International Journal of Scientific Research in Computer Science and Engineering
Review Paper Vol-1, Issue-3 ISSN: 2320–7639

Packet Mark Methodology for Reduce the Congestion

V.Ganesh Babu and K.Saraswathi
Department of CS, Government College for Women, Maddur, Mandya Dt Karnataka, INDIA
Department of CS, Nehru Memorial College, Puthanampatti, Trichy Dt. Tamil Nadu, INDIA

Available online at www.isroset.org

Received: 17 March 2013 Revised: 22 March 2013 Accepted: 16 May 2013 Published: 30 June 2013

Abstract-Congestion is one of the major problems of a communication network in routing. The function of a routing is to guide packets through the communication network to their correct destinations. Optimal Routing has been widely studied for interconnection networks. This paper provides two methodologies ECN and QBER for reduce the congestion.

Keywords- Routing, congestion, ECN, QBER

1. INTRODUCTION

The first methodology Explicit Congestion Notification (ECN) is an extension to the Internet protocol and to the Transmission Control Protocol and is defined in RFC3168(2001). ECN allows end-to-end notification of network congestion without dropping packets. ECN is an optional feature that is only used when both endpoints support it and are willing to use it. It is only effective when supported by the underlying network. Conventionally, TCP/IP networks signal congestion by dropping packets. When ECN is successfully negotiated, an ECN-aware router may set a mark in the IP header instead of dropping a packet in order to signal impending congestion. The receiver of the packet echoes the congestion indication to the sender, which reduces its transmission rate as though it detected a dropped packet.

In the second methodology QBER we have to maintain queue corresponding to the router capacity. This queue is maintained before each router. Depending on the current queue size the router identifies the congestion. The router sends the signal to sender to stop its tranmission whenever the router receives the packet with the current queue size is QS-1.

2. ECN OPERATION

2.1 Operation of ECN with IP

ECN uses the two least significant (right-most) bits of the Diffser field in the IPV4 or IPV6 header to encode four different codepoints:

- 00: Non ECN-Capable Transport — Non-ECT
- 10: ECN Capable Transport — ECT(0)
- 01: ECN Capable Transport — ECT(1)
- 11: Congestion Encountered — CE

When both endpoints support ECN they mark their packets with ECT(0) or ECT(1). If the packet traverses an active queue management(AQM) queue (e.g. a queue that uses random early detection (RED)) that is experiencing congestion and the corresponding router supports ECN, it may change the codepoint to CE instead of dropping the packet. This act is referred to as “marking” and its purpose is to inform the receiving endpoint of impending congestion. At the receiving endpoint, this congestion indication is handled by the upper layer protocol (transport layer protocol) and needs to be echoed back to the transmitting node in order to signal it to reduce its transmission rate.

2.2 Operation of ECN with TCP

TCP supports ECN using three flags in the TCP header. The first one, the *Nonce Sum* (NS), is used to protect against accidental or malicious concealment of marked packets from the TCP sender.[4]. The other two bits are used to echo back the congestion indication (i.e. signal the sender to reduce the amount of information it sends) and to acknowledge that the congestion-indication echoing was received. These are the *ECN-Echo* (ECE) and *Congestion Window Reduced* (CWR) bits .Use of ECN on a TCP connection is optional; for ECN to be used, it must be negotiated at connection establishment by including suitable options in the SYN and SYN-ACK segments. When ECN has been negotiated on a TCP connection, the sender indicates that IP packets that carry TCP segments of that connection are carrying traffic from an ECN Capable Transport by marking them with an ECT code point. This allows intermediate routers that support ECN to mark those IP packets with the CE code point instead of dropping them in order to signal impending congestion.

Upon receiving an IP packet with the *Congestion Experienced* codepoint, the TCP receiver echoes back this congestion indication using the ECE flag in the TCP header. When an endpoint receives a TCP segment with the ECE bit it reduces its congestion window as for a packet drop. It then acknowledges the congestion indication by sending a segment with the CWR bit set. A node keeps transmitting TCP segments with the ECE bit set until it receives a segment with the CWR bit set.

2.3 Operation of ECN with TCP and other transport protocols

TCP does not perform congestion control on control packets (pure ACKs, SYN, FIN segments). So control packets are usually not marked as ECN-capable. A recent

Corresponding Author: *V.Ganesh Babu*

proposal [5] suggests marking SYN-ACK packets as ECN-capable. This improvement, known as ECN+, has been shown to provide dramatic improvements to performance of short-lived TCP connections.[6] ECN is also defined for other transport-layer protocols that perform congestion control, notably DCCP and SCTP. The general principle is similar to TCP, although the details of the on-the-wire encoding differ.It should in principle be possible to use ECN with protocols layered above UDP. However, UDP requires that congestion control be performed by the application, and current networking APIs do not give access to the ECN bits.

2.4 Effects on performance

ECN reduces the number of packets dropped by a TCP connection, which, by avoiding a retransmission, reduces latency and especially jitter. This effect is most drastic when the TCP connection has a single outstanding segment,[7] when it is able to avoid an RTO timeout; this is often the case for interactive connections (such as remote logins) and transactional protocols (such as HTTP requests, the conversational phase of SMTP, or SQL requests).Effects of ECN on bulk throughput are less clear [8] because modern TCP implementations are fairly good at resending dropped segments in a timely manner when the sender's window is large.Use of ECN has been found to be detrimental to performance on highly congested networks when using AQM algorithms that never drop packets.[6] Modern AQM implementations avoid this pitfall by dropping rather than marking packets at very high load.

2.5 ECN support in IP by routers

Since ECN marking in routers is dependent on some form of active queue management routers must be configured with a suitable queue discipline in order to perform ECN marking. Cisco IOS routers perform ECN marking if configured with the WRED queuing discipline since version 12.2(8)T.Linux routers perform ECN marking if configured with one of the RED or GRED queue disciplines with an explicit *ecn* parameter, by using the sfb discipline, or by using the CoDel Fair Queueing (fq_codel) discipline.Modern BSD implementations, such as FreeBSD, NetBSD and OpenBSD, have support for ECN marking in the ALTQ queueing implementation for a number of queuing disciplines, notably RED and Blue.

2.5 Alternate Semantics for the ECN Field

In ECN how routers know which ECN semantics to use with which packets. The end host sets the codepoint in the diffserv field to indicate to routers that alternate semantics to the ECN field are being used. Routers that understand this diffserv codepoint would know to use the alternate semantics for interpreting and setting the ECN field. Old ECN-capable routers that do not understand this diffserv codepoint would use the default ECN semantics in interpreting and setting the ECN field. In general, the diffserv codepoints are used to signal the per-hop behavior at router queues. One possibility would be to use one diffserv codepoint to signal a per-hop behavior with the default ECN semantics, and a separate diffserv codepoint to signal a similar per-hop behavior with the alternate ECN semantics. Another possibility would be to use a diffserv codepoint to signal the use of best-effort per-hop queueing and scheduling behavior, but with alternate ECN semantics.

2.6 Using the Diffserv Field for Signaling

There are two ways to use the diffserv field to signal the use of alternate ECN semantics. One way is to use an existing diffserv codepoint, and to modify the current definition of that codepoint, through approved IETF processes, to specify the use of alternate ECN semantics with that codepoint. A second way is to define a new diffserv codepoint, and to specify the use of alternate ECN semantics with that codepoint. We note that the first of these two mechanisms raises the possibility that some routers along the path will understand the diffserv codepoint but will use the default ECN semantics with this diffserv codepoint, or won't use ECN at all, and that other routers will use the alternate ECN semantics with this diffserv codepoint

3. Evaluation of the Alternate ECN Semantics

3.1 Verification of Feedback from the Router

In the default ECN semantics, two of the four ECN codepoints are used for ECN-Capable(0) and ECN-Capable(1). The use of two codepoints for ECN-Capable, instead of one, permits the data sender to verify the receiver's reports that packets were actually received unmarked at the receiver. In particular, the sender can specify that the receiver report to the sender whether each unmarked packet was received ECN-Capable(0) or ECN-Capable(1), as discussed in RFC 3540. This use of ECN-Capable(0) and ECN-Capable(1) is independent of the semantics of the other ECN codepoints, and could be used, if desired, with alternate semantics for the other codepoints.

If alternate semantics for the ECN codepoint don't include the use of two separate codepoints to indicate ECN-Capable, then the connections using those semantics have lost the ability to verify that the data receiver is accurately reporting the received ECN codepoint to the data sender. In this case, it might be necessary for the alternate-ECN framework to include alternate mechanisms for ensuring that the data receiver is reporting feedback appropriately to the sender. As one possibility, policers could be used in routers to ensure that end nodes are responding appropriately to marked packets.

3.2 Coexistence with Competing Traffic

If the traffic using the alternate ECN semantics is best-effort traffic, then it is subject to the general requirement of fair competition with TCP and other traffic along the path [RFC2914].If the traffic using the alternate ECN semantics is diffserv traffic, then the requirements are governed by the overall guidelines for that class of diffserv traffic.

3.3 Proposals for Alternate ECN with Edge-to-Edge Semantics

RFC 3168 specifies the use of the default ECN semantics by an end-to-end transport protocol, with the requirement that "upon the receipt by an ECN-Capable transport of a single CE packet, the congestion control algorithms followed at the end-systems MUST be essentially the same as the congestion control response to a *single* dropped packet"(RFC 3168).In contrast, some of the proposals for alternate ECN semantics are for ECN used in an edge-to-edge context between gateways at the edge of a network region, e.g., [BESFC06]. When alternate ECN is defined with edge-to-edge semantics, this definition needs to ensure that the edge-to-edge semantics do not conflict with a connection using other ECN semantics end-to-end. One way to avoid conflict would be for the edge-to-edge ECN proposal to include some mechanism to ensure that the edge-to-edge ECN is not used for connections that are using other ECN semantics (standard or otherwise) end-to-end. Alternately, the edge-to-edge semantics could be defined so that they do not conflict with a connection using other ECN semantics end-to-end.

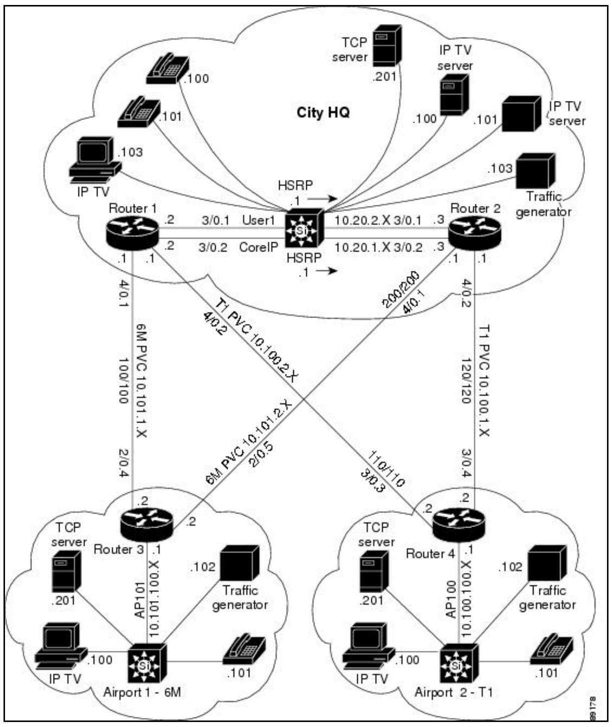
3.4 Robust ECN Signaling

The correct operation of ECN requires the cooperation of the receiver to return Congestion Experienced signals to the sender, but the protocol lacks a mechanism to enforce this cooperation. This raises the possibility that an unscrupulous or poorly implemented receiver could always clear ECN-Echo and simply not return congestion signals to the sender. This would give the receiver a performance advantage at the expense of competing connections that behave properly. More generally, any device along the path (NAT box, firewall, QOS bandwidth shapers, and so forth) could remove congestion marks with impunity. The above behaviors may or may not constitute a threat to the operation of congestion control in the Internet. However, given the central role of congestion control, it is prudent to design the ECN signaling loop to be robust against as many threats as possible. In this way, ECN can provide a clear incentive for improvement over the prior state-of-the-art without potential incentives for abuse. The ECN-nonce is a simple, efficient mechanism to eliminate the potential abuse of ECN. The ECN-nonce enables the sender to verify the correct behavior of the ECN receiver and that there is no other interference that conceals marked (or dropped) packets in the signaling path.The ECN- nonce protects against both implementation errors and deliberate abuse.
The ECN-nonce:

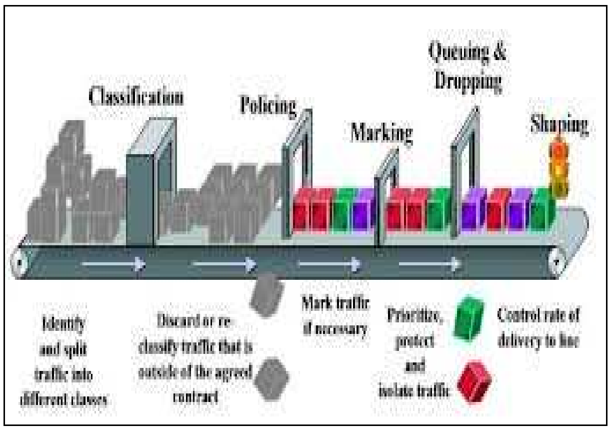
- catches a misbehaving receiver with a high probability, and never implicates an innocent receiver.
- does not change other aspects of ECN, nor does it reduce the benefits of ECN for behaving receivers.
- is cheap in both per-packet overhead (one TCP header flag) and processing requirements.
- is simple and, to the best of our knowledge, not prone to other attacks.

We also note that use of the ECN-nonce has two additional benefits, even when only drop-tail routers are used. First, packet drops cannot be concealed from the sender. Second, it prevents optimistic acknowledgements [Savage], in which TCP segments are acknowledged before they have been received. These benefits also serve to increase the robustness of congestion control from attacks.

4. FIGURES / CAPTIONS



a) Network with congestion


Classification
Policing
Marking
Queuing & Dropping
Shaping
Identify and split traffic into different classes
Discard or re-classify traffic that is outside of the agreed contract
Mark traffic if necessary
Prioritize, protect and isolate traffic
Control rate of delivery to line

b) Network without congestion in ECN

5 .QBER

QBER is Queue Before Each Router. In this methodology we have to maintain one queue before each router in the network. The sender transfers all the packets to the destination through multiple paths. Here all the packets are stored in the queue before they reach router. All the routers receive the packet one by one from their corresponding queue with current queue size(QS). So every router receive the packet with current queue size(QS).The router knows the current queue size(QS) whenever it receives every packet .The router easily identifies the congestion whenever it receives the packet with current queue size is QS-1.Then the router gives the signal to the sender to stop its transmission. Through this methodology the router identifies the congestion before it arrives and there is no loss of packet.

6. CONCLUSION

This paper is a study to overcome the problem of congestion using ECN and QBER. This will distribute the traffic of overloaded link to other preferred links. Hence the throughput of the network will be improved and the problem of congestion will be reduced.

REFERENCES

[1] Measuring the State of ECN Readiness in Servers, Clients, and Routers. Steven Bauer, Robert Beverly, and Arthur Berger. Internet Measurement Conference 2011.

[2] Measuring Interactions Between Transport Protocols and Middleboxes. Alberto Medina, Mark Allman, and Sally Floyd. Internet Measurement Conference 2004.

[3] http://www.icir.org/tbit/ecn-tbit.html

[4] RFC 3540 – Robus Explicit Congestion Notification.

[5] RFC 5562 - Adding Explicit Congestion Notification Capability to TCP's SYN/ACK Packets.

[6] Aleksandar Kuzmanovic. The power of explicit congestion notification. In Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications. 2005.

[7] Jamal Hadi Salim and Uvaiz Ahmed. Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks. RFC 2884. July 2000

[8] [8] Marek Malowidzki, Simulation-based Study of ECN Performance in RED Networks, In Proc. SPECTS'03. 2003.

[9] "New Networking Features in Windows Server 2008 and Windows Vista". http://technet.microsoft.com/en-us/library/bb726965.aspx.

[10] "ECN (Explicit Congestion Notification) in TCP/IP". http://www.icir.org/floyd/ecn.html#implementations.

[11] "/proc/sys/net/ipv4/* Variables:". http://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt.

[12] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC3168, September 2001.

[13] Babiarz, J., Chan, K., and V. Firoiu, "Congestion Notification Process for Real-Time Traffic", Work in Progress, July 2005.

[14] Briscoe, B., et al., "An edge-to-edge Deployment Model for Pre-Congestion Notification: Admission Control over a DiffServ Region", Work in Progress, June 2006.

[15] ECN Web Page, URL <www.icir.org/floyd/ecn.html>.

[16] S. Floyd, M. Allman, A. Jain, and P. Sarolahti, "Quick- Start for TCP and IP", Work in Progress, October 2006.

[17] Allman, M., Paxson, V., and W. Stevens, "TCP Congestion Control", RFC 2581, April 1999.

[18] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[19] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M Zhang, L., and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[20] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003.

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

[22] Floyd, S. and J. Kempf, "IAB Concerns Regardin Congestion Control for Voice Traffic in the Internet", RFC 3714, March 2004.

[23] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[24] Floyd, S. and E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID TCP-like Congestion Control", RFC 4341, March 2006.

[25] Y. Xia, L. Subramanian, I. Stoica, and S. Kalyanaraman, One More Bit Is Enough, SIGCOMM 2005, September 2005.