

# Self-Awareness and Adaptivity for Quality of Service

Erol Gelenbe, Michael Gellman and Pu Su

School of Electrical Engineering & Computer Science  
University of Central Florida  
Orlando, FL 32816  
{erol,michaelg,psu}@cs.ucf.edu

## Abstract

*Network self-awareness is the ability of a network to observe its own behavior using internal probing and measurement mechanisms, and to make effective autonomous use of these observations for self-management. Experiments are conducted to evaluate the Goal's impact on observed QoS for the user's payload. In addition to packet loss due to congestion, we also introduce an artificial packet loss at certain nodes to represent failures or other undesirable events. We see that just using delay in the QoS goal is a good way to reduce delay and loss if losses are only the result of congestion. However, as one would expect, using loss in the user's QoS Goal is seen to be useful if the paths which are selected by SPs are to avoid nodes where packet losses are occurring for reasons other than congestion. In general we see a good correlation between the QoS Goal that the SPs use to find paths, and the resulting QoS observed by DPs.*

## 1 Introduction

Broadly speaking, Quality of Service (QoS) is the performance of a network as perceived by some specific user or by a class of users [8]. Many authors have developed techniques for estimating QoS from given traffic characteristics (e.g. [3, 6, 7]), while others have studied routing schemes which try to achieve desired QoS objectives [9, 10, 18, 19, 20, 21].

This paper describes an experimental framework that exploits self-awareness and on-line network adaptation to offer user specified and goal based Quality of Service (QoS). Our approach is based on the "Cognitive Packet Network" (CPN) concept, and it describes work we have done on the experimental test-bed [11, 14, 15, 17] that we have designed and implemented. CPN uses smart packets (SPs) and ACK packets to continuously collect and store distributed state information in a packet switching network. Mailboxes store data about the delay and loss on paths in the network,

and Subsequent SP) then use this information to search for better paths in the network based on user specified QoS Goals. Dumb packets (DPs) carry the user's payload packets from source to destination along paths that SPs have discovered. This paper briefly describes the operating principles of CPN, and then develops QoS Goals which combine both packet loss and delay. Experiments are reported on two CPN test-beds in which QoS Goals are used in SP routing and path discovery.

## 2 CPN Routing

CPN includes three different types of packets which play different roles:

- Smart or cognitive packets (SP) are used to discover routes for connections; they are routed using a reinforcement learning (RL) algorithm [2] based on a QoS "Goal". They do not carry payload.
- When a smart packet arrives to its destination, an acknowledgment (ACK) packet is generated and the ACK stores the route followed by the original packet and the measurement data it collected and will return along the "reverse route" of the SP. ACKs deposit information in the mailboxes (MBs)[15] of the nodes they visit.
- Dumb packets carry payload and use source routing. Dumb packets also collect measurements at nodes. The route brought back to a source node by an ACK of a SP is used as a source route by subsequent dumb packets of the same QoS class having the same destination, until a new route is brought back by another ACK.

When an ACK for a packet which was going from  $S$  to  $D$  and was of class  $K$  enters some node  $N$  from node  $M$ , the following operation will be carried out: the difference between the local time-stamp and the time-stamp stored in

the ACK for this particular node is computed and divided by two. The resulting time is stored in the mailbox as the value  $W(K, D, M)$  – it is an estimate of the forward delay for a packet of QoS class  $K$  going from node  $N$  to  $D$  and which exited node  $N$  via the port leading to  $M$ . Note that the identity of the local node  $N$  is obvious and need not be stored. The source node  $S$  is also not relevant since the  $W(K, D, M)$  refers to the time to go from  $N$  to  $D$  using the next node  $M$ . The QoS class  $K$  is needed since the decision at each node, and the resulting observed delay, will depend on the requirements expressed by the QoS class  $K$ . The quantity  $W(K, D, M)$  is inserted in the Goal function (see equation (1) of the RL learning algorithm for the delay value  $W$ ).

Different approaches to learning could in principle be used to discover good routes in a network, including Hebbian learning, back-propagation [1], and reinforcement learning (RL) [2]. Hebbian learning is notoriously slow and was excluded from our consideration. Simulation experiments conducted on a 100 node network showed that RL is the most effective approach, and that it provides significantly better QoS than shortest-path routing [14]. In order to guarantee convergence of the RL algorithm to a single decision (i.e., selecting an output link for a given smart packet), CPN uses the random neural network (RNN) [4, 12] which has an unique solution to its internal state for any set of “weights” and input variables. CPN’s RL algorithm uses the observed outcome of a decision to “reward” or “punish” the routing algorithm, so that its future decisions are more likely to meet the desired QoS Goal. The “Goal” is the metric which characterizes the success of the outcome, such as packet delay, loss, jitter and so on. As an example, the QoS Goal  $G$  that SPs pursue may be formulated as minimizing Transit Delay  $W$ , or Loss Probability  $L$ , Jitter, or some weighted combination captured in the numerical Goal function  $G$  and the reward  $R = 1/G$ . Successive values of  $R$ , denoted by  $R_l$ ,  $l = 1, 2, \dots$ , are computed from the measured delay and loss data (see Section 3), and are used to update the neural network weights. Please refer [15, 17] to get more details about CPN RL algorithm.

### 3 Constructing Composite QoS Goals

For an application which has QoS needs that include both loss and delay, the QoS goal that may be used to route packets will include both the loss and delay incurred from source to destination. In this case, we can form the goal function  $G$  as follows:

$$G = (1 - \overline{L}_f)\overline{W} + \overline{L}_f(T + G) \quad (1)$$

where  $\overline{L}_f$  is an estimate of the forward packet loss ratio, and  $T$  is the additional time incurred by a packet which is

retransmitted after a loss, including the time-out delay before a non-acknowledged (and presumably lost) packet is retransmitted, and any additional overhead resulting from the retransmission of the lost packet. The expression (1) is based on the idea that if a loss occurs, with probability  $\overline{L}_f$ , then the resulting cost is the delay  $T$  until the packet is retransmitted, and this will be followed by the same equivalent total delay  $G$  incurred by the freshly retransmitted packet. If on the other hand a packet is not lost with probability  $[1 - \overline{L}_f]$ , then the cost is simply the delay  $\overline{W}$  that will be incurred by a packet as it traverses the network to reach its destination. Note that  $G$  appearing on both sides of (1) is written under the assumption that the subsequent packet sent out to replace the lost packet will on the average incur the same total cost  $G$ , since it too may be lost and could be retransmitted. This expression simplifies to yield the reward  $R = 1/G$ :

$$R = \frac{1}{T \frac{\overline{L}_f}{1 - \overline{L}_f} + \overline{W}} \quad (2)$$

In order to use  $R$  we must obviously be able to estimate  $\overline{W}$  and  $\overline{L}_f$ . In Section 2 we describe how ACK packets deposit an estimate of “delay to the destination” into the MBs of nodes that they visit. In order to select a particular path in the network based on composite path QoS metrics, CPN also needs to estimate path packet loss ratios defined as the number of packets sent but not received, divided by the number of packets which have been sent. We will discuss how to estimate link loss and path loss in the following.

#### 3.1 Estimating Link Loss and Path Loss

Packet loss ratios are simply the ratio of number of packets correctly received to the number of packets sent. The link loss ratio refers to the corresponding quantity measured over a single link connecting two nodes. Path loss ratio on the other hand refers to the quantity measured over a path, from a source to a destination. We will use the terms “cumulative” or “path” loss interchangeably. In CPN, we estimate the link loss ratio but forwarding, over the link and back to the predecessor node, the number of packets that have been received by the next node on the link. This information is in fact stored or “piggy-backed” in ACK packets. If  $N$  packets have been sent over a link and  $R$  packets have been received at the next node, then the loss ratio is:

$$L = 1 - \frac{R}{N} \quad (3)$$

To estimate the path loss ratio, we use ACKs coming back from the destination node. The source is able to estimate the round-trip loss ratio by keeping track of the number of packets sent and the number of ACKs it receives. However,

in addition the destination can keep track of the number of packets which are received at the destination, and this number can be piggy-backed inside ACK packets and returned to the source. The time stamp at the destination which is carried by the ACK, will allow the sender to estimate forward loss rates over a given period of time. Thus even if some ACK packets are lost, it is still possible to have a fairly accurate estimate at the source of forward (source to destination) path losses, and not of just round-trip losses. However, the loss ratio estimates at the source can be insensitive to *short term* changes which are important to QoS driven adaptive routing. We address this problem in CPN by using the following scheme:

- The sender maintains a smoothed average of the packet loss ratio:  $\bar{L} \leftarrow (1 - a)\bar{L} + aL$ .
- The receiver modifies  $R$  as follows for some threshold value of  $R_{max}$ :

$$\text{if } R > R_{max} \quad \text{then } R \leftarrow 0$$

- If  $R_i$  is  $l$ -th value of  $R$  received at the sender, the sender carries out the following operation:

$$\text{if } R_{i+1} < R_i \quad \text{then } N \leftarrow R_{i+1}.$$

As a result, large values of  $R$  are eliminated, while  $\bar{L}$  preserves an accurate estimate of the loss ratio over the link from the sender's perspective.

### 3.2 Simplifying the Cumulative Loss

Since it is impractical to have the destination nodes keep a count of the number of packets received for each possible route from every possible source, we need to find a scheme that will reduce the amount of data that is stored. This requires us to make a simplifying assumption based on the idea that forward and reverse routes generally use the same set of nodes and links. Thus, we assume that the DP loss ratio  $L_f$  from the source  $S$  to the destination  $D$  is proportional to the ACK loss ratio  $L_b$  in the opposite direction or  $L_f = \alpha L_b$ . Let  $N$  be the number of DPs sent from  $S$  to  $D$ , and  $A$  be the corresponding number of ACKs received by  $S$ . We can write:

$$\frac{A}{N} = 1 - L_f \cdot L_b = 1 - \alpha(L_f)^2, \quad (4)$$

so that

$$L_f = 1 - \sqrt{\frac{A}{\alpha N}}. \quad (5)$$

The source  $S$  therefore stores separate  $(N, A)$  values for each of its destinations. Assuming that forward and reverse loss rates are identical, we set  $\alpha = 1$  and  $L_f = 1 - \sqrt{\frac{A}{N}}$ .

If routing only selects paths which offer the lowest packet loss, there are several ways in which we can construct the reward  $R$ . One approach is to set  $\bar{W} = 0$  in the expression (2), obtaining:

$$R = \frac{1 - \bar{L}_f}{T\bar{L}_f}, \quad (6)$$

so that  $1/T$  acts as a constant multiplier. In practice, since we do not want  $R$  to be infinite when  $L_f = 0$ , we set:

$$R = \frac{1 - \bar{L}_f}{T(\bar{L}_f + \epsilon)}, \quad (7)$$

where  $\epsilon$  is a constant representing some minimal value for the loss. A simpler approach is to use  $R$  of the form:

$$R = \frac{\beta}{\bar{L}_f + \epsilon}, \quad (8)$$

which relates loss directly to the reward. This is the approach we have taken in our experiments when we just deal with loss (rather than loss and delay). In the experiments we report, we have used the following numerical values of the constants:  $\epsilon = 10^{-5}$  and  $\beta = 0.5$ .

## 4 Measurement Results

The measurements that we report in this section were performed under a variety of conditions on the two network test-beds of Figures 1 and 5. All tests were conducted using a flow of UDP packets entering the CPN network with constant bit rate (CBR) traffic and a packet size of 1024 Kb. All CPN links used 10Mbps point-to-point Ethernet. In wired networks, loss is typically correlated with congestion while congestion can be observed via packet forwarding delays which grow with congestion, so that using delay as a QoS Goal can serve to minimize loss as well as delay. However, in an environment where loss may be unrelated to congestion, such as in a wireless setting, minimizing delay will not necessarily minimize loss. To study the effect of packet loss when it is not just a consequence of congestion, an ‘‘artificial loss ratio’’ was introduced in one of the nodes (Node 1) in the network of Figure 1, with the possibility of varying the artificial loss rate.

We first conducted experiments with ‘‘zero artificial loss rate’’. In Figure 2, we can see that both delay and loss performance based on only using delay as the QoS goal is better than the performance based on using loss *and* delay, either at the link level or over whole paths (cumulative). Thus routing based on delay appears to minimize *both* observed loss and delay. We were surprised to observe that the end-to-end QoS for routing based on link Loss was smaller than that for the routing using cumulative Loss.

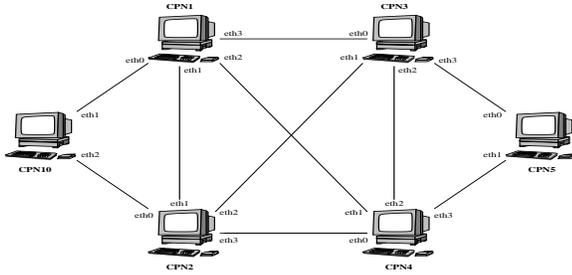


Figure 1. Small CPN test-bed.

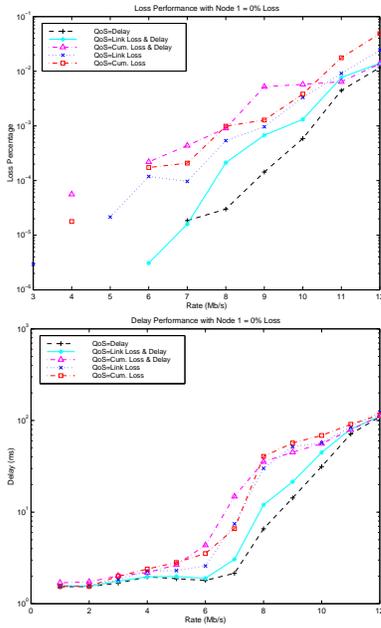


Figure 2. Loss (top) and delay (bottom) in the small test-bed with 0% artificial loss rate at Node 1.

When we introduce a very small (1%) artificial packet loss rate (see Figure 3) delay does not seem to change, but the loss performance curves that use cumulative loss as the goal function are clearly the best as long as the network is not heavily loaded (i.e., link traffic under 8Mbps). When traffic is close to saturation levels, losses due to congestion dominate and (as one would expect) the results become similar to those of Figure 2. Finally, in Figure 4 we increase the artificial packet loss ratio to 5 and observe once again that using cumulative loss as the goal in the routing algorithm provides the best overall results.

To provide further evaluation of composite goal functions, we conducted another set of measurements on the larger CPN testbed consisting of 26 nodes shown in Figure 5. The same UDP packet stream with CBR, as before, were sent into the larger CPN testbed into Node 10 as the

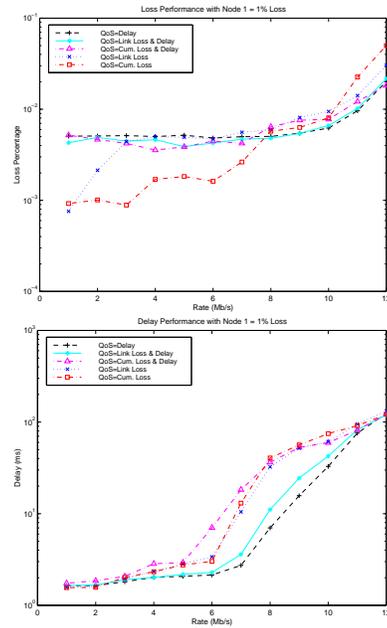


Figure 3. Loss (top) and delay (bottom) in the small test-bed with 1% artificial loss rate at Node 1.

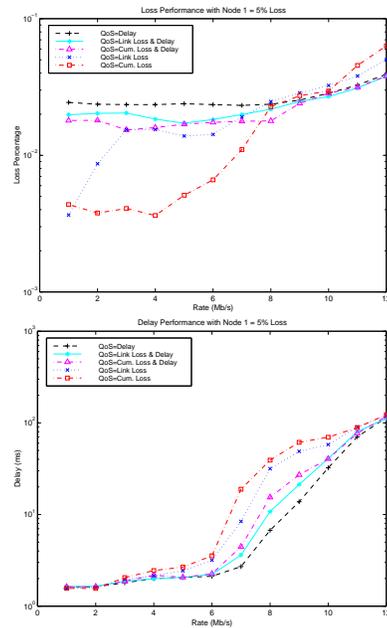


Figure 4. Loss (top) and delay (bottom) in the small test-bed with 5% artificial loss rate at Node 1.

source, for forwarding to Node 7 as the destination. The CPN protocol was run in the usual manner with paths being discovered by SPs, while DPs were used to carry and source route the payload UDP packets. No artificial losses were introduced. As shown in the top figure of Figure 6,

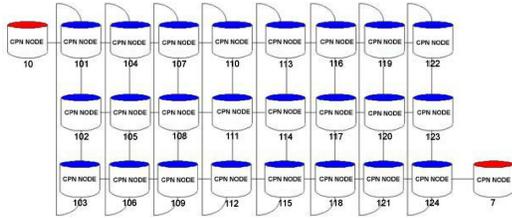


Figure 5. Larger CPN testbed.

the reduction in packet loss rate due to the use of cumulative loss alone, appears even more significantly in the larger test-bed. Here, using the goal function based only on cumulative loss results in the lowest observed loss rates. The curves in the bottom figure show that the lowest delay is obtained by using only delay in the goal function. The linear scale used for delay (y-axis) in this figure clearly shows a peak for traffic in the range of 11 to 12Mb/s, with a reduction in delay above that value, when cumulative loss or cumulative loss and delay are used in the RL routing goal function. The drop in delay at higher traffic values is presumably due to the significant loss of packets which results in lower congestion. Figure 7 summarizes measurements on the larger test-bed, with packet loss only resulting from congestion (i.e. no artificial packet loss) from the perspective of QoS perceived by the user. The purpose of these figures is to see whether the end user is indeed obtaining the outcome in QoS that it is requesting. On the y-axis of the three figures we show the Reward function value at the end user (and not the numerical value that is used by the SPs, which operate at the lower network level). The Reward is computed using the formulas given in equations (2) and (8), using the *measured* values of loss and delay. In the upper curves, we show the ‘loss based’ Reward for the user, as a function of user traffic. We see very clearly that the highest (therefore the best) Reward is obtained if we use *only Loss* in the Goal function, while using Delay and Loss and Delay are worse but equivalent from the user’s perspective. In the middle and lower figures, we see that using Delay as the QoS Goal will provide the user with the best Reward at low traffic; all three Goal functions provide equivalent QoS at higher traffic values. The two lower figures are nearly identical. The reason is that when we combine loss and delay to compute the reward function, the effect of the loss rate for (1) low loss values (i.e. less than 10%) is negligible in the reward function, while for (2) high loss values all of the congestion avoidance methods embodied in the three Goal functions provide equivalent performance and QoS. How-

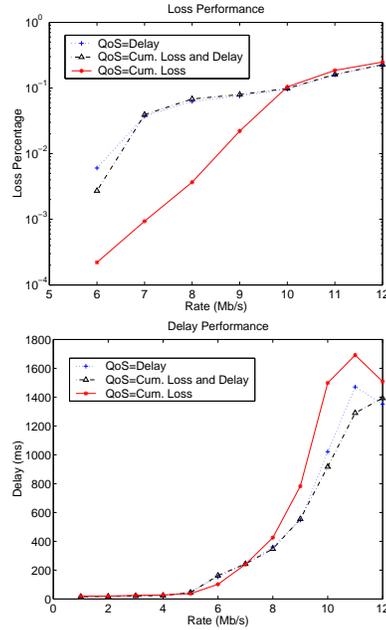
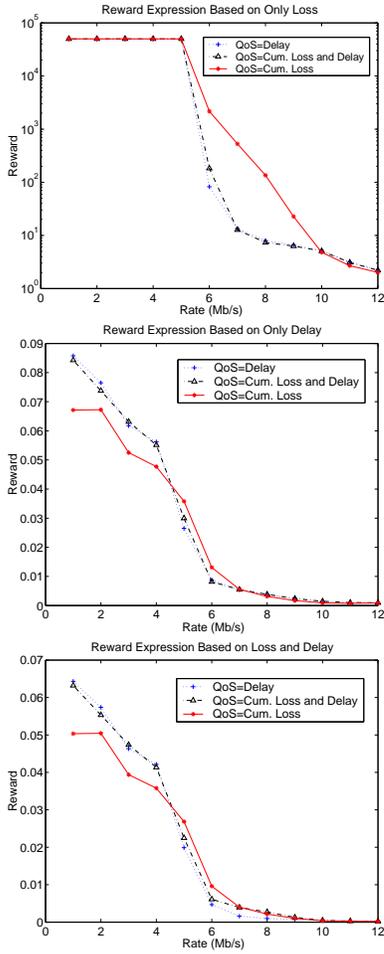


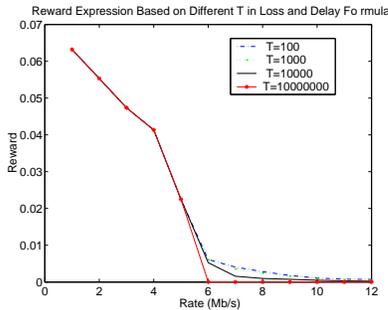
Figure 6. Loss (top) and delay (bottom) in the large test-bed with 0% artificial loss rate.

ever in view of the upper set of curves, we do see that using a Goal function which is specifically tuned to the user’s needs is justified since in some cases it will have a definite effect, while in other cases it will not be detrimental to user perceived QoS. One more set of measurements on the larger test-bed are reported in Figure 8. The purpose is to evaluate the impact of the parameter  $T$  used in the Goal functions. The curve shows that varying  $T$  between  $10^2$  and  $10^5$  has little impact on the values of the reward value perceived by the user; a more significant difference is only perceived at high traffic rates (hence high loss rates) when  $T = 10^7$ .

In the measurements on the small test-bed, we have observed that when packet loss is only due to congestion, the use of delay by itself in the routing goal function results *both* in the smallest delay and the smallest loss. This is a strong indication that measuring delay is a good sufficient indicator for congestion in the small test-bed. However, when artificial losses are introduced in a node, using loss alone results in the smallest loss. However, when the network is saturated, we see that all goal functions (with or without both metrics) essentially result in the same delay and loss. Also we observe using link loss or cumulative path loss in the goal function does not seem to have a significant difference under low to moderate load conditions, but cumulative loss is generally the better choice. In the larger test-bed, we observe when the network is operating close to congestion, the use of delay in the goal function results in better delay characteristics for DPs. Similarly, us-



**Figure 7. Measured Reward value at user level when RL is based on Loss (upper) and Delay (middle), and combined Loss and Delay (lower); large test-bed with 0% artificial loss.**



**Figure 8. Observed Reward for different values of T.**

ing loss alone in the goal function provides the smallest loss rate for DPs. Thus our experiments support the claim that there is a good correspondence between the QoS Goal that the SPs use to find paths, and the resulting QoS observed by the users' payload.

## 5 Future Work

In future work we will extend our approach to more general multi-dimensional QoS Goals, such as the need to maintain the QoS of a connection within a defined boundary defined by values of several different metrics such as loss, delay and jitter. We will include power limitations, which are relevant to wireless networks [16] in the QoS Goal. We will also consider the use of priority schemes for packets within CPN routers. These extensions will provide a broader framework within which network self-monitoring and self-awareness can be exploited dynamically to offer best-effort QoS to users.

## References

- [1] D.E. Rumelhart and J.L. McClelland, *Parallel distributed processing Vols. I and II*, Bradford Books and MIT Press, 1986.
- [2] R. S. Sutton, "Learning to predict the methods of temporal difference", *Machine Learning*, Vol.3, 9–44, 1988.
- [3] H. Ahmadi, R. Guerin and M. Nagshineh "Equivalent capacity and its application to bandwidth allocation in high speed networks", *IEEE J. on Sel. Areas in Comm.*, Vol. 9, 968–981, 1991.
- [4] E. Gelenbe "Learning in the recurrent random neural network", *Neural Computation*, Vol. 5(1), 154–164, 1993.
- [5] H. Zhang "Service disciplines for guaranteed performance service in packet switching networks", *Proc. IEEE*, Vol. 83, No. 10, 1374 – 1396, Oct. 1995.
- [6] E. Gelenbe, X. Mang and Y. Feng "Diffusion cell loss estimates for ATM with multiclass bursty traffic", *Computer Systems–Science and Engineering*, Special Issue: ATM Networks, Vol. 11, No. 6, pp. 325–334, November 1996.
- [7] V. Srinivasan, A. Ghanwani, E. Gelenbe "Block cell loss reduction in ATM systems", *Computer Communications*, Vol. 19, pp. 1077-1091, 1996.
- [8] D.D. Clark and W. Fang "Adding service discrimination to the Internet", *IEEE/ACM Transactions on Networking*, Vol. 6, No. 4, pp. 362-373., August 1998.

- [9] S. Chen and K. Nahrstedt, "An overview of Quality-of-Service routing for the next generation high-speed networks: Problems and Solutions", *Network Magazine*, Nov/Dec. 1998.
- [10] S. Chen, and K. Nahrstedt, "Distributed Quality-of-Service routing in ad-hoc networks", *IEEE J. on Sel. Areas in Comm.*, Vol. 17, No. 8, 1–19, 1999.
- [11] E. Gelenbe, E. Seref, Z. Xu "Towards networks with intelligent packets", *Proc. IEEE-ICTAI Conference on Tools for Artificial Intelligence*, Chicago, November 9-11, 1999.
- [12] E. Gelenbe, Z.-H. Mao, and Y. Da-Li. "Function approximation with spiked random networks", *IEEE Transactions on Neural Networks*, Vol. 10 (1), 3–9, 1999.
- [13] L. Zheng and L. Zhang. "Modeling and performance analysis for IP traffic with multi-class QoS in VPN", *Proc. MILCOM 2000. 21st Century Military Communications Conference*, Vol. 1, pp. 330 – 334, 2000.
- [14] E. Gelenbe, R. Lent, Z. Xu "Towards networks with cognitive packets," *Proc. IEEE MASCOTS Conference*, ISBN 0-7695-0728-X, pp. 3-12, San Francisco, CA, Aug. 29-Sep. 1, 2000.
- [15] E. Gelenbe, R. Lent and Z. Xu "Measurement and performance of Cognitive Packet Networks", *J. Comp. Networks*, Vol. 37, 691–701, 2001.
- [16] E. Gelenbe and R. Lent "Mobile Ad-Hoc Cognitive Packet Networks", *Proc. IEEE ASWN*, Paris, July 2-4, 2002.
- [17] E. Gelenbe, R. Lent, A. Montuori and Z. Xu "Cognitive packet networks: QoS and performance", Keynote Paper, to appear in *Proc. IEEE MASCOTS Conference*, San Antonio, TX, Oct. 14-16, 2002.
- [18] F. Hao, E.W. Zegura and M.H. Ammar "QoS routing for anycast communications: motivation and an architecture for Diffserv networks", *IEEE Comm. Magazine*, Vol. 46, No. 2, 48 – 56, June 2002.
- [19] Ying-Dar Lin, Nai-Bin Hsu and Ren-Hung Hwang "QoS routing granularity in MPLS networks", *IEEE Comm. Magazine*, Vol. 46, No. 2, 58 – 65, June 2002.
- [20] S. Nelakuditi and Zhi-Li Zhang "A localized adaptive proportioning approach to QoS routing granularity", *IEEE Comm. Magazine*, Vol. 46, No. 2, 66 – 71, June 2002.
- [21] M. Kodialam and T.V. Lakshman "Restorable quality of service routing", *IEEE Comm. Magazine*, Vol. 46, No. 2, 72 – 81, June 2002.