Internet Engineering Task Force
INTERNET-DRAFT
draft-shivkuma-ecn-diffserv-01.txt

S. Kalyanaraman/ D. Harrison/
S. Arora/ K. Wanglee/ G. Guarriello
Rensselaer Polytechnic Institute

March, 1998
Expires: September 1998

A One-bit Feedback Enhanced Differentiated Services Architecture

Status of this Memo

Abstract

This document proposes the use of one bit in the DS-byte to facilitate ECN-type control in the differentiated services architecture. Specifically during periods of congestion along a flow's path (indicated using the one-bit mechanism), the out-of-profile packets of the flow (packets for which the "In profile" bit is cleared) can be either delayed or dropped by the ingress network edge routers. This scheme would allow interior routers to preserve their resources for processing in-profile packets during congestion and guard against certain types of denial-of-service attacks. The proposed mechanism could also be used to build differentiated services networks with lower average delay, and aid the implementation of congestion-based pricing schemes in such architectures. The mechanism interoperates well with the baseline differentiated services architecture and can also interoperate with ECN-TCP and future ECN-enabled transports/applications. Further, the ECN-type flow control can be deployed within a differentiated services network even if end-to-end ECN support is unavailable, allowing a quick migration path.

1. Introduction

The DS-byte (formerly TOS octet [RFC791],[RFC1349]) in the IP header is being defined to allow a packet to receive differential treatment depending upon the bits set [Nichols]. The DS-byte is marked by traffic conditioners at the network boundaries. A DS-byte classifier and per-hop behaviors based on those classifications are required in all network nodes of a differentiated-services-capable network. The differentiated services architecture effort is aimed at specifying the number and type of the building blocks and services built from them [Nichols].

The operational model [Nichols] is based upon two sets of proposals. One set of proposals [Ellesson, Ferguson, Heinanen, SIMA] suggests various encodings of the DS-byte to get specific per-hop behaviors from nodes such as low delay, drop preference, priority queueing etc. The other set [Clark, 2BIT] suggests behaviors required from the network as a whole. The DS-byte definition currently allows one bit based upon the latter set, which is used to mark a packet as IN or OUT of its negotiated service profile (we use the term "in-profile" or "out-of-profile" to qualify such packets). Five bits (the per-hop behavior or PHB field) have been kept for defining per-hop behaviors. The last two bits are currently unused (CU) have been reserved for Explicit Congestion Notification (ECN) experimentation.

The purpose of this draft is to highlight some features of ECN as it relates to differentiated service, and propose changes in the DS-byte and ICMP message type allocations to help support these features. Specifically, we believe new levels of service differentiation can be created using *any* existing differentiated service class combined with an "edge-to-edge" flow control scheme. The term "edge-to-edge" refers to a proposed control loop between the ingress and egress network edge routers in the differentiated services architecture. Though ECN was originally defined for controling TCP traffic [Floyd-ECN], our proposed edge-to-edge ECN works at the IP layer and can control both TCP and non-TCP, and both unicast and multicast traffic. We propose a new "INTERNAL" ICMP message type to carry notifications from the egress edge to the ingress edge (see section 5), which eliminates the need for any reverse traffic or acknowledgement flow for a session. Within the context of the differentiated services working group charter, we propose the use of one bit which can be used by such "edge-to-edge" flow control schemes. The bits currently reserved for ECN in the DS-byte could be overloaded with this function or a new bit can be used within the PHB for this purpose.

We first review ECN concepts in section 2 and discuss problems/issues in a non-ECN differentiated services architecture in section 3. We then describe the role of ECN (specifically the capability to do

edge-to-edge flow control) in the differented services architecture in section 4. This discussion includes the description of a sample ECN-enhanced architecture and a set of strategies for ECN generation at routers and response by edge routers. The standardization issues including the possible use of PHB or CU field for edge-to-edge ECN is discussed in section 5. Section 6 outlines ECN issues with IPSEC tunneling and a discussion of its vulnerability to the denial-of-service attack problem. Section 7 discusses compatibility issues with older TOS semantics and section 8 gives a summary.

2. Explicit Congestion Notification (ECN) mechanisms

The concept of ECN mechanisms for TCP congestion control was developed by Sally Floyd [Floyd-ECN] and there is a recent proposal to reserve two bits for ECN in the IPv6 [IPv6] Class octet [Ramakrishnan]. To allow experimentation in IPv4, two bits (ECN-enabled bit, and ECN bit) in the DS-byte have also been set aside for ECN. As the name suggests ECN mechanisms are used by the network to unambigously declare a state of congestion. ECN-enabled transport protocols (like ECN-TCP) would respond to the notification through a reduction of load. Specifically in the case of TCP it has been suggested [Floyd-ECN] that the ECN response be similar to the response to a detection of packet loss (i.e. halving the source congestion window). There has also been a drive to urge all transport protocols to perform some kind of flow control for the collective benifit they can derive from the network, even though they may or may not care for error control [Floyd-RouterMech]. Since routers can utilize ECN bits to give a transport-neutral congestion indication, ECN is expected to play a significant role in the development of flow control methods in non-TCP transport protocols. As a result, more transport or application protocols could be expected to become "ECN-enabled" in the future.

In the context of TCP, ECN provides a clear, if modest benifit [Floyd-ECN]. When a router is entering a congestion phase it could use ECN instead of packet discard to notify the sources about congestion. ECN-enabled routers could set the ECN bit if the ECN-enabled bit is set on a packet instead of dropping the packet. The destination then sets the ECN bit in the acknowledgement to notify the source. The source responds by cutting its window size. Avoiding unnecessary packet drops is particularly attractive to low bandwidth, interactive applications [Floyd-ECN]. Bulk-data transfer applications using ECN-TCP get high throughput over a wide range of conditions and parameter values (TCP clock granularity, TCP window size, RED gateway variation etc).

The authors of this document were inspired by two features of ECN: the transport-neutral feedback property which would be used to

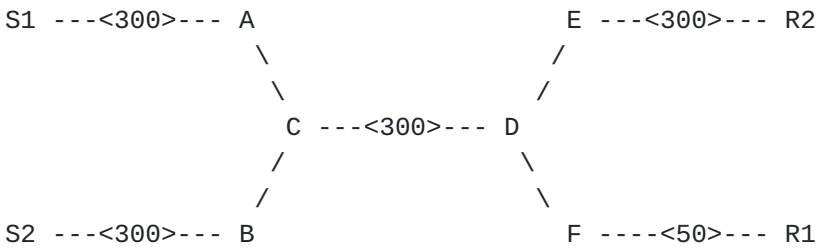
provide service differentiation at the IP-layer, and the potential of ECN to provide resistance to certain kinds of denial-of-service attacks and control the average delay inside the differentiated-services-network.

3. Issues in a non-ECN differentiated services network:

Consider a simple differentiated service network which provides a packet-drop priority for packets which are marked in-profile (i.e, out-of-profile packets are dropped before in-profile packets during congestion). Deering [Deering] argued that this situation may create a positive incentive for sources to overdrive the network by sending an unlimited amount of out-of-profile packets. An example is a source using layered encodings, but no congestion backoff. Misbehaving TCP sources (hacked or otherwise) also belong to this category.

When a node in the path is congested, out-of-profile packets are dropped first before dropping in-profile packets. The dropping of an out-of-profile packet inside the differentiated services network can be construed as a waste of shared resources consumed by the packet to reach the node where it was dropped. These wasted resources could have been utilized to service other in-profile packets or even other out-of-profile packets which could have made it to their destinations - a type of denial-of-service attack. Deering [Deering] showed an example where this can happen and we reproduce it here (with some editing to motivate ECN):

In the following diagram, S1 and S2 are sources of traffic sending to receivers R1 and R2, respectively. A through F are routers, and the numbers in angle brackets are the capacities of the links in Kbps.



Assume that each of the two sources is sending a 50 Kbps voice stream and a 200 Kbps video stream to its corresponding receiver, as part of a teleconferencing application. And assume that that's the only traffic in this simple network.

Further assume that both S1 and S2 are marking (by use of a 1-bit or multi- bit drop-preference field) all their video packets as "out-of-profile" or "less important" than their voice packets, so that if congestion is encountered, the video packets will be discarded before

the audio packets, to avoid audio degradation if at all possible.

Assuming the less important packets are dropped by the routers in favor of more important packets, and that packets of the same importance are dropped with equal probability, C will discard half of S1's video packets and half of S2's video packets, and F will discard the rest of S1's video packets. R1 will receive only S1's voice packets. R2 will receive S2's voice packets plus half of S2's video packets. Thus, S1, by sending video (or "out-of-profile") packets that cannot be delivered to R1, has denied half of S2's video packets access to the C-D link. (Further, if R2 is unable to produce an acceptible video image from only half of the video packets, the bandwidth consumed by those packets is also wasted, which could, in turn, deny bandwidth to other traffic flows in a more general example.) If S1 were instead to refrain from sending its undeliverable video packets, or were informed via feedback that the video packets were in fact undeliverable, R2 could receive the complete (or a considerably better) video stream from S2.

The key point is that upstream *shared* resources were wasted by out-of-profile packets which would never reach their destination. The need is therefore to develop network mechanisms, which can be used as building blocks to provide positive incentives for applications to adapt (keep all its traffic in-profile) and disincentives for those that do not adapt. We look at one such mechanism, RED-plus-penalty-box next and discuss its limitations to motivate the need for edge-to-edge ECN.

3.1 RED-plus-penalty-box:

One way to set up the system of incentives for end-to-end flow control is the "RED-plus-penalty-box" [RED] approach where a simple accounting technique in addition to Random Early Detection (RED) algorithm can help identify misbehaving connections, which would then be treated differently. The utility of this method is constrained in the differentiated services framework because in-profile packets are expected to be serviced uniformly for both well-behaving and misbehaving flows. Dropping more out-of-profile packets of the misbehaving flow than the others, while certainly a benifit, does not prevent the waste of resources upstream by this flow ("upstream" and "downstream" are defined in terms of the flows and the router where the packet is dropped).

"RED-plus-penalty-box" can still be used to avoid wastage of upstream resources - by requiring that all nodes in the differentiated services network implement the "penalty-box" scheme and identify misbehaving flows even if they are not congested. Then out-of-

profile packets belonging to the misbehaving flow (detected close to the entrance of the network) could be dropped at a higher rate even though there is no local congestion at the node. There are two problems with this scheme: first, when there is no congestion downstream, there is no need to identify the misbehaving flow and drop its out-of-profile packets. Second, it requires *all* nodes to implement the penalty-box scheme.

4. Role of ECN in the Differentiated Services Architecture:

We show in this section that schemes based upon ECN can be used to identify misbehaving sources which concentrate all "penalty" actions in the traffic conditioners at the edge of the network. Therefore all nodes need not implement the penalty actions. Moreover this behavior is possible while interoperating with ECN-enabled transports - the edges could choose not to impose edge-to-edge flow control if end-to-end flow control is enabled (subject to monitoring and policing).

More generally, the traffic conditioners at the network edges could "proxy" as ECN-capable transports. These schemes would have the desirable property that out-of-profile packets are admitted into the differentiated services network when the paths traversed by the flows which they belong to are not congested, but kept out of the network when the paths are congested, providing a strong incentive for applications to adapt to remain within profile. An advantage in terms of pricing is that, since ECN mechanisms indicate congestion unambigously, the network edges could rely on them to implement congestion-pricing schemes. Another potential advantage is to achieve lower-average-delay for in-profile packets, which could translate to throughput for bulk-data transfer (especially for high bandwidth transfers [Stevens, Sec 24.3, pp 346-347]) and enhanced interactive quality for delay-sensitive interactive applications [Floyd-ECN].

Next we study the benifits from adding the ECN functionality to the simple differentiated services network considered in section 3. In what follows, we refer to the the differentiated services network as simply "the network", where a collection of "interior routers" lie within a boundary defined by "edge routers" (ingress and egress). The "edge routers" implement traffic conditioning functions. The term "profiler" [Clark] is used to denote the component in the edge router which decides whether a packet is in/out of profile and marks/drops the out-of-profile packets. The profiler corresponds to the "policer" or "shaper" in the differentiated services baseline document [Nichols]. The "interior routers" detect congestion using techniques based upon RED [RED] or RIO [Clark].

The combination of marking/dropping of out-of-profile packets by the

profiler is currently implementation specific. One extreme is to mark out-of-profile packets and not drop them (unless they overwhelm local processing/queueing facilities). This behavior would run into the problems described in section 3. Another extreme is to drop all out-of-profile packets, which is again not a good idea if there is no congestion along the path [Qmgmt]. An intermediate behavior can be achieved using ECN.

4.1. A Sample ECN-enhanced Differentiated Services Architecture:

A sample architecture for edge-to-edge flow control based upon ECN is as follows. Profilers at network edges mark out-of-profile packets by default, but switch over to dropping them during phases of congestion in the path (indicated by the network using ECN). A ECN-capable interior router upon detecting congestion drops out-of-profile packets and sets ECN on in-profile packets of the same flows whose out-of-profile packets were dropped. The ECN-capable egress router detects the ECN indication and sends a "notification" packet (which could be defined to be new "INTERNAL" ICMP type, see section 5) to the source of the flow for which ECN was set. The egress router also clears the ECN bit on the packet going in the forward direction (before the PHB is mapped onto a corresponding PHB in the next differentiated services network). Assuming that the "notification" packet passes through the same ingress edge router as that seen by the packets of the flow in the forward direction. The ECN-capable ingress router responds to the notification by changing the policy of conditioning the incoming traffic. For example, some out-of-profile packets may be dropped for each notification received at the ECN-capable ingress router (section 4.2 suggests other ECN response options). The notification packet is filtered by the ingress router and not propagated further in the direction of the source.

To avoid the implosion of notification messages in a multicast scenario, the ECN-capable router may set ECN on packets on one of the many ECN-capable branches (i.e. a branch where an egress node is known to be ECN-capable). For robustness, a different ECN-capable branch could be chosen each time. ECN-capable egress routers could advertise that information by sending a packet (possibly as a new "INTERNAL" ICMP type, see section 5) in the reverse direction of flows passing through it (for the benifit of multicast routers). Ingress routers should filter these advertisement packets out and not propagate them to the sources of the flows. In case the packet needs to travel through a tunnel, the ECN bits would also need to be copied onto the outer header at the tunnel entrance and copied back into the internal header at the tunnel exit (otherwise ECN-bits set by the routers in the tunnel will be lost).

The architecture described here can be easily made to interoperate

with current and future ECN-capable transport protocols as follows. We assume here that the ECN-enabled transport protocols will set the ECN-capable bit. The routers set the ECN-bit if they are congested. This "end-to-end" ECN-bit could be the same or separate from the ECN-bit used in the edge-to-edge flow control described above. The egress router need not generate a notification packet in the reverse direction when it sees the ECN-capable bit set, because it would expect the application to do that on an end-to-end basis (for example, ECN-TCP would send a notification piggybacked onto its acknowledgements). The ingress router can also relax its containment strategy by waiting for an end-to-end ECN response rather than aggressively controlling the flows using edge-to-edge ECN. Additional controls could be used to monitor and guard against applications which fake themselves to be ECN-capable.

It is completely optional for the router to set the ECN bit and for the ingress edge router to respond to the ECN indication. But the egress router should reset the bit (unless edge-to-edge ECN shares the same codespace as end-to-end ECN (the CU bits, see section 5) and ECN-capable bit is also set) before the packet in forward direction leaves the network. It is also desireable for the ingress edge router to filter the notification packet sent by the egress router irrespective of whether it can or cannot respond to the notification. Having a flow-control or ECN bit will help the network where (key) routers set the bit during congestion and (key) profilers which respond without designing inefficient/incompatible proprietary methods for feedback. We note that, even without a flow-control bit, it is possible to use proprietary methods (eg: using explicit rate-control [TCPRateControl]) between ingress and egress routers. But these would need to use extra packets to work in an IPsec environment (overhead), could be complex to implement (cost), and may have compatibility problems with future ECN-enabled applications/transports.

4.2 Sample Strategies for Congestion Detection and Response

A sample method for the router to set ECN is as follows. Use the RIO scheme to manage the queue. When the average out-of-profile queue length is larger than its min RED thresholds, and a decision has been made to drop an out-of-profile packet, remember to set ECN on the next in-profile packet either received or transmitted from the same flow.

The ingress edge router has several options for responding to the ECN from the network, for example:

- Delay in-profile packets (do not service the in-profile queue)
- Cut the rate of in-profile packets (eg: use multiplicative

decrease)
- Drop out-of-profile packets
- If packets indicate that the application is ECN-enabled, wait for the application to respond to the notification, with a backup plan based on the previous points.

In the most general case, the ingress router can "proxy" as an adaptive, ECN-enabled transport. It could maintain a separate transmission rate for in-profile and out-of-profile packets and vary the rate based upon ECN indications (or the absence of them). A window-based scheme requires acknowledgements (or exchange of credits) to maintain the window which is not available in this architecture.

The out-of-profile drop policy could be adaptive. For example, the discards of out-of-profile packets could be gradually reduced and eventually stopped in the absence of further ECN indications. At the same time, if further ECNs are seen, the drops could be increased aggressively (eg: exponential increase in the number of out-of-profile packets to be dropped). Similarly, the "rate" of in-profile packets can be multiplicatively decreased during consistent ECN indications and additively increased in the absence of ECN indications. Packet dropping could also be randomized. The set of ECN-response strategies which provide adequate performance, stability and robustness is an open research issue.

5. Standardization Issues:

A simple encoding of the ECN-bit for use in differentiated services edge-to-edge control can be done in the bits reserved for ECN and which are currently unused (CU). There are two bits: ECN-capable bit and ECN-bit. The current (non-standard) interpretation of these bits is as follows. The ECN-capable transport would set the ECN-capable bit, and zero the ECN-bit. ECN-capable routers would set the ECN-bit during congestion periods if the ECN-capable bit is set. Floyd [Floyd-ECN] suggests that the destination would simply copy the ECN bits onto the acknowledgement, though additional mechanisms are needed to distinguish the forward and reverse ECN for scenarios involving two-way traffic and piggybacked acknowledgements.

The edge-to-edge flow control can work if the differentiated services ECN-capable routers are configured to mark the ECN-bit even if the ECN-capable bit is not set. The egress router could identify the pattern (ECN-capable not set, ECN-bit set) to trigger the edge-to-edge congestion notification packet, and reset the ECN-bit. The problem with this encoding is that the definition of CU bits is currently out of the scope of the differentiated services group

charter and routers within the differentiated services should ignore their value [Nichols].

An alternative would be to use a bit in the PHB for edge-to-edge ECN which is within the charter of the differentiated services charter. The disadvantage is a coding inefficiency, if it turns out later that the aforementioned CU encoding can be deployed quickly. On the other hand, the advantage is to interpret the PHB-ECN bit as a internal service-differentiator, and as an edge-to-edge ECN-based flow control enabler rather than to interpret edge-to-edge ECN as a special case of end-to-end ECN-based flow control. Further, it may be possible to deploy edge-to-edge ECN even when end-to-end ECN is unavailable for reasons of compatibility or scope of differentiated services standardization.

The compatibility problems (further explained in section 7) stem from the fact that end-to-end ECN facilities may not be available on legacy networks supporting the older TOS semantics. Using a bit in the PHB also allows edge-to-edge ECN-enabled differentiated services to be deployed even before the CU bits are agreed upon. We observe that bits 4 and 5 of the DS-byte are currently undefined. Since edge-to-edge ECN is completely internal to a differentiated services network and requires one bit, the only standardization requirement for edge-to-edge ECN is that at least one of the bits (4 or 5 of the DS-byte) should be left open for internal definition (for edge-to-edge ECN or for additional levels of priority drop/queueing or anything else the administrator wishes).

We have also seen the need to send two kinds of proprietary messages (for ECN notification and ECN-capable advertisements) in section 4.1. These messages are meant for edge-to-edge communications and such messages should be filtered by the border routers/edges of the differentiated services network. The reason a simple IP packet cannot be used with the IP addresses of the edge routers themselves is because packets arriving with the ECN-bit set use source and destination IP addresses and do not indicate edge router addresses. Our proposed solution is to standardize a new ICMP type called "INTERNAL", for proprietary control use within an administrative domain or a differentiated services network. Additional ICMP codes can be configured in a proprietary manner to allow multiple proprietary message types. The "INTERNAL" ICMP type messages could be used in the future for purposes other than differentiated services as well.

6. Security Considerations:

The IPSEC tunneling procedure [ESP] copies the TOS (DS) byte of the encapsulated packet (that is being tunneled) into the outer IP header

at the entrance to the tunnel. However, the byte is currently not copied back onto the header of the encapsulated packet at the exit of the tunnel. Hence, any ECN setting by intermediate routers in the tunnel is lost. Similarly, if one bit in the PHB is left open for internal use in a differentiated services network, the tunnel routers cannot participate in the ECN-based edge-to-edge flow control.

ECN does incur a risk of denial-of-service attacks. Assume that the sources or edges claim to be ECN-capable, but do not respond to ECN effectively. The router can detect this eventually using a RED-plus-penalty-box mechanism and penalize misbehaving flows by dropping their packets. However, the resources used by the dropped packets to reach the router (as well as the resources used during the time taken by the router to detect misbehavior) could have been used to service packets belonging to well-behaving flows - a denial-of-service attack. The deployment of effective ECN-capable edge routers will alleviate this problem because the edge routers will be administered by the same body which wants to guard against denial-of-service attacks.

7. Compatibility issues:

The IPv4 [RFC791,RFC1349] TOS octet does not support ECN marking. As a result, end-to-end ECN marking may not be supported in legacy networks which support the TOS semantics. We note that edge-to-edge ECN-based flow-control scheme could be supported and enforced within the differentiated services "cloud" even though an end-to-end ECN-based feedback facility may not be available.

8. Summary

Using one-bit for ECN-type flow control between the edges of the differentiated services network can potentially lead to service differentiation based upon flow-control methods used. Specifically, there is potential to address some kinds of denial-of-service attacks using an an edge-to-edge ECN mechanism. The out-of-profile packets of a flow can be admitted into the network during the absence of congestion, but kept out (or dropped at the ingress edge) during congestion along the path of the flow. Other advantages include the potential for implementing congestion-based pricing schemes, and building a low-average-delay differentiated service even with a high degree of resource overbooking. The mechanisms can be made to interoperate with ECN-TCP or other ECN-enabled transport/application protocols in the future.

9. Acknowledgements

Thanks are due to Steve Deering for engaging discussions on the

differentiated services mailing list which motivated this draft.

The research was supported in part by DARPA contract number: F30602-97-C-0274.

10. References

[Clark] D. Clark and J. Wroclawski, "An Approach to Service Allocation in the Internet", Internet Draft <draft-clark-diff-svc-alloc-00.txt>, July 1997.

[Deering] S. Deering, communications in integrated services mailing list, Thread: "support for packet drop priority", January 1998.

[Ellesson] E. Ellesson and S. Blake, "A Proposal for the Format and Semantics of the TOS Byte and Traffic Class Byte in IPv4 and IPv6", Internet Draft <draft-ellesson-tos-00.txt>, November 1997.

[ESP] S. Kent and R. Atkinson, "IP Encapsulating Security Payload", Internet Draft <draft-ietf-ipsec-esp-v2-01.txt>, October 1997.

[Ferguson] P. Ferguson, "Simple Differential Services: IP TOS and Precedence, Delay Indication, and Drop Preference, Internet Draft <draft-ferguson-delay-drop-00.txt>, November 1997.

[Floyd-ECN] S. Floyd, "TCP and Explicit Congestion Notification", ACM Computer Communication Review, V. 24 N. 5, October 1994, p. 10-23. URL "ftp://ftp.ee.lbl.gov/papers/tcp_ecn.4.ps.Z".

[Floyd-RouterMech] S. Floyd, and K. Fall, "Router Mechanisms to Support End-to-End Congestion Control", Technical report, February 1997. URL: "ftp://ftp.ee.lbl.gov/papers/collapse.ps".

[Heinanen] J. Heinanen, "Use of the IPv4 TOS Octet to Support Differentiated Services", Internet Draft <draft-heinanen-diff-tos-octet-01.txt>, November 1997.

[IPv6] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", Internet Draft <draft-ietf-ipngwg-ipv6-spec-v2-01.txt>, November 1997.

[Nichols] K. Nichols, B. Carpenter, "Differentiated Services Operational Model and Definitions", Internet Draft,

<draft-nichols-dsopdef-00.txt>, February 1998.

[Qmgmt] B. Braden, D. Clark, J. Crowcroft, B. Davie, S.Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", Internet draft draft-irtf-e2e-queue-mgt-00.txt, March, 1997.

[Ramakrishnan] K. Ramakrishnan and S. Floyd, "A Proposal to Add Explicit Congestion Notification (ECN) to IPv6 and to TCP", Internet Draft <draft-kksjf-ecn-00.txt>, November 1997.

[RED] S. Floyd, and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, p. 397-413. URL "ftp://ftp.ee.lbl.gov/papers/early.pdf".

[RFC791] Information Sciences Institute, "Internet Protocol", Internet RFC 791, September 1981.

[RFC1349] P. Almquist, "Type of Service in the Internet Protocol Suite", Internet RFC 1349, July 1992.

[SIMA] K. Kilkki, "Simple Integrated Media Access (SIMA)", Internet Draft <draft-kalevi-simple-media-access-01.txt>, June 1997.

[Stevens] W. Stevens, "TCP/IP Illustrated, Volume 1", Addison-Wesley, 1994.

[TCPRateControl] R. Satyavolu, K. Duvedi, S. Kalyanaraman, "Explicit rate control of TCP applications," ATM_Forum/98-0152R1, February 1998. Available from http://networks.ecse.rpi.edu/~shivkuma/tm-papers.html

[2BIT] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", Internet Draft <draft-nichols-diff-svc-arch-00.txt>, November 1997.

Authors:

Shivkumar Kalyanaraman, Surendra Arora, Khem Wanglee, Greg Guarriello
Dept of Electrical, Computer and Systems Engg,
Rensselaer Polytechnic Institute (RPI)
110, 8th Street, Room JEC 6003,
Troy NY 12180-3590
Phone: +1 (518) 276 8979
Fax: +1 (518) 276 2433
Email: shivkuma@ecse.rpi.edu, aroras@rpi.edu, wanglk@rpi.edu,
guarrg@rpi.edu

David Harrison
Department of Computer Science
Rensselear Polytechnic Institute
Troy, NY 12180
Phone: +1 (518) 273 2355
Fax: +1 (518) 276 4033
Email: harrisod@cs.rpi.edu