

Explicit Congestion Notification (ECN)

The Internet's end-to-end rate control schemes (notably the TransmissionControlProtocol (TCP)) traditionally have to rely on packet loss as the prime indicator of congestion (queueing delay is also used for congestion feedback, although mostly implicitly, except in newer mechanisms such as TCP FAST).

Both loss and delay are *implicit* signals of congestion. The alternative is to send *explicit* congestion signals.

ICMP Source Quench

Such a signal was indeed part of the original IP design, in the form of the "Source Quench" ICMP (Internet Control Message Protocol) message. A router experiencing congestion could send ICMP Source Quench messages towards a source, in order to tell that source to send more slowly. This method was deprecated by RFC1995 in June 1982. Oops, I meant by RFC1812 ("Requirements for IP Version 4 Routers") in June 1995. The biggest problem with ICMP Source Quench are:

- that the mechanism causes more traffic to be generated in a situation of congestion (although in the other direction)
- that, when ICMP Source Quench messages are lost, it fails to slow down the sender.

The ECN Proposal

The new ECN mechanism consists of two components:

- Two new "ECN" bits in the former TOS field of the IP header:
 - The "ECN-Capable Transport" (ECT) bit must only be set for packets controlled by ECN-aware transports
 - The "Congestion Experienced" (CE) bit can be set by a router if
 - the router has detected congestion on the outgoing link
 - and the ECT bit is set.
- Transport-specific protocol extensions which communicate the ECN signal back from the receiver to the sender. For TransmissionControlProtocol, this takes the form of two new flags in the TCP header, `ECN-Echo` (ECE) and `Congestion Window Reduced` (CWR). A similar mechanism has been included in SCTP.

ECN works as follows. When a transport supports ECN, it sends IP packets with ECT (ECN-Capable Transport) set. Then, when there is congestion, a router will set the CE (Congestion Experienced) bit in some of these packets. The receiver notices this, and sends a signal back to the sender (by setting the ECE flag). The sender then reduces its sending rate, as if it had detected the loss of a single packet, and sets the CWR flag so as to inform the receiver of this action.

(Note that the two-bit ECN field in the IP header has been redefined in the current ECN RFC (RFC3168), so that "ECT" and "CE" are no longer actual bits. But the old definition is somewhat easier to understand. If you want to know how these "conceptual" bits are encoded, please check out RFC 3168.)

Benefits of ECN

ECN provides two significant benefits:

- ECN-aware transports can properly adapt their rates to congestion without requiring packet loss
- Congestion feedback can be quicker with ECN, because detecting a dropped packet requires a timeout.

For more detailed information, see the Internet-Draft *The Benefits of using Explicit Congestion Notification (ECN)* mentioned in the References below.

Deployment Issues with ECN

ECN requires AQM, which isn't widely deployed

ECN requires routers to use an [Active Queue Management](#) (AQM) mechanism such as [Random Early Detection](#) (RED). In addition, routers have to be able to mark eligible (ECT) packets with the CE bit when the AQM mechanism notices congestion. [RED](#) is widely implemented on routers today, although it is rarely activated in actual networks.

ECN must be added to routers' forwarding paths

The capability to ECN-mark packets can be added to CPU- or Network-Processor-based routing platforms relatively easily - Cisco's CPU-based routers such as the 7200/7500 routers support this with newer software, for example, but if queueing/forwarding is performed by specialized hardware (ASICs), this function has to be designed into the hardware from the start. Therefore, most of today's high-speed routers cannot easily support ECN to my knowledge.

ECN "Blackholing"

Another issue is that attempts to use ECN can cause issues with certain "middlebox" devices such as firewalls or load balancers, which break connectivity when unexpected TCP flags (or, more rarely, unexpected IP TOS values) are encountered. The original ECN RFC (RFC 2481) didn't handle this gracefully, so activating ECN on hosts that implement this version caused much frustration because of "hanging" connections. RFC 3168 proposes a mechanism to deal with ECN-unfriendly networks, but that hasn't been widely implemented yet. In particular, the Linux ECN implementation doesn't seem to implement it as of November 2007 (Linux 2.6.23).

See [Floyd's ECN Problems page](#) for more.

How to Activate ECN

On Linux hosts

ECN for both IPv4 and IPv6 is controlled by a `sysctl` parameter `net.ipv4.tcp_ecn`. According to [ip-sysctl.txt](#), it may have one of three values: 0 means "never do ECN", 1 means "actively try to negotiate ECN", 2 means "do ECN when asked for". To activate on a running system:

```
echo 1 > /proc/sys/net/ipv4/tcp_ecn
```

To make persistent, add the following line to `/etc/sysctl.conf`:

```
net.ipv4.tcp_ecn=1
```

On Mac OS X

In a [message from 23 March 2015 to the iccr mailing list](#), Stuart Cheshire explains how to activate ECN on Mac OS X. The following commands activate it on the running system:

```
sysctl -w net.inet.tcp.ecn_negotiate_in=1
sysctl -w net.inet.tcp.ecn_initiate_out=1
```

To make it persistent, add the following to `/etc/sysctl.conf`:

```
net.inet.tcp.ecn_initiate_out=1
net.inet.tcp.ecn_negotiate_in=1
```

Suggested Improvements

For situations where ECN is not explicit enough, [RFC 7514 \(Really Explicit Congestion Notification \(RECN\)\)](#) suggests a protocol where senders can be told to back off in increasingly explicit terms.

References

- [The Addition of Explicit Congestion Notification \(ECN\) to IP](#), RFC 3168, K. Ramakrishnan, S. Floyd, D. Black, September 2001
- [Robust Explicit Congestion Notification \(ECN\) Signaling with Nonces](#), RFC 3540, N. Spring, D. Wetherall, D. Ely, June 2003 (reclassified as HISTORIC in November 2017)
- [ECN \(Explicit Congestion Notification\) in TCP/IP](#), Web page by Sally Floyd
- [RFC 8087, The Benefits of Using Explicit Congestion Notification \(ECN\)](#), G. Fairhurst, M. Welzl, March 2017
- [Really Explicit Congestion Notification \(RECN\)](#), RFC 7514, M. Luckie, 1 April 2015
- [RFC 7560, Problem Statement and Requirements for Increased Accuracy in Explicit Congestion Notification \(ECN\) Feedback](#), M. Kühlewind, Ed., R. Scheffenegger, B. Briscoe, August 2015
- [RFC 8311, Relaxing Restrictions on Explicit Congestion Notification \(ECN\) Experimentation](#), David Black, January 2018
- [RFC 8511, TCP Alternative Backoff with ECN \(ABE\)](#), N. Khademi, M. Welzl, G. Armitage, G. Fairhurst, December 2018
- [draft-ietf-tcpm-accurate-ecn-06, More Accurate ECN Feedback in TCP](#), Bob Briscoe, Mirja Kühlewind, Richard Scheffenegger, March 2018
- [draft-ietf-tsvwg-ecn-l4s-id-04, Identifying Modified Explicit Congestion Notification \(ECN\) Semantics for Ultra-Low Queuing Delay \(L4S\)](#), Koen De Schepper, Bob Briscoe, November 2018
- [draft-eastlake-sfc-nsh-ecn-support-02, Explicit Congestion Notification \(ECN\) and Congestion Feedback Using the Network Service Header \(NSH\)](#), Donald E. Eastlake, Bob Briscoe, Andrew G. Malis, October 2018