Packet Mark Methodology for Reduce the Congestion in the Network by using the Code Field in the TCP/IP header

K.Saraswathi[1]

Assistant Professor, Department of Computer Science, Nehru Memorial College, Puthanampatti(PO),Musiri(Tk), Trichy(Dt)-621 007.

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. Every router receives the packet with current queue size (CQS).The packet is marked by the router whenever there is congestion in the network. This paper provides one methodology PCN for reduce the congestion.

Keywords— Routing, Congestion, PCN

I.INTRODUCTION

The methodology Packet Congestion Notification (PCN) is an extension to the Internet protocol and to the Transmission Control Protocol and is defined in RFC3168(2001). PCN allows end-to-end notification of network congestion without dropping packets. PCN 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 PCN is successfully negotiated, an PCN-aware router may set a mark in the IP header instead of dropping a packet in order to signal 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.

II. PCN OPERATIONS

A.OPERATION OF PCN WITH IP

PCN 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 PCN-Capable Transport — Non-PCT
- 10: PCN Capable Transport — PCT(0)
- 01: PCN Capable Transport — PCT(1)
- 11: Congestion Encountered — CE

When both endpoints support PCN they mark their packets with PCT(0) or PCT(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 PCN, 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.

B.OPERATION OF PCN WITH TCP

TCP supports PCN 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 P*CN-Echo* (PCE) and *Congestion Window Reduced* (CWR) bits .Use of PCN on a TCP connection is optional. It must be negotiated at connection establishment by including suitable options in the SYN and SYN-ACK segments. When PCN has been negotiated on a TCP connection, the sender indicates that IP packets that carry TCP segments of that connection are carrying traffic from PCN Capable Transport by marking them with an PCT code point. This allows intermediate routers that support PCN to mark those IP packets with the CE code point instead of dropping them in order to signal congestion.

Upon receiving an IP packet with the *Congestion Experienced* codepoint, the TCP receiver echoes back this congestion indication using the PCE flag in the TCP header. When an endpoint receives a TCP segment with the PCE 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 PCE bit set until it receives a segment with the CWR bit.

C.OPERATION OF PCN WITH OTHER PROTOCOLS

TCP does not perform congestion control on control packets (pure ACKs, SYN, FIN segments). So control packets are usually not marked as PCN-capable. A recent proposal suggests marking SYN-ACK packets as PCN-capable. This improvement, known as PCN+, has been shown to provide dramatic improvements to performance of short-lived TCP connections.. PCN 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 PCN with protocols layered above UDP. However, UDP requires that congestion control be performed by the application, and current networking APIs does not give access to the PCN bits.

III. EFFECTS ON PERFORMANCE

PCN 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, 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 PCN on bulk throughput are less clear because modern TCP implementations are fairly good at resending dropped segments in a timely manner when the sender's window is large. Use of PCN has been found to be detrimental to performance on highly congested networks when using AQM algorithms that never drop packets. Modern AQM implementations avoid this pitfall by dropping rather than marking packets at very high load.

A. ECN SUPPORT IN IP BY ROUTERS

Since PCN 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 PCN marking. Cisco IOS routers perform PCN marking if configured with the WRED queuing discipline.Linux routers perform PCN 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 PCN marking in the ALTQ queueing implementation for a number of queuing disciplines, notably RED and Blue.

B. ALTERNATE SEMANTICS FOR THE PCN FIELD

In PCN how routers know which PCN 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 PCN field are being used. Routers that understand this diffserv codepoint would know to use the alternate semantics for interpreting and setting the PCN field. Old PCN-capable routers that do not understand this diffserv codepoint would use the default PCN semantics in interpreting and setting the PCN 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 PCN

Semantics and a separate diffserv codepoint to signal a similar per-hop behavior with the alternate PCN 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 PCN semantics.

C. USING THE DIFFSERV FIELD FOR SIGNALING

There are two ways to use the diffserv field to signal the use of alternate PCN 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 PCN semantics with that codepoint. A second way is to define a new diffserv codepoint, and to specify the use of alternate PCN 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 PCN semantics with this diffserv codepoint, or won't use PCN at all, and that other routers will use the alternate PCN semantics with this diffserv codepoint

IV. EVALUATION OF THE ALTERNATE PCN SEMANTICS

A. VERIFICATION OF FEEDBACK FROM THE ROUTER

In the default PCN semantics, two of the four PCN codepoints are used for PCN-Capable (0) and PCN-Capable (1). The use of two codepoints for PCN-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 PCN-Capable(0) or PCN-Capable(1), as discussed in RFC 3540. This use of PCN-Capable (0) and PCN-Capable(1) is independent of the semantics of the other PCN codepoints, and could be used, if desired, with alternate semantics for the other codepoints. If alternate semantics for the PCN codepoint don't include the use of two separate codepoints to indicate PCN-Capable, then the connections using those semantics have lost the ability to verify that the data receiver is accurately reporting the received PCN codepoint to the data sender. In this case, it might be

necessary for the alternate-PCN 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.

B. COEXISTENCE WITH COMPETING TRAFFIC

If the traffic using the alternate PCN 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 PCN semantics is diffserv traffic, then the requirements are governed by the overall guidelines for that class of diffserv traffic.

C. ALTERNATE PCN WITH EDGE-TO-EDGE SEMANTICS

RFC 3168 specifies the use of the default PCN semantics by an end-to-end transport protocol, with the requirement that "upon the receipt by an PCN-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 PCN semantics are for PCN used in an edge-to-edge context between gateways at the edge of a network region, e.g., [BESFC06].

When alternate PCN 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 PCN semantics end-to-end. One way to avoid conflict would be for the edge-to-edge PCN proposal to include some mechanism to ensure that the edge-to-edge PCN is not used for connections that are using other PCN 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 PCN semantics end-to-end.

D. ROBUST PCN SIGNALLING

The correct operation of PCN 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 PCN-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 PCN signaling loop to be robust against as many threats as possible. In this way, PCN can provide a clear incentive for improvement over the prior state-of-the-art without potential incentives for abuse. The PCN-nonce is a simple, efficient mechanism to eliminate the potential abuse of PCN. The PCN-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 PCN- nonce protects against both implementation errors and deliberate abuse.

The PCN-nonce:

- catches a misbehaving receiver with a high probability, and never implicates an innocent receiver.
- does not change other aspects of PCN, nor does it reduce the benefits of PCN 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 PCN-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.

V. FIGURES/CAPTIONS

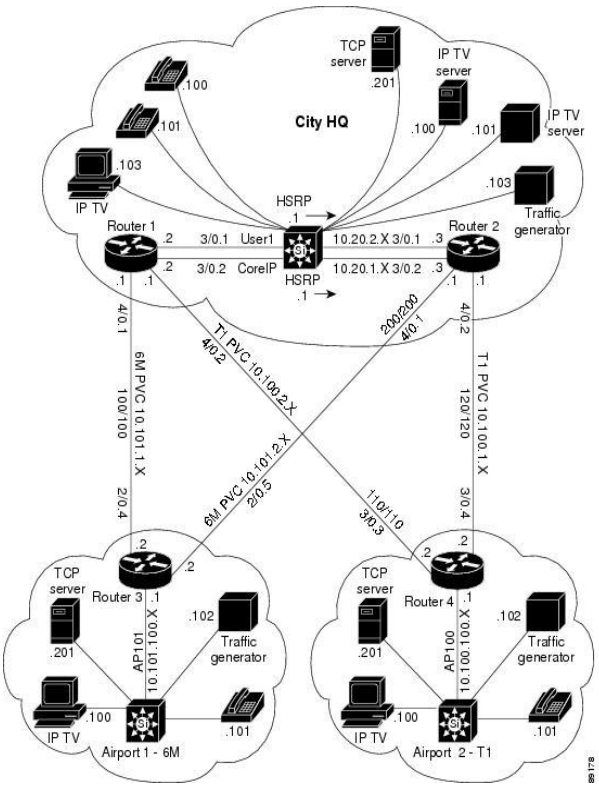

Fig. 1 Network with congestion

VI.CONCLUSION

This paper is a study to overcome the problem of congestion using PCN. Through this methodology the router identifies the congestion before it arrives and there is no loss of packet. Hence the throughput of the network will be improved and the problem of congestion will be reduced.

VII.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 Middle boxes. Alberto Medina, Mark Allman, and Sally Floyd. Internet Measurement Conference 2004.

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

[4] *RFC 3540 - Robust Explicit Congestion Notification.*
RFC 5562 - Adding Explicit Congestion Notification Capability to TCP's SYN/ACK Packets.

[5] Aleksandar Kuzmanovic. The power of explicit congestion notification. In Proceedings *of the 2005 conference on Applications, technologies,*

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

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

[13] 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.