TSVWG
Internet-Draft
Expires: April 27, 2006

J. Babiarz
K. Chan
Nortel Networks
V. Firoiu
BAE Systems
October 24, 2005

Congestion Notification Process for Real-Time Traffic
draft-babiarz-tsvwg-rtecn-05

Status of this Memo

Copyright Notice

Abstract

This document specifies the usage of Explicit Congestion Notification (ECN) markings for real-time inelastic flows such as voice, video conferencing, and multimedia streaming. We build on the principles of RFC 3168, "The Addition of Explicit Congestion Notification to IP", and apply them to real-time inelastic traffic in DiffServ networks. Defined in this document are, new ECN semantics that

provide two levels of experienced congestion along the path for real-time inelastic flows, the required ECN marking behavior in network nodes, and state the required behavior of end-systems that support this function with an explanation of how the two ECN schemes can co-exist safely in the network.

Table of Contents

1. Introduction

Proposed is an additional usage of Explicit Congestion Notification (ECN) for real-time inelastic flows such as voice, video conferencing, and multimedia streaming. RFC 3168 [7] specifies the incorporation of ECN for IP, including ECN's use of two bits in the IP header. This document builds on the concepts of RFC 3168, "The Addition of Explicit Congestion Notification to IP", and applies them to real-time inelastic flows in DiffServ enabled networks to perform admission control and session preemption.

To address a wider usage of this mechanism, it is necessary to introduce new semantics for the ECN field of the IP header (bits 6 and 7 of byte two in IPv4 header) that can provide two levels of congestion indication for real-time inelastic flows. There are applications and services that need to provide different treatment at the application level based on the importance or precedence of the flow for a given level of congestion experienced. For example, situation critical or life threatening VoIP flows within a service class used for real-time traffic may need to get priority access to the network resources over regular VoIP traffic. In the specification of the required behavior for network nodes and end-systems that are configured to provide ECN-capability for real-time flows, we factor in the requirements specified in Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field [17].

The operating environment is discussed first, then a framework is provided, and then functions are defined that need to be performed in the network for real-time flows. Specifically, this includes (1) congestion detection through the use of flow measurement, (2) marking of ECN bits in the IP header of real-time packets for a given DSCP-marked service class and (3) actions that the end-systems needs to take based on the ECN bits marking. Since real-time inelastic flows like voice and video conferencing are very delay sensitive, a different method than what is described in RFC 3168 for determining levels of congestion needs to be used.

The proposal is to use ECN as a method to notify the end-systems that packets flowing on this path are above the engineered capacity of the service class that is used for real-time traffic in the network. Based on this information, the end-system needs to take action to reduce its sending rate by whatever means is appropriate; for example stop sending packets, or reduce its rate, or not admit new flows while the path remains congested. The detailed mechanism used in the end-system to achieve the required behavior to the ECN marking is not specified in this document as it will depend on the application. It is left up to application designers to define how applications in

end-systems should perform the required actions to conform to ECN bit marking in the network. It is expected that application specific documents will be produced to explain the application's usage of this real-time ECN mechanism, one such example is Real-time ECN Use Cases [11] which defines admission control and preemption procedures for VoIP flows. For illustration purposes, a high level example of admission control is provided in this document.

1.1. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [3].

1.2. Applicability and Operating Environment

Networks that need to support real-time inelastic services may need to provide a controlled environment that allows for a high level of guarantees on the quality of service to be honored. We suggest the use of DiffServ service classes [12] to separate the real-time inelastic traffic from the other traffic for such a controlled environment, and applying the Real-Time ECN process discussed in this document.

This document addresses the use of the ECN markings in a DiffServ controlled environment, with both marking as defined herein and in RFC 3168 [7] co-existing in the same network but in different service classes. As well, there may be network segments that do not deploy any ECN processing at all, or there may be a router in the path that does not support DiffServ but performs ECN marking as defined in RFC 3168. These operating environments are explored and discussed herein. But in all cases, DiffServ separation of the real-time inelastic traffic from the other traffic should be supported. With the basic rules of:

- No mixing of Real-Time ECN and RFC 3168 ECN marking in the same service class.
- Allow mixing of traffic from Real-Time ECN capable and incapable end-systems in the same service class and using ECN bits to provide the differentiation.
- Allow traffic from both Real-Time ECN capable and RFC 3168 conformant end-systems as long as the traffic is marked with different DSCP values (using different service classes).

2. Framework

When the packet forwarding network is used to support Session Based Real Time Application usage, the application behavior requires different support from the network. For example, random packet drops does not trigger application to lessen the offered load to the network, it just degrade the session service the application needs. The mechanism described by this document is used within a framework where we try to achieve the following goals:

1. Allow the application control and signaling be separate from the packet forwarding network control and signaling.

2. Keeping the packet forwarding network and its control simple.

3. Allow application features be controlled in the application layer.

The above set of goals point us to an architecture where:

1. The network plays a role of providing current network information concerning the availability of network resource to support the application needs.

2. The application uses the information provided by the network to make application level decisions.

In the context of Session Admission Control for Real Time traffic with Voice Over IP as an example, we will discuss the framework in the following sections.

2.1. Application Decisions based on Network Information

For this document the application decisions is for support of real time traffic session admission control. This includes the decision of:

1. should normal priority real time sessions be admitted.

2. should higher priority real time sessions be admitted.

3. should normal priority real time sessions be terminated.

4. should higher priority real time sessions be terminated.

Notice these are decisions local to the application and the actual actions can be controlled by the use of local application level policies.

For normal priority real time session admission control, the goal is to make sure the additional offered load by the new normal priority real time session does not push the network resource utilization into a state that will negatively affect the packet delay, delay variation, loss requirements of the already admitted real time sessions. Hence the application decision is to admit or not to admit a new real time session based on a set of network information.

For higher priority real time session admission control, the goal is to make sure the additional offered load by the new higher priority real time session does not push the network resource utilization into a state that will negatively affect the packet delay, delay variation, loss requirements of the already admitted real time sessions. Hence the application decision is to admit or not to admit a new real time session based on a set of network information. Notice by the nature of higher priority real time session, the decision can be to admit higher priority session when not admitting normal priority sessions while approaching fuller utilization of network resources. With the notion of normal priority and higher priority sessions being application level notions and the decisions be application level decisions, possibly determined using local application policies.

For normal priority real time session termination decision, the goal is to have a method to decrease the offered load to the network when the network resource utilization approaches the level of affecting the packet delay, delay variation, loss requirements of the already admitted real time sessions. This action is taken due to the fact that for real time sessions like VoIP, the affect will be on all real time sessions that shares the limiting network resource. This decision is to sacrifice a single (or limited number of) real time session on behave of all the other affected real time sessions.

For higher priority real time session termination decision, the goal is to have a method to decrease the offered load to the network when the network resource utilization approaches the level of affecting the packet delay, delay variation, loss requirements of the already admitted real time sessions. This action is taken due to the fact that for real time sessions like VoIP, the affect will be on all real time sessions that shares the limiting network resource. This decision is to sacrifice a single (or limited number of) real time session on behave of all the other affected real time sessions. By the nature of higher priority, the normal priority sessions are sacrificed/terminated before higher priority sessions. This can be controlled using local application level policies.

Notice the action of the decisions are controlled by local application policies, one example is to do nothing when higher

priority real time session termination decision needs to be made, hence not supporting the higher priority session termination function.

We call this Session Admission Control at the application layer.

2.2. Network Information for Supporting Application Decisions

For Session Admission Control of Real Time traffic, due to the nature of real time traffic like VoIP, the network information needs to indicate will the offered load introduced by the to-be-admitted session affect the existing admitted sessions. The application also requires the network to indication when more aggressive action is needed by the application to allow acceptable network service be provided to the admitted sessions. Hence we propose the following network information:

1. Network information for session admission control. This information is used to assist the application to decide if a new application session should be admitted to use the packet forwarding network resource. For real time in-elastic session, this information needs to indicate the on-set of additional packet delay/delay-variation/loss that affects the real time in-elastic sessions sharing the packet forwarding network resource. For normal priority session admission control, we allow the network device along the session's packet forwarding path to indicate if a certain level of resource utilization is reached. We call this Admission-Level, and uses bits per second as the basic measurement unit at the network device. When the network resource utilization at a network device exceeds Admission -Level, the network device indicates this condition by using some field in the packet (proposing the use of the ECN field) being forwarded to the session end point. Please notice that the setting of Admission -Level is a local network provisioning policy for the network device. And it may depend on the link speed being used, other traffic sharing the link resource, and other local network device provisioning criterion. This approach allows the network devices' local provisioning policy to control the allocation of resources of the network device, allowing network devices with different capabilities be used in the session packet forwarding path, and providing the information to the application to assist the making of session admission control decision.

2. Network information for session termination. This information is used to assist the application to decide if the network load offered by the application should be lessened to allow existing application sessions to receive acceptable service from the

packet forwarding network. For real time in-elastic session, this information needs to indicate a more severe on-set of additional packet delay/delay-variation/loss that affects the real time in-elastic sessions sharing the packet forwarding network resource, as compared with session admission control information. For normal priority session termination, we allow the network device along the session's packet forwarding path to indicate if a certain level of resource utilization is reached. We call this Termination-Level, and uses bits per second as the basic measurement unit at the network device. When the network resource utilization at a network device exceeds Termination-Level, the network device indicates this condition by using some field in the packet (proposing the use of the ECN field) being forwarded to the session end point. Please notice that the setting of Termination-Level is a local network provisioning policy for the network device. And it may depend on the link speed being used, other traffic sharing the link resource, and other local network device provisioning criterion. This approach allows the network devices' local provisioning policy to control the allocation of resources of the network device, allowing network devices with different capabilities be used in the session packet forwarding path, and providing the information to the application to assist the making of session termination decision. The Termination-Level indication can also be used for higher priority sessions' admission control decision. For example, when Termination-Level indication is received by the application session decision point, higher priority sessions may not be admitted but their session initiation request being queued, depending on the application level higher priority session admission control policy. There are also mechanisms for smoothing out the session termination process and to provide network information to allow multiple higher level session termination. These will be discussed in the detail mechanism sections of this document.

In the following section, we provide an operational overview using SIP to show how the above information are used together. In later sections we provide the details of the packet forwarding network mechanism itself.

2.3. Operational Overview with SIP Applications

This operational overview section provides some high level indication of how the goals and ideas of previous sections of the framework can be used together in a SIP application environment. We separated the overview into subsections, each describing session admission control operation and session preemption operation.

2.3.1. Session Admission Control

For session admission control application level decisions, the application requires the knowledge of "will the offered load of the new session push the network resource utilization along the application media packet forwarding path (called media path for short) beyond the network resource's ability to support the required service".

The network provides this information by performing a measurement of packet traffic along the media path and comparing the measurement with a preset Admission-Level. Notice that this Admission-Level can be setup using static or dynamic network device configuration. The mechanism described by this document is independent from network control, provisioning, signaling. Hence allowing network devices freedom to setup Admission-Level based on its capability and local network control/provisioning policy.

Since this measurement is performed prior to any application media traffic can flow, we use a pre-media probing method to allow the application to learn this information for session admission control decision. For SIP environment this is described in "RTP Payload Format for ECN Probing" [15]. We also use the SIP application level signaling pre-condition mechanism to indicate the use of this method. Please see A Congestion Status Precondition for the Session Initiation Protocol (SIP) [18] for details.

We provided a simple high level operational walk through below, with the presumption that probing starts at a point during session setup when the Receiver endpoint addressing information is known by the Sender. As the immediate application is admission control, the endpoints involved SHOULD NOT begin alerting or otherwise notifying the user of a new session until the admission control procedures determine whether to admit the new session.

With an unidirectional probing mechanism, after the Sender knows addressing information specific to the Receiver, it can begin generating probe packets to the Receiver. When the probe packets are received at the Receiver, an admission decision can be made based on the congestion level indicated by the probe packets and the priority associated with the session. The process of making this decision may involve a unilateral decision made in the Receiver, or it may involve communication either back to the Sender, or to another device responsible for making the session admission decision.

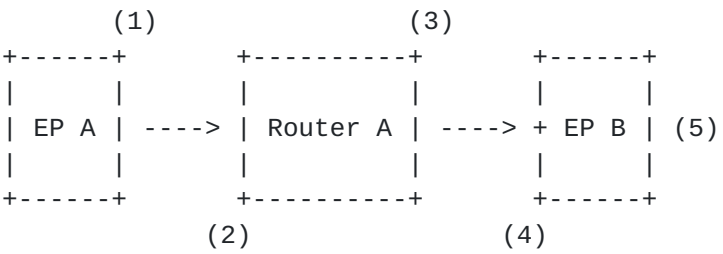

Figure 1: Session Admission Control

1. Endpoint (EP) A starts the process. It creates a Probe Packet and sends it to the address and port it has for EP B.

2. Router A receives the Probe Packet, and applies a metering and marking mechanism for real-time inelastic traffic.

3. Router A re-marks the ECN field in the IP header of the Probe Packet based on the metering and marking mechanism being used, then forwards the packet.

4. EP B receives the Probe Packet. The ECN value received in the Probe Packet in the IP header indicates exceeding the admission control level of congestion on the path towards EP B.

5. EP B inspects the ECN value received in the IP header, and uses it to begin the decision process on whether to admit the session.

In the context of admission control, application level session admission policy may dictate that higher priority sessions may get admitted when the Probe Packets indicate pre-congestion level that prevent normal priority sessions from being admitted.

2.3.2. Session Preemption

For application level session preemption decisions, the application requires the knowledge of "is the current offered load of network traffic pushing the network resource utilization along the application media packet forwarding path beyond the network resource's ability to support the required service for the admitted sessions?". This condition can occur even when session admission control have been applied. For example, packet network routing change can cause previously different session packet forwarding paths to merge at some points in the network and pushed the network

resource utilization at these points to beyond their ability to support the required service. These packet network routing change may be caused by router failures or other reasons, this framework allows the packet forwarding network to use whatever method it chooses to provide the best packet forwarding network service without getting in its way. The application just need to know if it needs to take action based on the resulting packet forwarding network conditions.

The above knowledge can be provided to the application by the network. As with the probe packets described in the context of session admission control, endpoints can similarly monitor the ECN value of RTP media packets and react accordingly, initiating preemption mechanisms when necessary. If the preemption threshold level is exceeded, as indicated by the ECN value of the media packets, application level local policy may dictate session preemption as a required action to keep bandwidth resource usage within engineered limits.

The example provided here shows only a unidirectional media flow. A bidirectional session would operate similarly, but with two unidirectional flows in opposite directions.

The example provided here describes an application level local policy dictating that preemption occurs when the preemption threshold level has been exceeded, as indicated by the ECN value of the media packets.

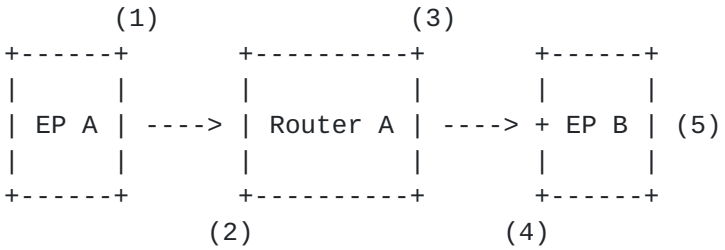

Figure 2: Preemption

1. Endpoint (EP) A starts the process. It creates an RTP media packet and sends it to the address and port it has for EP B. Each media packet is marked with default ECN value to indicate no congestion.

2. Router A receives the RTP media packets, and applies a metering and marking mechanism for real-time inelastic traffic.

3. Router A re-marks the ECN field in the IP header of the RTP media packets based on the metering and marking mechanism being used, then forwards the packet. For the purposes of this example, it is presumed the network resource condition is such that Router A marks the ECN bits to indicate that the preemption level of congestion has been exceeded.

4. EP B receives the RTP media packets. EP B monitors the ECN value of RTP media packets, and upon receiving packets marked with the ECN value indicating that the preemption level of congestion has been exceeded somewhere along the path in the network, it follows local application level session preemption policy to initiate session preemption.

5. The specific actions taken to carry out preemption may vary, as they are local application level session preemption policy.

2.3.3. Operational Overview Summary

The simple high level walk through of operational overview for session admission control and session preemption in previous sections provided some overview of how we:

1. Keep the application features being controlled in the application layer using application level session policies.

2. Keep the packet forwarding network and its control simple by just adding simple network condition indication to the forwarded packet with existing packet header field.

3. Allow the application control and signaling be separate from the packet forwarding network control and signaling. And allowing the packet forwarding network to do what it does best without adding application level baggage to it. And allowing the application layer to do what it does best on utilizing the packet forwarding network resources.

3. General Principles

In this section, some of the important design principles and assumptions guiding the development of this proposal are described.

- Because ECN for real-time flows is likely to be adopted gradually and selectively in nodes, accommodating migration and selective

deployment is essential. Some nodes may not be able to detect congestion or mark the ECN bits within IP packet headers. Also there may be parts of the network where congestion is very unlikely and therefore there is no need for an ECN function. The most viable strategy is one that accommodates selective or incremental deployment in a network with both ECN-capable and non-ECN-capable nodes.

- Asymmetric routing is likely to be a normal occurrence within IP networks. That is, the path (the sequence of links and nodes) taken by forward and reverse packet flows may be different.

- Many nodes process the "regular" header in IP packets more efficiently than they process the header information in IP options. This suggests that the ideal approach is to keep "congestion experienced" information in the regular header of an IP packet.

- A specific DiffServ service class would be implemented for real-time traffic. The assumption is that a signaling protocol (SIP, H.323, MGCP, H.248, etc.) will be used to determine if the end-systems are capable of understanding ECN bit marking and thus, are willing to participate in congestion control prior to marking packets as ECN-Capable Transport (ECT).

- Flow measurement and marking of ECN bits as defined herein is performed on flows from ECN-capable end-systems that is mapped to a ECN-enabled service class, and may be performed on selected node links in the network where congestion is likely to occur. Other traffic flows are not affected by this function. Nodes that do not support this function forward packets without modifying the ECN field of the IP header.

- ECN procedure as defined in RFC 3168 [7] may also be applied to DiffServ service classes in the IP network. Both methods may co-exist in the network, but in different DiffServ service classes.

4. Definition of Congestion for Real-Time Traffic

Real-time traffic generated by applications such as voice, video conferencing, and multimedia streaming have different performance requirements when compared with non-real-time elastic applications that use a protocol such as TCP. One such requirement is that end-to-end delay be bounded to a small value, and that packets should not be dropped.

It is generally accepted that such performance requirements can be

achieved when the real-time flows are serviced by the nodes in their path through a real-time service class such as one based on the Expedited Forwarding (EF) PHB [8] treatment. This treatment can be provided only when the real-time service class is not overloaded (i.e., when the aggregate of input traffic never exceeds the class' capacity, and thus no congestion condition occurs). It should also be noted that when the overloaded condition occurs, all real-time traffic flows within the real-time service class at the congestion point will be affected, not just the offending traffic flow. Hence, it is desirable to avoid the overloaded condition as much as possible.

With the above performance requirements for real-time inelastic traffic in mind, "congestion of real-time inelastic traffic" is defined to be the network condition when aggregated packet flows within the service class exceed an engineered traffic level. The engineering of the network is such that traffic exceeding this engineered traffic level by a defined and limited amount does not generally cause an increase in packet queuing or packet dropping (service class overload) in the network. Instead, the ECN field is used to provide an indication that traffic is above the engineered traffic level. This can be viewed as explicit notification to prevent congestion. However, uncontrolled or prolonged increase in traffic above the defined amount may result in an increase in packet queuing and/or packet dropping, and therefore may cause overload of the real-time service class.

4.1. Avoiding Congestion for Real-Time Traffic

Congestion notification can be utilized in a flow admission control scheme to ensure sufficient forwarding resources (bandwidth), hence for Real-Time Traffic, ECN is used to prevent congestion. In this scheme, a continuous process at selected link(s)/node(s) measures the traffic going through a specified real-time service class and indicates a level of congestion (such as "not congested", "mildly congested" or "severely congested"). This congestion indication as described in Section 4.4.2 is then used by the application to select the action that will be taken by the application controlling the service. The action could be to admit or not to admit a new flow into that real-time service class in the network, or have the sending rate of ECN marked flows reduced or stopped, or terminate a flow. All with the effect of reducing level of offered traffic. Based on the performance requirements of real-time traffic, it is desirable that the measurement process indicate congestion of real-time traffic before any significant packet accumulates in the queue occurs. This is such that no significant queuing delay is added to existing real-time flows' end-to-end delay.

An alternative method to avoid the overloaded condition of a service class is through resource reservation and admission control. A (centralized or distributed) database maintains a record of available resources (bandwidth) for the real-time service class on each link in the network and a new flow is admitted only if there are enough resources on the links in its path so that the overloaded condition is avoided. Checking for available resources can be done with a reservation protocol or through a policy based protocol. An important issue is that the maintenance and per-flow querying of the resource database in conjunction with the routing database is an important overhead that is undesirable in many implementations.

4.2. Congestion Detection for Real-Time Traffic

One of the goals is to keep the amount of processing that is performed in the network to be very small and not require any other computations or state information to be kept in network nodes. One way to achieve this is through monitoring the aggregate rate of traffic in the specified real-time service class and to indicate congestion when a certain traffic threshold is exceeded. Hence the network nodes need only to perform flow measurement of packets marked with specific DS codepoint and with ECN indication that the end-system is ECN-capable. Another policy that may be used is where network nodes perform flow measurement of packets marked with a specific DS codepoint and if the rate is exceeded, only ECN mark packets that indicate that they are ECN-capable. The application monitors the ECN field, and takes an appropriate action based on the marking.

Figure 3 below shows a block diagram of the traffic measurement and ECN marking function.

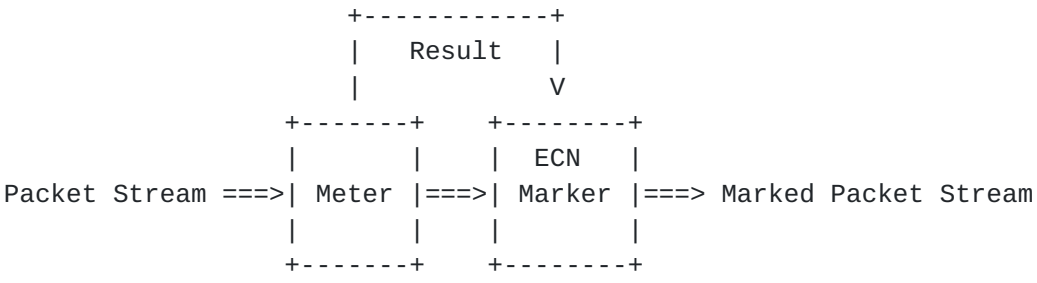

Figure 3: Block Diagram of Meter and Marker Function

The Meter meters each packet within the real-time service class and passes the packet and the metering result to the ECN Marker.

The Marker sets the ECN congestion level for each packet that is marked as ECN-capable within the real-time service class based on the results of the Meter.

The traffic rate of the specified service class may be measured with a simple token bucket meter, an exponentially weighted moving average meter, or other methods. The goals of a rate measuring method are simplicity and minimum or no added delay to traffic forwarding.

The specification of the traffic measurement mechanism is outside the scope of this document. The intention is that an existing traffic measurement mechanisms may be used. In Appendix A, an example of a simple token bucket method for measurement and marking is provided.

4.3. Behavior of Meter and Marker

When the measured rate exceeds the engineered traffic level for ECN-capable marked packets, the Meter sets its result flag and passes it to the Marker. The Marker, sets the appropriate ECN value for all ECN-capable marked packets belonging to the service class that is measured until the result flag from the Meter is cleared.

When the measured traffic rate is reduced below the engineered rate the Meter clears the result flag if set. The clearing of the result flag output from the Meter stops marking ECN bits by the Marker. The metering function has built-in hysteresis for setting and clearing the result flag. The amount of hysteresis is controlled by the configuration parameters (example size of the token bucket) of the traffic measurement mechanism and should be configured to meet the characteristics of the real-time inelastic traffic that is being measured. See report, Simulation of RT-ECN based Admission Control and Preemption [13] for results of the above metering and marking as well analysis of different metering approaches in Discussion of Congestion Marking with RT-ECN [14].

4.4. Marking for Congestion Notification

Marking for Explicit Congestion Notifications is done through the use of the two ECN bits in the IP header.

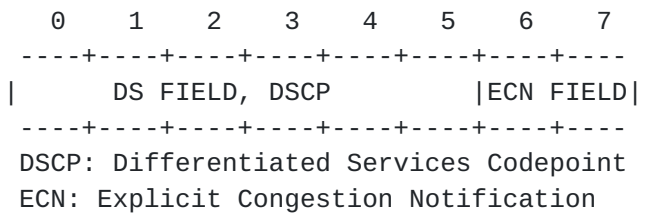

Figure 4: DS and ECN Fields in IP Header

Bits 6 and 7 in the IPv4 octet are designated as the ECN field. This octet of the IPv4 header corresponds to the Traffic Class octet in IPv6, and the ECN field is defined identically in both cases. The

definitions for the IPv4 TOS octet RFC 791 [1] and the IPv6 Traffic Class octet have been superseded by the six-bit Differentiated Services (DS) field RFC 2474 [4], RFC 2780 [6]. Bits 6 and 7 are listed in RFC 2474 [4] as Currently Unused, and are specified in RFC 2780 as approved for experimental use for ECN. Finally, RFC 3168 [7] standardizes the use of the ECN bits.

4.4.1. ECN Marking of Real-Time Inelastic Flows

Marking of ECN bits for real-time inelastic flows is defined so that nodes in the path need only to perform an ECN set function when an engineered rate is exceeded for packets that are marked as ECN-capable. With this approach there is no need to perform a test to determine the congestion level marking of the received packet before a node performs ECN marking. Other approaches could be used, but for simplicity we have chosen this one. In Section 7 we discuss the different marking approaches that were studied.

Network nodes that are configured to support congestion notification for real-time flows need to provide the following capability:

- Congestion detection of packets marked indicating that the end-system is ECN-capable (ECT) SHOULD be performed using a real-time measurement mechanism (e.g., flow metering).

- When the flow rate exceeds configured rate "A" (i.e., the first level of congestion), ECN bit 7 of ECT marked packets is set to '1'.

- When the flow rate exceeds configured rate "B" (i.e., the second level of congestion), ECN bits 6 and 7 of ECT market packets are set to '01'.

- Measured rate "B" SHOULD be greater than rate "A".

4.4.2. ECN Semantics for Real-Time Traffic

Some real-time applications or services need the indication of two levels of congestion experienced, CE(1) and CE(2), for first and second level respectively. Other applications may only need a single indication. To address a wide range of usage, we have selected the following ECN semantics for real-time inelastic traffic.

ECN Marking:

Bits 6 and 7 values		
0	0	Not-ECT - End-system is Not ECN-Capable Transport
1	0	ECT(0) - End-system is ECN-Capable Transport
1	1	CE(1) - Congestion Experienced at 1st level
0	1	CE(2) - Congestion Experienced at 2nd level

Figure 5: ECN Semantics for Real-Time Flows

4.5. End-System Behavior

Listed below are reactions to ECN marking that needs to be supported by end-systems that indicated that they are ECN-capable:

- Under normal (non emergency or situation critical) conditions during new flow establishment, ECN-capable end-system that receive packets marked indicating that congestion was experienced at 1st level CE(1), SHOULD NOT proceed with the session setup. New flows that indicate that they are "emergency or situation critical" SHOULD be allowed admission at CE(1). The definition and verification of what flow is classified as "emergency or situation critical" is left up to the application.

- Once a flow has been admitted and the path experience 1st level of congestion CE(1), ECN-capable end-systems that have ability to dynamically react to ECN marking, SHOULD reduce their sending rate as agreed to during session setup.

- During new flow establishment, ECN-capable end-systems that receive packets marked indicating that congestion was experience at 2nd level CE(2) SHOULD NOT proceed with session setup.

- Systems with established flows that experience the 2nd level of congestion CE(2) SHOULD terminating sessions or reduce sending rate in a controlled manner, and continue until the congestion level drops below the 2nd level of congestion.

Generally, at 1st level of congestion, no additional traffic should be added with the exception for flows that are identified as being potentially life threatening i.e., E911 or situation critical. End-systems that have the ability to reduce their sending rate should do so. At 2nd level of congestion, no additional traffic is added to the congested path plus selected flows on the congested path are removed to reduce congestion below 2nd level.

5. Detection of Inappropriate Changes to the ECN Field

This section discusses in detail some of the possible inappropriate changes to the ECN field in the network, such as reducing level of congestion, falsely reporting no congestion, or modifying ECN bits to indicate that the path is or is not Real-Time ECN compliant. Reducing the level of congestion or falsely reporting no congestion will be referred to here as "cheating".

The Real-Time ECN mechanism provides two levels of congestion indication. Therefore, the cheating detection mechanism as defined in RFC 3168 [7] and RFC 3540 [10] that uses ECT(0) and ECT(1) states can not be used. Instead, the procedure outlined in this memo SHOULD be used to detect if inappropriate changes to the ECN bits have occurred between originator and receiver. The outlined procedure is executed under the control of the application prior to admission of a new real-time flow. The testing for conformance is performed between the two Real-Time ECN-capable and conformant applications running in the end-systems. These two end-systems will be referred to as sender and receiver.

In the implementation of a Real-Time ECN mechanism in the network, the network administrator through the use of policies or through the use of signaling/control protocols such as SIP, can verify the capabilities and conformance of the end-systems. As stated earlier, only applications running in end-systems that are capable and conformant to this Real-Time ECN mechanism may use it. End-systems that are not Real-Time ECN-capable or conformant MUST set ECN field to '00' when sending packets.

5.1. During Session Setup

During the admission of a new real-time flow, application signaling is used to exchange end-system capabilities, including whether the end-systems are Real-Time ECN capable. The Real-Time ECN mechanism defined in this memo can be used to provide admission control capabilities to end-systems. In order to provide this capability, the signaling used by the end-systems requires a means to both trigger the Real-Time ECN probing process as described in "RTP Payload Format for ECN Probing" [15] and "Real-time ECN Use Cases" [11], as well as to delay session setup; - specifically user indication of a new session, i.e., ringing until an admission control decision has been made. In the context of SIP, preconditions can be used to accomplish both of these requirements. For more details, see A Congestion Status Precondition for the Session Initiation Protocol (SIP) [18].

If signaling indicates that the destination end-system does not

support the Real-Time ECN mechanism, the originating end-system MAY reattempt the session without stipulating the use of the Real-Time ECN mechanism. While such a session would not undergo the Real-Time ECN admission control tests, this does allow session creation between end-systems that support the Real-Time ECN mechanism and those that do not.

Prior to admission of a new real-time flow, the following procedure SHOULD be used to detect inappropriate changes to ECN field. Note that this procedure can be independent of an actual admission control procedure.

Under the control of the application, the sender generates and sends a stream of RTP-probe packets. This stream of RTP-probe packets is specified as follows:

1. A random sequence of the four possible ECN bit combinations is selected, and this sequence of values is used as the ECN field values on the initial four RTP-probe packets. The ECN field marking is also copied into the RTP-probe payload [15] and the payload should be secured. SRTP RFC 3711 [9] is one mechanism that can provide the necessary security for the probe packets.

2. After the sequence of four RTP-probe packets is sent, the sender sends RTP-probe packets with randomly generated ECN bit of '10', '11' and '01' until probing is stopped. As before, the ECN field marking is also copied into the RTP-probe payload and the payload should be secured.

The probing sequence just described is one method that can be implemented for probing to detect inappropriate changes to the ECN field. Because the ECN field marking on probe packets is always copied into the RTP-probe payload, the receiving end-system is always able to compare the received ECN value in the IP header with that in the secured RTP-probe payload. Therefore the ability to detect inappropriate changes to the ECN Field via probing is independent of the ECN bit sequence used.

Upon reception of the RTP-probe packet, the receiver compares the received ECN marking in the IP header to ECN marking in RTP-probe payload.

A packet sent with ECN field '10' and received as indicated below have the meanings as indicated:

o Received as '10': path is valid, no congestion.

- Received as '11': path is valid, congestion experienced at 1st level.
- Received as '01': path is valid, congestion experienced at 2nd level.
- Received as '00': path is not valid, device in path zeroed ECN field.

Packet sent with ECN field '01' and received as indicated below have the meanings as indicated:

- Received as '01': path is valid.
- Received as '11': path is not valid, device in path reduced congestion experienced to 1st level, this could also mean that the path is through a congested router that used RFC 3168 for ECN marking of this flow.
- Received as '10': path is not valid, device in path cleared indication of congestion experienced.
- Received as '00': path is not valid, device in path zeroed ECN field.

Packet sent with ECN field '11' and received as indicated below have the meanings as indicated:

- Received as '11': path is valid.
- Received as '01': path is valid, congestion experienced at 2nd level.
- Received as '10': path is not valid, device in path cleared indication of congestion experienced.
- Received as '00': path is not valid, device in path zeroed ECN field.

Packet sent with ECN field '00' and received as indicated below have the meanings as indicated:

- Received as '00': path is valid.
- Received other than '00':, path is not conformant.

The above mechanism will detect when a device in the path tries to cheat by changing the ECN field to indicate no congestion, or reduce

the level of congestion, or incorrectly indicate a path is or is not ECN capable. Due to the nature of the Real-time ECN process described in this memo, it is not possible to detect the presence of devices which raise the level of congestion in the ECN marking.

The detection of presence of devices that lower ECN marking is only possible if there are no other Real-Time ECN capable routers down stream from the cheating device along the network path legitimately marking the ECN bits, masking out the cheating condition. However, this is not a problem if the down steam router marks ECN back to the appropriate congestion level, as the end-system will take the correct action for that flow.

The outlined procedure for detection of inappropriate changes to the ECN field is also applied to RTP-probe flow in the direction from call originator to the far-end as outlined in Real-time ECN Use Cases [11].

During session setup phase, if it is determined that the path is invalid, non-conformant or congested, the setup of the session SHOULD be terminated with cause information provided to the user.

5.2. After Session is Established

Once a new real-time flow has been admitted, the following procedure SHOULD be used to detect cheaters and routers that use ECN procedure defined in RFC 3168:

- For real-time flow, the media packets are sent with an ECN value of '10' set in the IP header. At random intervals, a single media packet will be sent with '01' in order to detect the cheaters and congested RFC 3168-capable routers. The receiver must be aware of which packets the sender marked with '01', so that it does not misinterpret such packets as indicating congestion.

- In order that both the sender and receiver are synchronized as to which packets are marked '01', they will use a common function to determine which packets are marked. The Mersenne Twister MT 19937 [16] random number generator, seeded with the initial RTP Sequence Number used by the sender, is utilized in this synchronization process.

- During the initial RTP-probe sequence [15] that was used to verify conformance of path during session setup, the initial Sequence Number used by the sender is sent to the receiver in the payload of the probe packets. This value is used as a seed in a Mersenne Twister MT 19937 random number generator in both the sender and receiver. The sender begins sending media packets marked with ECN

'10'. The sender and receiver each use the MT 19937 function to select the next pseudorandom number in the sequence, modulus 4, to determine the next packet to mark with ECN '01'. The use of modulus 4 is to keep the interval between packets used for detection of cheaters, etc., bounded.

In the sender, the following steps describe this process:

1. During probing, send the Initial RTP Sequence Number (IRSN), in the payload of the probe packet.

2. Seed the MT 19937 function with the initial sequence number: MT19937.seed(IRSN).

3. Calculate N, the number of media packets to be marked with ECN '10', before marking a single media packet with ECN '01', as follows: N = (MT19937.next)(MOD 4) + 1.

4. Once the real-time flow begins, mark N packets with ECN '10'.

5. Send the next packet with ECN '01'.

6. Repeat starting with Step 3, until the real-time flow is terminated.

In the receiver, the following steps describe this process:

1. During probing, pull the Initial RTP Sequence Number (IRSN) from the payload of a received probe packet[15]. Initialize NSN (described in Step 4 below): NSN = IRSN.

2. Seed the MT19937 function with the initial sequence number: MT19937.seed(IRSN).

3. Calculate N, the number of media packets to be marked with ECN '10', before marking a single media packet with ECN '01', as follows: N = (MT19937.next)(MOD 4) + 1.

4. Calculate NSN, the sequence number of the next packet expected to be marked with ECN '01': NSN = NSN + N.

5. Receive media packets, watching for a packet with a sequence number equal to or greater than NSN, until the real-time flow is terminated:

   A. Packets with a sequence number less than NSN are used for congestion detection. Repeat Step 5.

B. Packets with a sequence number equal to NSN are used for detection of cheaters and congested RFC 3168-enabled routers. If the ECN value of this packet is not '01', a cheater or congested RFC 3168-enabled router is present along the network path. Jump to Step 3 and repeat.

C. Packets with a sequence number greater than NSN should result in calculation of a new NSN value. Jump to Step 3 and repeat.

The previous algorithm is one that should be used. Other methods can be used as long as both the sender and receiver agree to it.

Upon receipt of an RTP packet the receiver expects to be marked '01', the end-system compares the received ECN field with the expected value of '01', or CE(2). If a cheater is present, and is not being overridden by marking of one or more Real-Time ECN capable routers along the path through the network, the end-system detects the presence of a cheater if the received ECN value is '10' or '11' or '00', (ECT(0) or CE(1) or Not-ECT). Also, this procedure can be used to detect the presence of congested routers that use RFC 3168 as its ECN marking. A congested router that uses the ECN marking procedure as defined in RFC 3168 will interpret ECN value '01' as ECT(1) and change the marking to '11' (CE).

If after a session has been admitted it is detected that there is a cheater or congested router that uses RFC 3168 for ECN marking of real-time flows, the established session SHOULD be terminated with cause information provided to the user.

6. Network with DiffServ and Real-Time ECN Support

The real-time ECN process requires that the real-time inelastic traffic is separated from the other traffic. Within a DiffServ network, it is perfectly fine to deploy RFC 3168 ECN marking for service classes that are used for elastic TCP traffic and to deploy Real-Time ECN marking as defined herein for service classes that are used for inelastic real-time traffic. DiffServ is used to separate the real-time traffic from the other traffic flows, and Real-Time ECN processing is applied to this separated traffic to provide control within the service class. Under this condition, the most optimal deployment is to have all network segments support DiffServ, with Real-Time ECN marking capability on selected nodes where congestion within the real-time service class is likely. Over time, as traffic levels within the real-time service class become complex and/or the network topology becomes more complex, it may be preferable that Real-Time ECN capability is extended to all or most network nodes.

This approach allows for incremental deployment of DiffServ and Real-Time ECN. Specific network nodes where congestion is not likely to occur may delay its use of DiffServ and ECN.

6.1. Workable Approach for Co-existence of RT-ECN

Routers need a method to understand which ECN mechanism should be applied based on the traffic being forwarded. We believe DiffServ architecture can provide this capability. The normal practice is to segregate elastic and real-time inelastic flows into different service classes that use different PHBs with the DS codepoint being used as the indicator for selection of the PHB. Router can implement both ECN procedures and use DS codepoint for indicating which ECN mechanism applies to each packet.

The approach that we selected allows for two different ECN mechanisms, one for elastic flows and the other for real-time inelastic flows with the following rules for deployment:

1. ECN as per RFC 3168 applies to the Default Forwarding [4] PHB '000000'.

2. In DiffServ enable networks, DS codepoints is used to indicate the ECN mechanisms that may be supported.

   A. ECN as per RFC 3168 should be used with the AF [5] PHBs.

   B. RT-ECN should be used with EF [8] PHB.

   C. Class Selector PHBs may use RFC 3168 or RT-ECN and should be based on traffic characteristic assigned to the PHB. See Configuration Guidelines for DiffServ Service Classes [12] for guidance.

3. As RFC 3168 defines the default behavior, all other mechanisms that are defined should not cause harm to the default ECN behavior or network if the alternate ECN mechanism encounters RFC 3168 marking.

As stated in Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field [17] "that the default ECN semantics defined in RFC 3168 are the current default semantics for the ECN field, regardless of the contents of any other fields in the IP header. In particular, the default ECN semantics apply for more than best-effort traffic with a codepoint of '000000' for the diffserv field - the default ECN semantics currently apply regardless of the contents of the diffserv field." We propose that an amendment to RFC 3168 be considered that would allow non default DS codepoints

to be used for indicating alternate ECN semantics with guidance as stated above.

7. Discussion on Different Marking Approaches

In our analysis of ECN marking to address the needs of real-time flows like voice and video conferencing, we look at different marking schemes so that the ECN bit definition and meanings be closely aligned with what is defined in RFC 3168. Below is one example worth noting that was considered but rejected even if it uses the same ECN semantics as what is defined in RFC 3168.

7.1. Alternate RT-ECN marking

During session setup (flow admission), end-system marks its' packets with ECT(0) ECN='10' and once the flow is admitted the end-system marks its' packets with ECT(1) ECN='01'.

With the following metering and marking policy in router, packets that are marked as ECN-capable (ECN= 10, 11 or 01) are meter by both meters:

- If traffic exceeds 1st threshold level, mark ECN-capable ECT(0) '10' packets to indicate that congestion was experienced CE '11'.

- If traffic exceeds 2nd threshold level, mark ECN-capable ECT(1) '01'packets to indicate that congestion was experienced CE '11'.

ECN Marking (identical to RFC 3168):

```
Bits 6 and 7 values
     0     0    Not-ECT - End-system is Not ECN-capable
     0     1    ECT(1)  - End-system is ECT, once flow is admitted
     1     0    ECT(0)  - End-system is ECT, during flow admission
     1     1    CE      - Congestion Experienced
```

Figure 6: Alternate ECN Semantics for Real-Time Flows

7.2. Implication of this Marking

During session setup (admission control) end-systems send their packets marked with ECT(0) '10' and once the flow is admitted, end-systems mark their packets with ECT(1) '01'. Routers have a policy that meters all ECN-capable traffic, packets marked with ECT(0) are measured to the 1st threshold level and packets marked with ECT(1) are measured against the 2nd threshold level. The receiving end-system must be informed (some how) if the packet is ECT(0) or ECT(1),

during RTP-probing as well after, and there are methods to achieve this.

7.3. Reason for Not Using this RT-ECN Marking

The concern that the authors of this draft had with the use of the same ECN semantics for two different congestion control mechanism is that there is no simple method for detection of the wrong ECN behavior being applied by router in the path. As stated in Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field [17] a method is needed for end-systems to verify that routers are applying the compliant marking for safe co-existence of alternate ECN behavior. For the proposed ECN semantics (see Section 4.4.2), and Section 5 of this document defines a method that can be used to detect the presence of congested routers that use RFC 3168 for its ECN marking. As well, we would like to note that the procedure outlined in RFC 3540 can be used to detect presence of congested routers that use RT-ECN marking.

8. Example of ECN usage for Admission Control

Normally real-time VoIP bearer traffic is marked with EF DSCP and is mapped into a DiffServ service class that produces very low latency, jitter and packet loss when the traffic load is within the specified parameters. Currently there is no method defined that can limit (without dropping packets) the amount of traffic that can be aggregated onto a link. As a result, controlling loads to within engineered limits is difficult. To address this issue, we propose that for real-time flows we use the metering and ECN marking method defined in this document.

Here we describe how ECN can be used in real-time VoIP solution to provide end-to-end admission of new media flows. This is only a simple example of how admission control may be implemented using rate metering and ECN bit marking in the network. Different applications may use modified approaches to verify if there is sufficient bandwidth before admitting a new flow.

Let us assume that the network is configured to mark real-time VoIP payload packets with EF DSCP, and only this traffic is mapped into a DiffServ service class referred to as Telephony service class. Mapping of real-time traffic marked with other DSCP values is possible but to keep this example simple we will only talk about EF marked packets.

For example, before a session (i.e., a call) is established between two clients, the two endpoints involved in the call will execute a

request/response transaction where the called party (Client B) sends a Request probe packet to the calling party (Client A) and the calling party correspondingly sends back a Response probe packet to the called party. Probe packets are marked with EF DSCP and are mapped into the Telephony service class.

A DiffServ style traffic conditioner, meter and ECN marker are used on selected nodes in the network along the path to measure the aggregated (real-time media and probe packets) flow rate of EF marked packets. If the flow rate of the EF marked packets as measured by the meter is greater than rate "A", bit 7 in the ECN field of IP header is set to 1 and the packet is forwarded as usual. The metering and marking of ECN bit only needs to be performed on selected nodes where bandwidth constraints exist and where congestion is likely to occur.

Upon receipt of the Request probe packet, the calling party generates and sends a Response probe packet to the called party, echoing the status of the received ECN bits in the Response probe packet. Again, a DiffServ style traffic conditioner, meter and ECN marker are used on selected nodes in the network along the reverse path to measure the aggregated flow rate of EF marked packets. If the flow rate of EF marked packets as measured by the meter is greater than rate "A", bit 7 in ECN field of IP header is set to 1 and the packet is forwarded as usual. On receipt of the Response probe packet, the called party could send a notification with the ECN Status to relay the ECN bit status results for the media path to a server in the network where call admission control is performed. Based on the received congestion status (bandwidth usage) for that path, the admission control function will make a decision as to whether or not to continue with call setup and admit this new real-time flow. Should bandwidth usage parameters as indicated by ECN bit marking be exceeded, then this new real-time flow will not be admitted.

9. Non-compliance

Because of the unstable history of the TOS octet, the use of the ECN field as specified in this document cannot be guaranteed to be backwards compatible with any past uses of these two bits that pre-date ECN. The potential dangers of this lack of backwards compatibility are discussed in RFC 3168 [7] Section 22.

10. Changes from Previous Version

NOTE TO RFC EDITOR: Please remove this section during the publication process.

This version of the draft was rework to address the requirements stated in Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field [17], updated section that define "Detection of Inappropriate Changes to the ECN Field" and changes to Appendix A.

11. Security Considerations

This document discusses detection of congestion for real-time traffic flows and also describes a common policy configuration, for the use and application of ECN bit marking. If implemented as described, it should require the network to do nothing that the network has not already allowed. If that is the case, no new security issues should arise from the use of such a policy.

It is possible for the policy to be applied incorrectly, or for a wrong policy to be applied in the network for the defined congestion detection point. In that case, a policy issue exists that the network must detect, assess, and deal with. This is a known security issue in any network dependent on policy-directed behavior.

A well known flaw appears when bandwidth is reserved or enabled for a service (for example, voice transport) and another service or an attacking traffic stream uses it. This possibility is inherent in DiffServ technology, which depends on appropriate packet markings. When bandwidth reservation or a priority queuing system is used in a vulnerable network, the use of authentication and flow admission is recommended. To the author's knowledge, there is no known technical way to respond to or act upon a data stream that has been admitted for service but that it is not intended for authenticated use.

12. IANA Considerations

To be completed.

13. Acknowledgements

The authors acknowledge a great many inputs, most notably from Sally Floyd, Nabil Bitar, Hadriel Kaplan, David McDysan, Mike Pierce, Alia Atlas, John Rutledge, Francois Audet, Tony MacDonald, Mary Barnes, Greg Thor, Corey Alexander, Jeremy Matthews, Marvin Krym, Stephen Dudley, Bob Briscoe, David Black, and Ralph Carsten.

14. Appendix A

Below is a description of a Single Rate Meters and ECN Marker. For scenarios that require two traffic levels within a service class to be measured for congestion indication, two instances of the single rate meter can be used, one for configured rate "A" and the other for configured rate "B", with a single ECN marker. The meter parameters should be selected to meet the characteristics and performance requirements of traffic being measured.

14.1. Introduction

The Single Rate Meters and ECN Marker is configured by assigning values to four parameters: Committed Information Rate (CIR), Token Bucket Size (TBS), ECN set threshold 'm' as a percentage of TBS and ECN clear threshold 'n' as a percentage of TBS. The meter has an internal state "result flag" which when set indicates a condition where the measured traffic has exceeded the CIR and token in the token bucket where exhausted below 'm' threshold. When the rate decreases below CIR, token bucket is filled above 'n' threshold and the internal "result flag" is cleared.

The Meter meters each packet within the real-time service class and passes the packet and the metering result to the Marker:

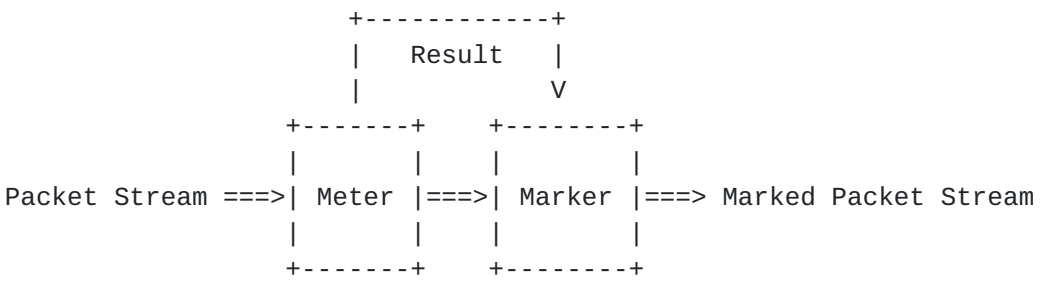

Figure 7: Block Diagram of Meter and Marker Function

The Marker sets the ECN bit values for each packet within the real-time service class based on the results of the Meter.

14.2. Meter Configuration

The Single Rate Meter and ECN Marker is configured by assigning values to four traffic parameters: Committed Information Rate (CIR), Token Bucket Size (TBS), and values 'm' and 'n' representing percentage fullness of Token Bucket. CIR is measured in bytes of IP packets per second, i.e., it includes the IP header, but not link specific headers TBS is measured in bytes, and represents the variants of the rate being measured. 'm' and 'n' are percentages that change size of TBS depending on the state of the result flag. Normally, variable rate traffic will need larger token bucket than

constant rate traffic, and the size will depend on the characteristics of traffic being measured. TBS should be configured such that traffic variation within the specified rate as measured at the node should not use up all the available tokens during a single measurement duration.

14.3. Meter Behavior

The behavior of the Meter is specified in terms of its Committed Information Rate (CIR), Token Bucket Size (TBS) and its ECN set and clear thresholds 'm' and 'n'.

Initially (at time 0), token bucket (TBS) is full, i.e., the token count is represented by Tp.

Where Tp(0) = TBS

As well, the internal meter result flag (1st_result_flag) is cleared.

When a packet of size B arrives at time t, the following happens:

```
Tp = Tp(t) + (CIR x delta_t)     //tokens added at CIR
Tp = Min (Tp(t), TBS)            //keep Tp from growing pass full
Tp = Tp(t) - B                   //decrement Tp by B bytes
Tp = Max (0,Tp(t))               //keep Tp from going negative
if (1st_result_flag = = 0)       //internal result_flag not set?
    if (Tp < TBS x m)            //has traffic exceeded CIR?
        1st_result_flag = 1      //set internal result_flag
        Tp = 0                   //set token bucket to zero
else                             //if internal result_flag is set
    if (Tp > TBS x n)            //is traffic below CIR?
        1st_result_flag = 0      //clear internal result_flag
        Tp = TBS                 //set Tp to TBS
return
```

Where m and n is 1-99%

Notes:
- delta_t is the elapsed time from the previous received packets.
- m controls token bucket threshold for setting the result flag.
- n controls token bucket threshold for clearing the result flag.

A second instance of this single rate meter can be configured for measurement of rate "B" (the second rate).

The actual implementation of a Meter doesn't need to be modeled according to the above formal specification.

14.4. Marking

The ECN Marker reflects the result flag setting received from the Meters. If 1st_result_flag is set, all packets indicating that the end-system is ECN-capable, have their ECN bit 7 set to '1'. If the 2nd_result-flag is set, all packets indicating that the end-system is ECN-capable have their ECN bits 6 and 7 set to '01'. The ECN Marker sets the ECN bit of ECN-capable marked packets as long as the result flag(s) from the meters is set.

14.5. Summary of the Behavior

When the measured rate is exceeded (token bucket runs out of tokens) the meter sets the "result flag" and passes it to the ECN Marker. The ECN Marker, sets the ECN bit(s) of all ECN-capable marked packets marked with a specific DS codepoint flowing through the interface being measured until the traffic rate is reduced below the measuring threshold; thereby the token bucket becomes full. When the token bucket becomes full, the meter clears the "result flag". The clearing of the result flag output from the meter, stops marking of ECN bit by the Marker.

15. References

15.1. Normative References

[1] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[2] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

[5] Heinanen, J., Baker, F., Weiss, W., and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.

[6] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.

[7] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168,

September 2001.

[8] Davie, B., Charny, A., Bennet, J., Benson, K., Le Boudec, J., Courtney, W., Davari, S., Firoiu, V., and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, March 2002.

[9] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

15.2. Informative References

[10] Spring, N., Wetherall, D., and D. Ely, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces", RFC 3540, June 2003.

[11] Alexander, C., "Real-time ECN Use Cases", draft-alexander-rtecn-use-cases-00 , July 2005.

[12] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", draft-ietf-tsvwg-diffserv-service-classes-01 , July 2005.

[13] Dudley, S., "Simulation of RT-ECN based Admission Control and Preemption", draft-dudley-tsvwg-rtecn-simulation-00 , July 2005.

[14] Babiarz, J., Dudley, S., and K. Chan, "Discussion of Congestion Marking with RT-ECN", draft-babiarz-rtecn-marking-00 .

[15] Alexander, C. and J. Babiarz, "RTP Payload Format for ECN Probing", draft-alexander-rtp-payload-for-ecn-probing-01 , July 2005.

[16] Matsumoto, M. and T. Nishimura, "Mersenne Twister MT 19937", http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html http://en.wikipedia.org/wiki/Mersenne_twister, March 2004.

[17] Floyd, S., "Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field", draft-floyd-ecn-alternates-01 , July 2005.

[18] Alexander, C. and J. Rutledge, "A Congestion Status Precondition for the Session Initiation Protocol (SIP)", draft-alexander-congestion-status-preconditions-00 , July 2005.

Authors' Addresses

Jozef Z. Babiarz
Nortel Networks
3500 Carling Avenue
Ottawa, Ont K2H 8E9
Canada

Phone: +1-613-763-6098
Fax: +1-613-768-2231
Email: babiarz@nortel.com

Kwok Ho Chan
Nortel Networks
600 Technology Park Drive
Billerica, MA 01821
US

Phone: +1-978-288-8175
Fax: +1-978-288-4690
Email: khchan@nortel.com

Victor Firoiu
BAE Systems
6 New England Executive Park
Burlington, MA 01803
US

Phone:
Fax:
Email: victor.firoiu@baesystems.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.