of a network graph $G(N,E)$ with nodes N, $n=|N|$, and a set E of directional links (i,j). the well known "Warshall's algorithm" [28] determines whether there is a path from node i to node j by computing the Boolean matrix K, the transitive closure of the graph's adjacency matrix Q, in less than n^3 Boolean operations.

$$K = \bigcup_{k=1}^{n} Q^k \quad (4)$$

or

$$K^n[i,j] = K^{n-1}[i,j] \bigvee \Big(K^{n-1}[i,n] \bigwedge K^{n-1}[n,j] \Big) \quad (5)$$

where $K^1[i,j] = Q[i,j]$ and the matrix elements are treated as boolean values with $\bigvee$ being the logical "OR" and $\bigwedge$ the logical "AND".

Floyd's algorithm [5] extends Warshall's algorithm to obtain the cost of the "smallest cost path" between any pair of vertices in the form of a real-valued matrix.

$$K^n[i,j] = min\Big\{ K^{n-1}[i,j], \Big(K^{n-1}[i,n] + K^{n-1}[n,j] \Big) \Big\} \quad (6)$$

Thus, our algorithm can use Floyd-Warshall's technique to construct K_v from Q_v, and hence $\hat{K}_v$ from $\hat{Q}_v$ if the QoS metric q_v is additive, so that $K_v(i,j)$ is the smallest value of the QoS metric among all known paths from i to j. Note that delay and the variance of delay, are both additive metrics. Although loss rate is not additive, the number of lost packets is an additive metric.

For non-additive metrics we have developed a generalisation of the Floyd-Warshall, which is described next.

4.5. Generalisation of the Floyd-Warshall algorithm to non-additive QoS metrics

Consider the matrix Q_v mentioned above, whose entries are the measured QoS values $r \geq 0$ over links (i,j) whenever such a link exists, or otherwise have the value "unknown".

The matrix K_v which is calculated as shown below, provides us with the "best QoS value" for every path between every pair of vertices (i,j).

$$K_v = \bigoplus_{k=1}^{n} [Q_v]^k \quad (7)$$

or

$$K_v^n[i,j] = K_v^{n-1}[i,j] \otimes \left(K_v^{n-1}[i,n] \oplus K_v^{n-1}[n,j] \right) \quad (8)$$

where in (8) $K_v^1 = Q_v$ and in (7) the operator $\bigoplus$ between two real valued matrices B, C ($D_v = B_v \bigoplus C_v$) is defined as $D_v(i,j) = \otimes_{t=1}^{n} \left[B_v(i,t) \oplus C_v(t,j) \right]$. The operator $\oplus$ between two QoS parameters depends on the QoS metric that is being considered and can be the addition (+) for delay and variance, the minimum (min) for bandwidth etc. The $\otimes$ is also an operator that depends on the specific QoS metric q, and selects the "best value" among the elements on which it operates, e.g. in case of the delay, loss or variance metric it will obtain the minimum value, while for bandwidth or security it will select the maximum value for all paths going from i to j.

5. Experimental results

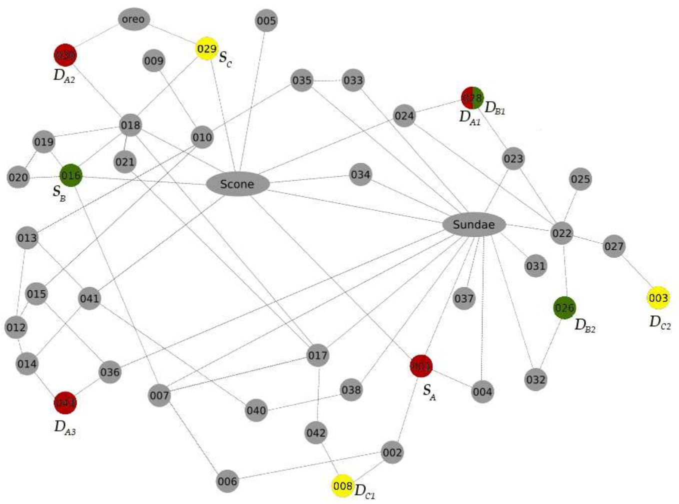

Figure 2. The CPN testbed used in our experiments

Configuration of the experiments:

- The experiments where conducted in a 44-node testbed representing the SwitchLAN network topology[1].
- All links have the same capacity ($10\,Mbits/s$)
- All users have the same QoS requirements: $delay \leq 50\,ms$, $jitter \leq 5\,ms$, $bandwidth \geq 3\,Mbits/s$, and $packetloss \leq 5\,\%$. These values are quite fastidious (for high-quality video the delay should be only less than $150ms$) and are chosen in order to evaluate our algorithm for very demanding requests.
- There are 7 Source-Destination (S-D) pairs ($S_A - D_{A1}, S_A - D_{A2}, S_A - D_{A3}, S_B - D_{B1}, S_B - D_{B2}, S_C - D_{C1}, S_C - D_{C2}$, that correspond to 7 users $(A1, A2, A3, B1, B2, C1, C2)$.
- In order to avoid having more than one users requesting to enter the network at the same time which would lead to multiple flows of probe traffic and misleading measurements, each user enters a queue ("request queue") at the data gathering point.
- The users that are denied access, instead of being considered rejected, they enter another queue ("reject queue") and request to enter again. This process continues until they are finally accepted or a specific period of time ("user lifetime") has past. In our experiments the user lifetime of each user is $150\,s$, meaning that every user will wait to be served for at most $150\,s$ and if that time expires the user will leave the network and will be considered rejected. The "reject queue" has bigger priority than the "request queue" and is served first.
- After making a request, if the request is satisfied then the user will wait for a constant time W and then make another request. The same is true if its request is not satisfied.
- Our experiments covered three cases:
  - The Admission Control is disabled (NO AC).
  - The AC is enabled (WITH AC).
  - The AC is enabled but also the length of the feasible paths is restricted (WITH AC & MPL).

  The third approach tries to take under consideration the fact that the algorithm may accept a very long feasible path which CPN would possibly not use for the new traffic. To avoid this conflict of the two intelligent mechanisms, we set a maximum path length (MPL) limit for the length of the feasible path. In our experiments the limit of the feasible path length was set to 6.
- In total 99 experiments were conducted (33 for each case), lasting $15\,min$ each.
- In all cases each experiment has a time W in the $(150, 120, 90, 80, 70, 60, 50, 40, 30, 20, 15)\,s$ range which correspond to total arrival rate λ for all three users of $(2.8, 3.5, 4.67, 5.25, 6, 7, 8.4, 10.5, 14, 21, 28)\,rqsts/min$ respectively. Thus the load on the system is increased

[1]The Swiss Education & Research Network, http://www.switch.ch/network/

in each of the successive experiments. Each experiment was conducted 3 times and the results presented in this paper are the average value of those three runs.

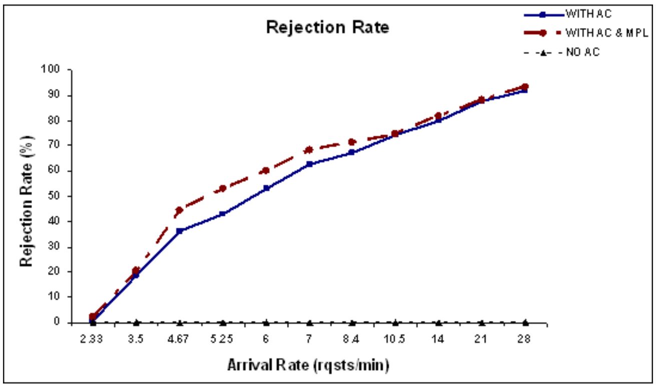

Figure 3. Average rejection rate

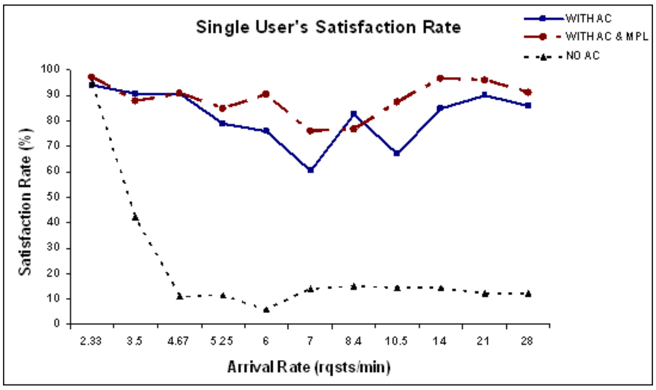

Figure 4. Average user satisfaction

The experimental results are summarize by figures 3, 4, 5. In figure 3 we compare the total rejection rate for the connection requests in all three cases, while figure 4 reports the satisfaction rate of a user (here user $A1$) in all three cases described above. Satisfaction of a user means that all four QoS requirements of that user are fulfilled at all times. Of course when the AC is disabled all users are immediately accepted and so the rejection rate is zero. Figure 5 shows the average time a user has to wait until it is served or the user lifetime expires, when the AC is enabled. In the case where the AC is disabled all users are served the moment it is possible.

We observe that when the AC algorithm is enabled the satisfaction of user $A1$ is much higher than when there is no AC. When the feasible path's length is limited the percentage of the user traffic that is led through the feasible path increases so the results are more accurate and the satisfaction rate is better than when only the AC is enable without

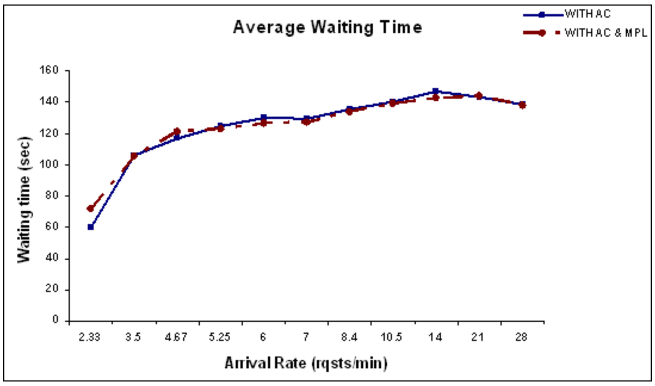

Figure 5. Average time a user waits before he is served

path length restrictions (figure 4). Of course restricting the length of the feasible path which is used to make the decision, makes the algorithm more strict, and thus the rejection rate should increase, as figure 3 confirms. Finding the optimal limit of the feasible paths length is something that could further improve our algorithm and should therefore be further investigated.

6. Conclusions

This paper proposes a measurement based AC algorithm that uses measurements to estimate the impact that a new connection will have on the QoS of both the new and the existing users. The monitoring of the system is being done by exploiting the Self-Aware CPN architecture. We provide a description of the algorithm and some experimental results conducted in a laboratory test-bed showing the effectiveness of the AC by studying the satisfaction throughout a user's lifetime.

A basic difference between our algorithm and other measurement-based AC schemes that use probing is that our algorithm estimates the new flow's impact by probing at a small rate, so that probe packets will not contribute noticeably to the network's congestion. Also the users are the ones that specify the QoS constraints they need in order to obtain the network service they require for a successful connection. The decision of whether to accept a new user is based on a novel algebra of QoS metrics which investigates whether there is a feasible path which can accommodate the new request without affecting the ongoing connections.

Further work following this paper will provide experimental results showing the effectiveness of the algorithm compared to other measurement-based algorithms. Also experiments with different non-additive QoS metrics should be conducted. Finally optimising the rate and duration of the probing would most probably lead to more accurate estimations and further improve our algorithm.

References

[1] G. Bianchi, A. Capone, and C. Petrioli. Throughput analysis of end-to-end measurement-based admission control in ip. In *Proceedings of IEEE INFOCOM 2000*, pages 1461–1470, Tel Aviv, Israel, Mar. 2000. IEEE.

[2] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint admission control: Architectural issues and performance. In *Proceedings of ACM SIGCOMM 2000*, pages 57–70, Stockholm, Sweden, Oct. 2000. Acm.

[3] C. Cetinkaya and E. W. Knightly. Egress admission control. In *Proceedings of IEEE INFOCOM 2000*, volume 1, pages 1471–1480, Tel Aviv, Israel, Mar. 2000. IEEE.

[4] V. Elek, G. Karlsson, and R. Ronngren. Admission control based on end-to-end measurements. In *Proceedings of IEEE INFOCOM 2000*, volume 2, pages 623–630, Tel Aviv, Israel, Mar. 2000.

[5] R. W. Floyd. Algorithm 97: Shortest path. *Communications of ACM*, 5(6):345, June 1962.

[6] S. Floyd. Comments on measurement-based admissions control for controlled-load services. *Submitted to Computer Communication Review*, July 1996.

[7] A. J. Ganesh, P. Key, D. Polis, and R.Srikant. Congestion notification and probing mechanisms for endpoint admission control. *IEEE/ACM Transactions on Networking*, 14(3):568–578, June 2005.

[8] E. Gelenbe. Learning in the recurrent random neural network. *Neural Computation*, 5(1):154–164, Jan. 1993.

[9] E. Gelenbe, M. Gellman, R. Lent, P. Liu, and P. Su. Autonomous smart routing for network qos. In *Proceedings of the First International Conference on Autonomic Computing (ICAC)*, pages 232–239, New York, NY, USA, May 2004.

[10] E. Gelenbe, M. Gellman, and P. Su. Self-awareness and adaptivity for quality of service. In A. Tantawy and K. Inan, editors, *Proceedings of the IEEE International Symposium on Computers and Communications (ISCC'03)*, pages 3–9, Kemer-Antalya, Turkey, June/July 2003. IEEE Computer Society. Invited Paper.

[11] E. Gelenbe, R. Lent, A. Montuori, and Z. Xu. Towards networks with cognitive packets. In *Proceedings of the International Conference on Performance and QoS of Next Generation Networking*, pages 3–17, Nagoya, Japan, Nov. 2000. Opening Invited Paper.

[12] E. Gelenbe, R. Lent, A. Montuori, and Z. Xu. Cognitive packet networks: Qos and performance. In *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS'02)*, pages 3–12, Fort Worth, TX, Oct. 2002. IEEE Computer Society. Opening Keynote Paper.

[13] E. Gelenbe, R. Lent, and A. Nunez. Self-aware networks and qos. *Proceedings of the IEEE*, 92(9):1478–1489, Sep. 2004.

[14] E. Gelenbe, R. Lent, and Z. Xu. Design and performance of cognitive packet networks. *Performance Evaluation*, 46(2-3):155–176, Oct. 2001.

[15] E. Gelenbe, R. Lent, and Z. Xu. Measurement and performance of a cognitive packet network. *Computer Networks*, 37(6):691–701, Dec. 2001.

[16] E. Gelenbe and P. Liu. Qos and routing in the cognitive packet network. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 517–521, Taormina, Italy, June 2005.

[17] E. Gelenbe, X. Mang, and R. Oenvural. Diffusion based statistical call admission control in atm. *Performance Evaluation*, 27/28(Com):411–436, Oct. 1996.

[18] E. Gelenbe, Z. Xu, and E. Seref. Cognitive packet networks. In *Proceedings of the 11th International Conference on Tools with Artificial Intelligence (ICTAI '99)*, pages 47–54, Chicago, IL, Nov. 1999. IEEE Computer Society Press.

[19] R. J. Gibbens and F. Kelly. Distributed connection acceptance control for a connectionless network. In *Proceedings of the 16th International Teletraffic Congress (ITC 99)*, volume 2, pages 941–52, Edinburgh, UK, Feb. 1999.

[20] R. J. Gibbens, F. P. Kelly, and P. B. Key. A decision-theoretic approach to call admission control in atm networks. *IEEE Journal on Selected Areas of Communications*, 13(6):1101–1114, Feb. 1995.

[21] R. Guerin, H. Ahmadi, and M. Naghshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE Journal on Selected Areas in Communications*, 9(7):968–981, Sep. 1991.

[22] R. Guerin and L. Gun. A unified approach to bandwidth allocation and access control in fast packet-switched networks. In *Proceeding of INFOCOM'92*, volume 1, pages 1–12, Florence, Italy, May 1992.

[23] S. Jamin, P. Danzig, S. Shenker, and L. Zhang. A measurement-based admission control algorithm for integrated services packet networks. *IEEE/ACM Transactions on Networking*, 5(1):56–70, Aug./Sep. 1997.

[24] S. Jamin, S. J. Shenker, and P. B. Danzig. Comparison of measurement-based admission control algorithms for controlled-load service. In *Proceedings of the Conference on Computer Communications (IEEE INFOCOM '97)*, volume 3, pages 973–980, Kobe, Japan, Apr. 1997. IEEE Computer Society Press.

[25] H. G. Perros and K. M. Elsayed. Call admission control schemes: A review. *IEEE Communications Magazine*, 34(11):82–91, Nov. 1996.

[26] G. Sakellari, M. D' Arienzo, and E. Gelenbe. Admission control in self aware networks. In *Proceedings of the 49th annual IEEE Global Telecommunications Conference, GLOBECOM 2006*, San Francisco, CA, USA, Nov./Dec. 2006.

[27] D. Tse and M. Grossglauser. Measurement-based call admission control: Analysis and simulation. In *Proceedings of the INFOCOM '97*, volume 3, pages 981–989, Kobe, Japan, Apr. 1997. IEEE Computer Society Press.

[28] S. Warshall. A theorem on boolean matrices. *Journal of the ACM*, 9(1):11–12, Jan. 1962.