Stream: Internet Engineering Task Force (IETF)
RFC: 9599
BCP: 89
Updates: 3819
Category: Best Current Practice
Published: July 2024
ISSN: 2070-1721
Authors: B. Briscoe, *Independent*; J. Kaippallimalil, *Futurewei*

RFC 9599
Guidelines for Adding Congestion Notification to Protocols That Encapsulate IP

Abstract

The purpose of this document is to guide the design of congestion notification in any lower-layer or tunnelling protocol that encapsulates IP. The aim is for explicit congestion signals to propagate consistently from lower-layer protocols into IP. Then, the IP internetwork layer can act as a portability layer to carry congestion notification from non-IP-aware congested nodes up to the transport layer (L4). Specifications that follow these guidelines, whether produced by the IETF or other standards bodies, should assure interworking among IP-layer and lower-layer congestion notification mechanisms. This document is included in BCP 89 and updates the single paragraph of advice to subnetwork designers about Explicit Congestion Notification (ECN) in Section 13 of RFC 3819 by replacing it with a reference to this document.

Status of This Memo

This memo documents an Internet Best Current Practice.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on BCPs is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9599.

Copyright Notice

Table of Contents

1. Introduction

In certain networks, it might be possible for traffic to congest non-IP-aware nodes. In such networks, the benefits of Explicit Congestion Notification (ECN) described in [RFC8087] and summarized below can only be fully realized if support for congestion notification is added to the relevant subnetwork technology, as well as to IP. When a lower-layer buffer implicitly notifies congestion by dropping a packet, it obviously does not just drop at that layer; the packet disappears from all layers. In contrast, when active queue management (AQM) at a lower layer buffer explicitly notifies congestion by marking a frame header, the marking needs to be explicitly propagated up the layers. The same is true if AQM marks the outer header of a packet that encapsulates inner tunnelled headers. Forwarding ECN is not as straightforward as other headers because it has to be assumed ECN may be only partially deployed. If a lower-layer header that contains congestion indications is stripped off by a subnet egress that is not ECN-aware, or if the ultimate receiver or sender is not ECN-aware, congestion needs to be indicated by dropping the packet, not marking it.

The purpose of this document is to guide the addition of congestion notification to any subnet technology or tunnelling protocol so that lower-layer AQM algorithms can signal congestion explicitly and that signal will propagate consistently into encapsulated (higher-layer) headers. Otherwise, the signals will not reach their ultimate destination.

ECN is defined in the IP header (IPv4 and IPv6) [RFC3168] to allow a resource to notify the onset of queue buildup without having to drop packets by explicitly marking a proportion of packets with the congestion experienced (CE) codepoint.

Given a suitable marking scheme, ECN removes nearly all congestion loss and it cuts delays for two main reasons:

- It avoids the delay when recovering from congestion losses, which particularly benefits small flows or real-time flows, making their delivery time predictably short [RFC2884].
- As ECN is used more widely by end systems, it will gradually remove the need to configure a degree of delay into buffers before they start to notify congestion (the cause of bufferbloat). This is because drop involves a trade-off between sending a timely signal and trying to avoid impairment, whereas ECN is solely a signal and not an impairment, so there is no harm triggering it earlier.

Some lower-layer technologies (e.g., MPLS, Ethernet) are used to form subnetworks with IP-aware nodes only at the edges. These networks are often sized so that it is rare for interior queues to overflow. However, until recently, this was more due to the inability of TCP to saturate

the links. For many years, fixes such as window scaling [RFC7323] proved hard to deploy and the Reno variant of TCP remained in widespread use despite its inability to scale to high flow rates. However, now that modern operating systems are finally capable of saturating interior links, even the buffers of well-provisioned interior switches will need to signal episodes of queuing.

Propagation of ECN is defined for MPLS [RFC5129] and TRILL [RFC7780] [RFC9600], but it has yet to be defined for a number of other subnetwork technologies.

Similarly, ECN propagation is yet to be defined for many tunnelling protocols. [RFC6040] defines how ECN should be propagated for IP-in-IPv4 [RFC2003], IP-in-IPv6 [RFC2473], and IPsec [RFC4301] tunnels, but there are numerous other tunnelling protocols with a shim and/or a Layer 2 (L2) header between two IP headers (IPv4 or IPv6). Some address ECN propagation between the IP headers, but many do not. This document gives guidance on how to address ECN propagation for future tunnelling protocols, and a companion Standards Track specification [RFC9601] updates existing tunnelling protocols with a shim between IP headers that are under IETF change control and still widely used.

Incremental deployment is the most delicate aspect when adding support for ECN. The original ECN protocol in IP [RFC3168] was carefully designed so that a congested buffer would not mark a packet (rather than drop it) unless both source and destination hosts were ECN-capable. Otherwise, its congestion markings would never be detected and congestion would just build up further. However, to support congestion marking below the IP layer or within tunnels, it is not sufficient to only check that the two layer 4 transport endpoints support ECN; correct operation also depends on the decapsulator at each subnet or tunnel egress faithfully propagating congestion notification to the higher layer. Otherwise, a legacy decapsulator might silently fail to propagate any congestion signals from the outer header to the forwarded header. Then, the lost signals would never be detected and congestion would build up further. The guidelines given later require protocol designers to carefully consider incremental deployment and suggest various safe approaches for different circumstances.

Of course, the IETF does not have standards authority over every link-layer protocol; thus, this document gives guidelines for designing propagation of congestion notification across the interface between IP and protocols that may encapsulate IP (i.e., that can be layered beneath IP). Each lower-layer technology will exhibit different issues and compromises, so the IETF or the relevant standards body must be free to define the specifics of each lower-layer congestion notification scheme. Nonetheless, if the guidelines are followed, congestion notification should interwork between different technologies using IP in its role as a 'portability layer'.

Therefore, the capitalized terms '**SHOULD**' or '**SHOULD NOT**' are often used in preference to '**MUST**' or '**MUST NOT**' because it is difficult to know the compromises that will be necessary in each protocol design. If a particular protocol design chooses not to follow a '**SHOULD**' or '**SHOULD NOT**' given in the advice below, it **MUST** include a sound justification.

It has not been possible to give common guidelines for all lower-layer technologies because they do not all fit a common pattern. Instead, they have been divided into a few distinct modes of operation: feed-forward-and-up, feed-up-and-forward, feed-backward, and null mode. These modes are described in Section 3, and separate guidelines are given for each mode in subsequent sections.

1.1. Update to RFC 3819

This document updates the brief advice to subnetwork designers about ECN in Section 13 of [RFC3819] by adding this document (RFC 9599) as an informative reference and replacing the last two paragraphs with the following sentence:

> By following the guidelines in [RFC9599], subnetwork designers can enable a layer-2 protocol to participate in congestion control without dropping packets via propagation of Explicit Congestion Notification (ECN) [RFC3168] to receivers.

1.2. Scope

This document only concerns wire protocol processing of explicit notification of congestion. It makes no changes or recommendations concerning algorithms for congestion marking or congestion response because algorithm issues should be independent of the layer that the algorithm operates in.

The default ECN semantics are described in [RFC3168] and updated by [RFC8311]. Also, the guidelines for AQM designers [RFC7567] clarify the semantics of both drop and ECN signals from AQM algorithms. [RFC4774] is the appropriate best current practice specification of how algorithms with alternative semantics for the ECN field can be partitioned from Internet traffic that uses the default ECN semantics. There are two main examples for how alternative ECN semantics have been defined in practice:

- [RFC4774] suggests using the ECN field in combination with a Diffserv codepoint, such as in Pre-Congestion Notification (PCN) [RFC6660], Voice over 3G [UTRAN], or Voice over LTE (VoLTE) [LTE-RA].
- [RFC8311] suggests using the ECT(1) codepoint of the ECN field to indicate alternative semantics, such as for the experimental Low Latency, Low Loss, and Scalable throughput (L4S) service [RFC9331].

The aim is that the default rules for encapsulating and decapsulating the ECN field are sufficiently generic that tunnels and subnets will encapsulate and decapsulate packets without regard to how algorithms elsewhere are setting or interpreting the semantics of the ECN field. [RFC6040] updates [RFC4774] to allow alternative encapsulation and decapsulation behaviours to be defined for alternative ECN semantics. However, it reinforces the same point -- it is far preferable to try to fit within the common ECN encapsulation and decapsulation behaviours because expecting all lower-layer technologies and tunnels to be updated is likely to be completely impractical.

Alternative semantics for the ECN field can be defined to depend on the traffic class indicated by the Differentiated Services Code Point (DSCP). Therefore, correct propagation of congestion signals could depend on correct propagation of the DSCP between the layers and along the path. For instance, if the meaning of the ECN field depends on the DSCP (as in PCN or VoLTE) and the outer DSCP is stripped on decapsulation, as in the pipe model of [RFC2983], the special semantics of the ECN field would be lost. Similarly, if the DSCP is changed at the boundary between Diffserv domains, the special ECN semantics would also be lost. This is an important implication of the localized scope of most Diffserv arrangements. In this document, correct propagation of traffic class information is assumed while the meaning of 'correct' and how it is achieved is covered elsewhere (e.g., [RFC2983]) and is outside the scope of this document.

The guidelines in this document do ensure that common encapsulation and decapsulation rules are sufficiently generic to cover cases where ECT(1) is used instead of ECT(0) to identify alternative ECN semantics (as in L4S [RFC9331]) and where ECN-marking algorithms use ECT(1) to encode three severity levels into the ECN field (e.g., PCN [RFC6660]) rather than the default of two. All these different semantics for the ECN field work because it has been possible to define common default decapsulation rules that allow for all cases [RFC6040].

Note that the guidelines in this document do not necessarily require the subnet wire protocol to be changed to add support for congestion notification. For instance, the feed-up-and-forward mode (Section 3.2) and the null mode (Section 3.4) do not. Another way to add congestion notification without consuming header space in the subnet protocol might be to use a parallel control plane protocol.

This document focuses on the congestion notification interface between IP and lower-layer or tunnel protocols that can encapsulate IP, where the term 'IP' includes IPv4 or IPv6, unicast, multicast, or anycast. However, it is likely that the guidelines will also be useful when a lower-layer protocol or tunnel encapsulates itself, e.g., Ethernet Media Access Control (MAC) in MAC ([IEEE802.1Q]; previously 802.1ah), or when it encapsulates other protocols. In the feed-backward mode, propagation of congestion signals for multicast and anycast packets is out of scope (because the complexity would make it unlikely to be attempted).

2. Terminology

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Further terminology used within this document:

Protocol data unit (PDU): Information that is delivered as a unit among peer entities of a layered network consisting of protocol control information (typically a header) and possibly user data (payload) of that layer. The scope of this document includes Layer 2 and Layer 3

networks, where the PDU is respectively termed a frame or a packet (or a cell in ATM). PDU is a general term for any of these. This definition also includes a payload with a shim header lying somewhere between layer 2 and 3.

Transport: The end-to-end transmission control function, conventionally considered at layer 4 in the OSI reference model. Given the audience for this document will often use the word transport to mean low-level bit carriage, the term will be qualified whenever it is used, e.g., 'L4 transport'.

Encapsulator: The link or tunnel endpoint function that adds an outer header to a PDU (also termed the 'link ingress', the 'subnet ingress', the 'ingress tunnel endpoint', or just the 'ingress' where the context is clear).

Decapsulator: The link or tunnel endpoint function that removes an outer header from a PDU (also termed the 'link egress', the 'subnet egress', the 'egress tunnel endpoint', or just the 'egress' where the context is clear).

Incoming header: The header of an arriving PDU before encapsulation.

Outer header: The header added to encapsulate a PDU.

Inner header: The header encapsulated by the outer header.

Outgoing header: The header forwarded by the decapsulator.

CE: Congestion Experienced [RFC3168]

ECT: ECN-Capable (L4) Transport [RFC3168]

Not-ECT: Not ECN-Capable (L4) Transport [RFC3168]

Load Regulator: For each flow of PDUs, the transport function that is capable of controlling the data rate. Typically located at the data source, but in-path nodes can regulate load in some congestion control arrangements (e.g., admission control, policing nodes, or transport circuit-breakers [RFC8084]). Note that "a function capable of controlling the load" deliberately includes a transport that does not actually control the load responsively, but ideally it ought to (e.g., a sending application without congestion control that uses UDP).

ECN-PDU: A PDU at the IP layer or below with a capacity to signal congestion that is part of a congestion control feedback loop within which all the nodes necessary to propagate the signal back to the Load Regulator are capable of doing that propagation. An IP packet with a non-zero ECN field implies that the endpoints are ECN-capable, so this would be an ECN-PDU. However, ECN-PDU is intended to be a general term for a PDU at lower layers, as well as at the IP layer.

Not-ECN-PDU: A PDU at the IP layer or below that is part of a congestion control feedback loop that is not capable of propagating ECN signals back to the Load Regulator because at least one of the nodes necessary to propagate the signals is incapable of doing that propagation. Note that this definition is a property of the feedback loop, not necessarily of the PDU itself; certainly the PDU will self-describe the property in some protocols, but in others, the property might be carried in a separate control plane context (which is somehow bound to the PDU).

3. Modes of Operation

This section sets down the different modes by which congestion information is passed between the lower layer and the higher one. It acts as a reference framework for the subsequent sections that give normative guidelines for designers of congestion notification protocols, taking each mode in turn:

Feed-Forward-and-Up: Nodes feed forward congestion notification towards the egress within the lower layer, then up and along the layers towards the end-to-end destination at the transport layer. The following local optimization is possible:

Feed-Up-and-Forward: A lower-layer switch feeds up congestion notification directly into the higher layer (e.g., into the ECN field in the IP header), irrespective of whether the node is at the egress of a subnet.

Feed-Backward: Nodes feed back congestion signals towards the ingress of the lower layer and (optionally) attempt to control congestion within their own layer.

Null: Nodes cannot experience congestion at the lower layer except at the ingress nodes of the subnet (which are IP-aware or equivalently higher-layer-aware).

3.1. Feed-Forward-and-Up Mode

Like IP and MPLS, many subnet technologies are based on self-contained PDUs or frames sent unreliably. They provide no feedback channel at the subnetwork layer, instead relying on higher layers (e.g., TCP) to feed back loss signals.

In these cases, ECN may best be supported by standardising explicit notification of congestion into the lower-layer protocol that carries the data forwards. Then, a specification is needed for how the egress of the lower-layer subnet propagates this explicit signal into the forwarded upper-layer (IP) header. This signal continues forwards until it finally reaches the destination transport (at L4). Typically, the destination will feed this congestion notification back to the source transport using an end-to-end protocol (e.g., TCP). This is the arrangement that has already been used to add ECN to IP-in-IP tunnels [RFC6040], IP-in-MPLS, and MPLS-in-MPLS [RFC5129].

This mode is illustrated in Figure 1. Along the middle of the figure, layers 2, 3, and 4 of the protocol stack are shown. One packet is shown along the bottom as it progresses across the network from source to destination, crossing two subnets connected by a router and crossing two switches on the path across each subnet. Congestion at the output of the first switch (shown as *) leads to a congestion marking in the L2 header (shown as C in the illustration of the packet). The chevrons show the progress of the resulting congestion indication. It is propagated from link to link across the subnet in the L2 header. Then, when the router removes the marked L2 header,

it propagates the marking up into the L3 (IP) header. The router forwards the marked L3 header into subnet B. The L2 protocol used in subnet B does not support congestion notification, but the signal proceeds across it in the L3 header.

Note that there is no implication that each 'C' marking is encoded the same; a different encoding might be used for the 'C' marking in each protocol.

Finally, for completeness, we show the L3 marking arriving at the destination, where the host transport protocol (e.g., TCP) feeds it back to the source in the L4 acknowledgement (the 'C' at L4 in the packet at the top of the diagram).

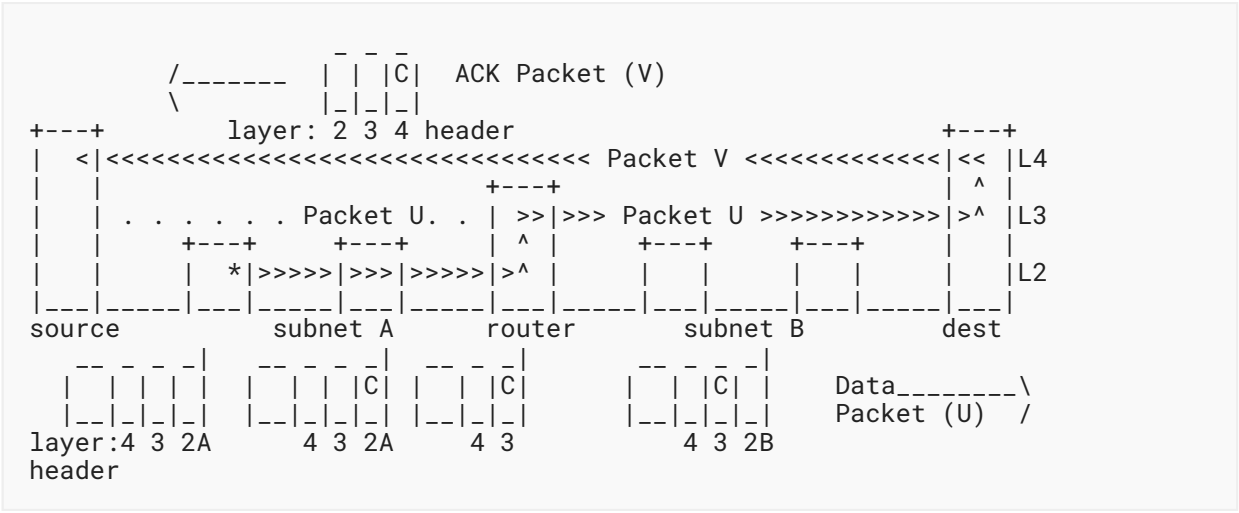

Figure 1: Feed-Forward-and-Up Mode

Of course, modern networks are rarely as simple as this textbook example, often involving multiple nested layers. For example, a Third Generation Partnership Project (3GPP) mobile network may have two IP-in-IP GTP [GTPv1] tunnels in series and an MPLS backhaul between the base station and the first router. Nonetheless, the example illustrates the general idea of feeding congestion notification forward then upward whenever a header is removed at the egress of a subnet.

Note that the Forward Explicit Congestion Notification (FECN) bit in Frame Relay [Buck00] and the Explicit Forward Congestion Indication (EFCI) [ITU-T.I.371] bit in ATM user data cells follow a feed-forward pattern. However, in ATM, this arrangement is only part of a feed-forward-and-backward pattern at the lower layer, not feed-forward-and-up out of the lower layer -- the intention was never to interface with IP-ECN at the subnet egress. To our knowledge, Frame Relay FECN is solely used by network operators to detect where they should provision more capacity.

3.2. Feed-Up-and-Forward Mode

Ethernet is particularly difficult to extend incrementally to support congestion notification. One way is to use so-called 'Layer 3 switches'. These are Ethernet switches that dig into the Ethernet payload to find an IP header and manipulate or act on certain IP fields (specifically Diffserv and ECN). For instance, in Data Center TCP [RFC8257], Layer 3 switches are configured to mark the ECN field of the IP header within the Ethernet payload when their output buffer becomes congested. With respect to switching, a Layer 3 switch acts solely on the addresses in the Ethernet header; it does not use IP addresses and it does not decrement the TTL field in the IP header.

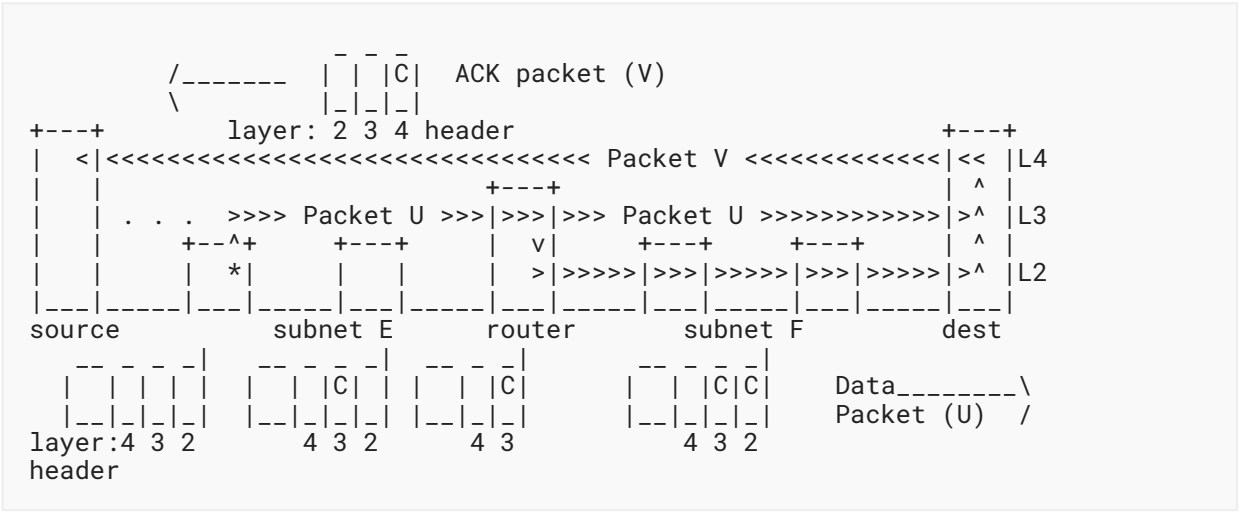

Figure 2: Feed-Up-and-Forward Mode

By comparing Figure 2 with Figure 1, it can be seen that subnet E (perhaps a subnet of Layer 3 Ethernet switches) works in feed-up-and-forward mode by notifying congestion directly into L3 at the point of congestion, even though the congested switch does not otherwise act at L3. In this example, the technology in subnet F (e.g., MPLS) does support ECN. So, when the router adds the Layer 2 header, it copies the ECN marking from L3 to L2 as well, as shown by the 'C's in both layers.

3.3. Feed-Backward Mode

In some Layer 2 technologies, congestion notification has been defined for use internally within the subnet with its own feedback and load regulation but the interface with IP for ECN has not been defined.

For instance, the relative rate mechanism was one of the more popular ways to manage traffic for the Available Bit Rate (ABR) service in ATM, and it tended to supersede earlier designs. In this approach, ATM switches send special resource management (RM) cells in both the forward and backward directions to control the ingress rate of user data into a virtual circuit. If a switch

buffer is approaching congestion or is congested, it sends an RM cell back towards the ingress with respectively the No Increase (NI) or Congestion Indication (CI) bit set in its message type field [ATM-TM-ABR]. The ingress then holds or decreases its sending bit rate accordingly.

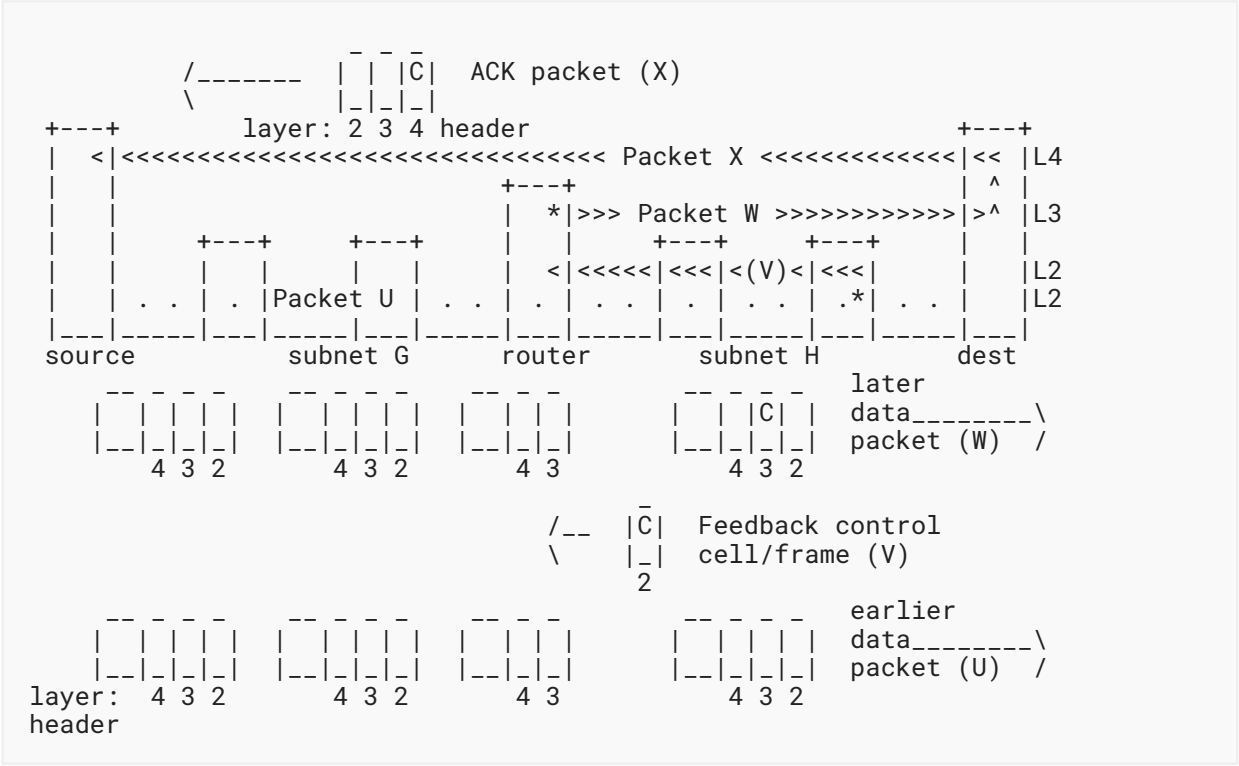

Figure 3: Feed-Backward Mode

ATM's feed-backward approach does not fit well when layered beneath IP's feed-forward approach unless the initial data source is the same node as the ATM ingress. Figure 3 shows the feed-backward approach being used in subnet H. If the final switch on the path is congested (*), it does not feed forward any congestion indications on the packet (U). Instead, it sends a control cell (V) back to the router at the ATM ingress.

However, the backward feedback does not reach the original data source directly because IP does not support backward feedback (and subnet G is independent of subnet H). Instead, the router in the middle throttles down its sending rate, but the original data sources don't reduce their rates. The resulting rate mismatch causes the middle router's buffer at layer 3 to back up until it becomes congested, which it signals forwards on later data packets at layer 3 (e.g., packet W). Note that the forward signal from the middle router is not triggered directly by the backward signal. Rather, it is triggered by congestion resulting from the middle router's mismatched rate response to the backward signal.

In response to this later forward signalling, end-to-end feedback at layer 4 finally completes the tortuous path of congestion indications back to the origin data source as before.

Quantized Congestion Notification (QCN) [IEEE802.1Q] would suffer from similar problems if extended to multiple subnets. However, QCN was clearly characterized as solely applicable to a single subnet from the start (see Section 6).

3.4. Null Mode

Link- and physical-layer resources are often 'non-blocking' by design. Congestion notification may be implemented in these cases, but it does not need to be deployed at the lower layer; ECN in IP would be sufficient.

A degenerate example is a point-to-point Ethernet link. Excess loading of the link merely causes the queue from the higher layer to back up, while the lower layer remains immune to congestion. Even a whole meshed subnetwork can be made immune to interior congestion by limiting ingress capacity and sufficient sizing of interior links, e.g., a non-blocking fat-tree network [Leiserson85]. An alternative to fat links near the root is numerous thin links with multi-path routing to ensure even worst-case patterns of load cannot congest any link, e.g., a Clos network [Clos53].

4. Feed-Forward-and-Up Mode: Guidelines for Adding Congestion Notification

Feed-forward-and-up is the mode already used for signalling ECN up the layers through MPLS into IP [RFC5129] and through IP-in-IP tunnels [RFC6040], whether encapsulating with IPv4 [RFC2003], IPv6 [RFC2473], or IPsec [RFC4301]. These RFCs take a consistent approach and the following guidelines are designed to ensure this consistency continues as ECN support is added to other protocols that encapsulate IP. The guidelines are also designed to ensure compliance with the more general best current practice for the design of alternate ECN schemes given in [RFC4774] and extended by [RFC8311].

The rest of this section is structured as follows:

- Section 4.1 addresses the most straightforward cases, where [RFC6040] can be applied directly to add ECN to tunnels that are effectively IP-in-IP tunnels, but with a shim header(s) between the IP headers.
- The subsequent sections give guidelines for adding congestion notification to a subnet technology that uses feed-forward-and-up mode like IP, but it is not so similar to IP that [RFC6040] rules can be applied directly. Specifically:
  - Sections 4.2, 4.3, and 4.4 address how to add ECN support to the wire protocol and to the encapsulators and decapsulators at the ingress and egress of the subnet, respectively.
  - Section 4.5 deals with the special but common case of sequences of tunnels or subnets that all use the same technology.
  - Section 4.6 deals with the question of reframing when IP packets do not map 1:1 into lower-layer frames.

4.1. IP-in-IP Tunnels with Shim Headers

A common pattern for many tunnelling protocols is to encapsulate an inner IP header with a shim header(s) then an outer IP header. A shim header is defined as one that is not sufficient alone to forward the packet as an outer header. Another common pattern is for a shim to encapsulate an L2 header, which in turn encapsulates (or might encapsulate) an IP header. [RFC9601] clarifies that [RFC6040] is just as applicable when there are shims and even an L2 header between two IP headers.

However, it is not always feasible or necessary to propagate ECN between IP headers when separated by a shim. For instance, it might be too costly to dig to arbitrary depths to find an inner IP header, there may be little or no congestion within the tunnel by design (see null mode in Section 3.4 above), or a legacy implementation might not support ECN. In cases where a tunnel does not support ECN, it is important that the ingress does not copy the ECN field from an inner IP header to an outer. Therefore Section 4 of [RFC9601] requires network operators to configure the ingress of a tunnel that does not support ECN so that it zeros the ECN field in the outer IP header.

Nonetheless, in many cases it is feasible to propagate the ECN field between IP headers separated by shim headers and/or an L2 header. Particularly in the typical case when the outer IP header and the shim(s) are added (or removed) as part of the same procedure. Even if a shim encapsulates an L2 header, it is often possible to find an inner IP header within the L2 PDU and propagate ECN between that and the outer IP header. This can be thought of as a special case of the feed-up-and-forward mode (Section 3.2), so the guidelines for this mode apply (Section 5).

Numerous shim protocols have been defined for IP tunnelling. More recent ones, e.g., Geneve [RFC8926] and Generic UDP Encapsulation (GUE) [INTAREA-GUE] cite and follow [RFC6040]. Some earlier ones, e.g., CAPWAP [RFC5415] and LISP [RFC9300], cite [RFC3168], which is compatible with [RFC6040].

However, as Section 9.3 of [RFC3168] pointed out, ECN support needs to be defined for many earlier shim-based tunnelling protocols, e.g., L2TPv2 [RFC2661], L2TPv3 [RFC3931], GRE [RFC2784], PPTP [RFC2637], GTP [GTPv1] [GTPv1-U] [GTPv2-C], and Teredo [RFC4380], as well as some recent ones, e.g., VXLAN [RFC7348], NVGRE [RFC7637], and NSH [RFC8300].

All these IP-based encapsulations can be updated in one shot by simple reference to [RFC6040]. However, it would not be appropriate to update all these protocols from within the present guidance document. Instead, a companion specification [RFC9601] has the appropriate Standards Track status to update Standards Track protocols. For those that are not under IETF change control [RFC9601] can only recommend that the relevant body updates them.

4.2. Wire Protocol Design: Indication of ECN Support

This section is intended to guide the redesign of any lower-layer protocol that encapsulates IP to add built-in congestion notification support at the lower layer using feed-forward-and-up mode. It reflects the approaches used in [RFC6040] and in [RFC5129]. Therefore, IP-in-IP tunnels or IP-in-MPLS or MPLS-in-MPLS encapsulations that already comply with [RFC6040] or [RFC5129] will already satisfy this guidance.

A lower-layer (or subnet) congestion notification system:

1. **SHOULD NOT** apply explicit congestion notifications to PDUs that are destined for legacy layer-4 transport implementations that will not understand ECN; and
2. **SHOULD NOT** apply explicit congestion notifications to PDUs if the egress of the subnet might not propagate congestion notification onward into the higher layer.

   We use the term ECN-PDU for a PDU on a feedback loop that will propagate congestion notification properly because it meets both the above criteria. Additionally, a Not-ECN-PDU is a PDU on a feedback loop that does not meet at least one of the criteria, and therefore will not propagate congestion notification properly. A corollary of the above is that a lower-layer congestion notification protocol:

3. **SHOULD** be able to distinguish ECN-PDUs from Not-ECN-PDUs.

Note that there is no need for all interior nodes within a subnet to be able to mark congestion explicitly. A mix of drop and explicit congestion signals from different nodes is fine. However, if *any* interior nodes might generate congestion markings, Guideline 2 above says that all relevant egress nodes **SHOULD** be able to propagate those markings up to the higher layer.

In IP, if the ECN field in each PDU is cleared to the Not ECN-Capable Transport (Not-ECT) codepoint, it indicates that the L4 transport will not understand congestion markings. A congested buffer must not mark these Not-ECT PDUs; therefore, it has to signal congestion by increasingly applying drop instead.

The mechanism a lower layer uses to distinguish the ECN capability of PDUs need not mimic that of IP. The above guidelines merely say that the lower-layer system as a whole should achieve the same outcome. For instance, ECN-capable feedback loops might use PDUs that are identified by a particular set of labels or tags. Alternatively, logical-link protocols that use flow state might determine whether a PDU can be congestion marked by checking for ECN support in the flow state. Other protocols might depend on out-of-band control signals.

The per-domain checking of ECN support in MPLS [RFC5129] is a good example of a way to avoid sending congestion markings to L4 transports that will not understand them without using any header space in the subnet protocol.

In MPLS, header space is extremely limited; therefore, [RFC5129] does not provide a field in the MPLS header to indicate whether the PDU is an ECN-PDU or a Not-ECN-PDU. Instead, interior nodes in a domain are allowed to set explicit congestion indications without checking whether

the PDU is destined for a L4 transport that will understand them. Nonetheless, this is made safe by requiring that the network operator upgrades all decapsulating edges of a whole domain at once as soon as even one switch within the domain is configured to mark rather than drop some PDUs during congestion. Therefore, any edge node that might decapsulate a packet will be capable of checking whether the higher-layer transport is ECN-capable. When decapsulating a CE-marked packet, if the decapsulator discovers that the higher layer (inner header) indicates the transport is not ECN-capable, it drops the packet -- effectively on behalf of the earlier congested node (see Decapsulation Guideline 1 in Section 4.4).

It was only appropriate to define such an incremental deployment strategy because MPLS is targeted solely at professional operators who can be expected to ensure that a whole subnetwork is consistently configured. This strategy might not be appropriate for other link technologies targeted at zero-configuration deployment or deployment by the general public (e.g., Ethernet). For such 'plug-and-play' environments, it will be necessary to invent a fail-safe approach that ensures congestion markings will never fall into black holes, no matter how inconsistently a system is put together. Alternatively, congestion notification relying on correct system configuration could be confined to flavours of Ethernet intended only for professional network operators, such as Provider Backbone Bridges (PBB) ([IEEE802.1Q]; previously 802.1ah).

ECN support in TRansparent Interconnection of Lots of Links (TRILL) [RFC9600] provides a good example of how to add congestion notification to a lower-layer protocol without relying on careful and consistent operator configuration. TRILL provides an extension header word with space for flags of different categories depending on whether logic to understand the extension is critical. The congestion-experienced marking has been defined as a 'critical ingress-to-egress' flag. So, if a transit RBridge sets this flag on a frame and an egress RBridge does not have any logic to process it, the egress RBridge will drop the frame, which is the desired default action anyway. Therefore, TRILL RBridges can be updated with support for congestion notification in no particular order and, at the egress of the TRILL campus, congestion notification will be propagated to IP as ECN whenever ECN logic has been implemented at the egress, or as drop otherwise.

QCN [IEEE802.1Q] is not intended to extend beyond a single subnet or interoperate with IP-ECN. Nonetheless, the way QCN indicates to lower-layer devices that the endpoints will not understand QCN provides another example that a lower-layer protocol designer might be able to mimic for their scenario. An operator can define certain Priority Code Points (PCPs [IEEE802.1Q]; previously 802.1p) to indicate non-QCN frames. Then an ingress bridge has to map each arriving not-QCN-capable IP packet to one of these non-QCN PCPs.

When drop for non-ECN traffic is deferred to the egress of a subnet, it cannot necessarily be assumed that one congestion mark is equivalent to one drop, as was originally required by [RFC3168]. [RFC8311] updated [RFC3168] to allow experimentation with congestion markings that are not equivalent to drop, particularly for L4S [RFC9331]. ECN support in TRILL [RFC9600] is a good example of a way to defer drop to the egress of a subnet both when marks are equivalent to drops (as in [RFC3168]) and when they are not (as in L4S). The ECN scheme for MPLS [RFC5129] was defined before L4S, so it only currently supports deferred drop that is

equivalent to ECN marking. Nonetheless, in principle, MPLS (and potentially future L2 protocols) could support L4S marking by copying TRILL's approach for determining the drop level of any non-ECN traffic at the subnet egress.

4.3. Encapsulation Guidelines

This section is intended to guide the redesign of any node that encapsulates IP with a lower-layer header when adding built-in congestion notification support to the lower-layer protocol using feed-forward-and-up mode. It reflects the approaches used in [RFC6040] and [RFC5129]. Therefore, IP-in-IP tunnels or IP-in-MPLS or MPLS-in-MPLS encapsulations that already comply with [RFC6040] or [RFC5129] will already satisfy this guidance.

1. Egress Capability Check: A subnet ingress needs to be sure that the corresponding egress of a subnet will propagate any congestion notification added to the outer header across the subnet. This is necessary in addition to checking that an incoming PDU indicates an ECN-capable (L4) transport. Examples of how this guarantee might be provided include:
   - by configuration (e.g., if any label switch in a domain supports congestion marking, [RFC5129] requires all egress nodes to have been configured to propagate ECN).
   - by the ingress explicitly checking that the egress propagates ECN (e.g., an early attempt to add ECN support to TRILL used IS-IS to check path capabilities before adding ECN extension flags to each frame [RFC7780]).
   - by inherent design of the protocol (e.g., by encoding congestion marking on the outer header in such a way that a legacy egress that does not understand ECN will consider the PDU corrupt or invalid and discard it; thus, at least propagating a form of congestion signal).
2. Egress Fails Capability Check: If the ingress cannot guarantee that the egress will propagate congestion notification, the ingress **SHOULD** disable congestion notification at the lower layer when it forwards the PDU. An example of how the ingress might disable congestion notification at the lower layer would be by setting the outer header of the PDU to identify it as a Not-ECN-PDU, assuming the subnet technology supports such a concept.
3. Standard Congestion Monitoring Baseline: Once the ingress to a subnet has established that the egress will correctly propagate ECN, on encapsulation, it **SHOULD** encode the same level of congestion in outer headers as is arriving in incoming headers. For example, it might copy any incoming congestion notifications into the outer header of the lower-layer protocol.

   This ensures that bulk congestion monitoring of outer headers (e.g., by a network management node monitoring congestion markings in passing frames) will measure congestion accumulated along the whole upstream path, starting from the Load Regulator and not just starting from the ingress of the subnet. A node that is not the Load Regulator **SHOULD NOT** re-initialize the level of CE markings in the outer header to zero.

It would still also be possible to measure congestion introduced across one subnet (or tunnel) by subtracting the level of CE markings on inner headers from that on outer headers (see Appendix C of [RFC6040]). For example:

- If this guideline has been followed and if the level of CE markings is 0.4% on the outer header and 0.1% on the inner header, 0.4% congestion has been introduced across all the networks since the Load Regulator, and 0.3% (= 0.4% - 0.1%) has been introduced since the ingress to the current subnet (or tunnel).
- Without this guideline, if the subnet ingress had re-initialized the outer congestion level to zero, the outer and inner headers would measure 0.1% and 0.3%. It would still be possible to infer that the congestion introduced since the Load Regulator was 0.4% (= 0.1% + 0.3%), but only if the monitoring system somehow knows whether the subnet ingress re-initialized the congestion level.

As long as subnet and tunnel technologies use the standard congestion monitoring baseline in this guideline, monitoring systems will know to use the former approach rather than having to 'somehow know' which approach to use.

4.4. Decapsulation Guidelines

This section is intended to guide the redesign of any node that decapsulates IP from within a lower-layer header when adding built-in congestion notification support to the lower-layer protocol using feed-forward-and-up mode. It reflects the approaches used in [RFC6040] and in [RFC5129]. Therefore, IP-in-IP tunnels or IP-in-MPLS or MPLS-in-MPLS encapsulations that already comply with [RFC6040] or [RFC5129] will already satisfy this guidance.

A subnet egress **SHOULD NOT** simply copy congestion notifications from outer headers to the forwarded header. It **SHOULD** calculate the outgoing congestion notification field from the inner and outer headers using the following guidelines. If there is any conflict, rules earlier in the list take precedence over rules later in the list.

1. If the arriving inner header is a Not-ECN-PDU, it implies the L4 transport will not understand explicit congestion markings. Then:
   - If the outer header carries an explicit congestion marking, it is likely that a protocol error has occurred, so drop is the only indication of congestion that the L4 transport will understand. If the outer congestion marking is the most severe possible, the packet **MUST** be dropped. However, if congestion can be marked with multiple levels of severity and the packet's outer marking is not the most severe, this requirement can be relaxed to: the packet **SHOULD** be dropped.
   - If the outer is an ECN-PDU that carries no indication of congestion or a Not-ECN-PDU the PDU **SHOULD** be forwarded, but still as a Not-ECN-PDU.
2. If the outer header does not support congestion notification (a Not-ECN-PDU), but the inner header does (an ECN-PDU), the inner header **SHOULD** be forwarded unchanged.
3. In some lower-layer protocols, congestion may be signalled as a numerical level, such as in the control frames of QCN [IEEE802.1Q]. If such a multi-bit encoding encapsulates an ECN-

capable IP data packet, a function will be needed to convert the quantized congestion level into the frequency of congestion markings in outgoing IP packets.

4. Congestion indications might be encoded by a severity level. For instance, increasing levels of congestion might be encoded by numerically increasing indications, e.g., PCN can be encoded in each PDU at three severity levels in IP or MPLS [RFC6660] and the default encapsulation and decapsulation rules [RFC6040] are compatible with this interpretation of the ECN field.

   If the arriving inner header is an ECN-PDU, where the inner and outer headers carry indications of congestion of different severity, the more severe indication **SHOULD** be forwarded in preference to the less severe.

5. The inner and outer headers might carry a combination of congestion notification fields that should not be possible given any currently used protocol transitions. For instance, if Encapsulation Guideline 3 in Section 4.3 had been followed, it should not be possible to have a less severe indication of congestion in the outer header than in the inner header. It **MAY** be appropriate to log unexpected combinations of headers and possibly raise an alarm.

   If a safe outgoing codepoint can be defined for such a PDU, the PDU **SHOULD** be forwarded rather than dropped. Some implementers discard PDUs with currently unused combinations of headers just in case they represent an attack. However, an approach using alarms and policy-mediated drop is preferable to hard-coded drop so that operators can keep track of possible attacks, but currently unused combinations are not precluded from future use through new standards actions.

4.5. Sequences of Similar Tunnels or Subnets

In some deployments, particularly in 3GPP networks, an IP packet may traverse two or more IP-in-IP tunnels in sequence that all use identical technology (e.g., GTP).

In such cases, it would be sufficient for every encapsulation and decapsulation in the chain to comply with [RFC6040]. Alternatively, as an optimization, a node that decapsulates a packet and immediately re-encapsulates it for the next tunnel **MAY** copy the incoming outer ECN field directly to the outgoing outer header and the incoming inner ECN field directly to the outgoing inner header. Then, the overall behaviour across the sequence of tunnel segments would still be consistent with [RFC6040].

Appendix C of [RFC6040] describes how a tunnel egress can monitor how much congestion has been introduced within a tunnel. A network operator might want to monitor how much congestion had been introduced within a whole sequence of tunnels. Using the technique in Appendix C of [RFC6040] at the final egress, the operator could monitor the whole sequence of tunnels, but only if the above optimization were used consistently along the sequence of tunnels, in order to make it appear as a single tunnel. Therefore, tunnel endpoint implementations **SHOULD** allow the operator to configure whether this optimization is enabled.

When congestion notification support is added to a subnet technology, consideration **SHOULD** be given to a similar optimization between subnets in sequence if they all use the same technology.

4.6. Reframing and Congestion Markings

The guidance in this section is worded in terms of framing boundaries, but it applies equally whether the PDUs are frames, cells, or packets.

Where an AQM marks the ECN field of IP packets as they queue into a Layer 2 link, there will be no problem with framing boundaries because the ECN markings would be applied directly to IP packets. The guidance in this section is only applicable where a congestion notification capability is being added to a Layer 2 protocol so that Layer 2 frames can be marked by an AQM at layer 2. This would only be necessary where AQM will be applied at pure Layer 2 nodes (without IP awareness).

Where congestion marking has had to be applied at non-IP-aware nodes and framing boundaries do not necessarily align with packet boundaries, the decapsulating IP forwarding node **SHOULD** propagate congestion markings from Layer 2 frame headers to IP packets that may have different boundaries as a consequence of reframing.

Two possible design goals for propagating congestion indications, described in Section 5.3 of [RFC3168] and Section 2.4 of [RFC7141], are:

1. approximate preservation of the presence (and therefore timing) of congestion marks on the L2 frames used to construct an IP packet;
2. a. at high frequency of congestion marking, approximate preservation of the proportion of congestion marks arriving and departing;
   b. at low frequency of congestion marking, approximate preservation of the timing of congestion marks arriving and departing.

In either case, an implementation **SHOULD** ensure that any new incoming congestion indication is propagated immediately; not held awaiting the possibility of further congestion indications to be sufficient to indicate congestion on an outgoing PDU [RFC7141]. Nonetheless, to facilitate pipelined implementation, it would be acceptable for congestion marks to propagate to a slightly later IP packet.

At decapsulation in either case:

- ECN-marking propagation logically occurs before application of Decapsulation Guideline 1 in Section 4.4. For instance, if ECN-marking propagation would cause an ECN congestion indication to be applied to an IP packet that is a Not-ECN-PDU, then that IP packet is dropped in accordance with Guideline 1.
- Where a mix of ECN-PDUs and non-ECN-PDUs arrives to construct the same IP packet, the decapsulation specification **SHOULD** require that packet to be discarded.
- Where a mix of different types of ECN-PDUs arrives to construct the same IP packet, e.g., a mix of frames that map to ECT(0) and ECT(1) IP packets, the decapsulation specification might consider this a protocol error. But, if the lower-layer protocol has defined such a mix of types of ECN-PDU as valid, it **SHOULD** require the resulting IP packet to be set to either ECT(0) or ECT(1). In this case, it **SHOULD** take into account that the RFC Series has so far

allowed ECT(0) and ECT(1) to be considered equivalent [RFC3168]; or ECT(1) can provide a less severe congestion marking than CE [RFC6040]; or ECT(1) can indicate an unmarked but ECN-capable packet that is subject to a different marking algorithm to ECT(0) packets, e.g., L4S [RFC8311] [RFC9331].

The following are two ways that goal 1 might be achieved, but they are not intended to be the only ways:

- Every IP PDU that is constructed, in whole or in part, from an L2 frame that is marked with a congestion signal has that signal propagated to it.
- Every L2 frame that is marked with a congestion signal propagates that signal to one IP PDU that is constructed from it in whole or in part. If multiple IP PDUs meet this description, the choice can be made arbitrarily but ought to be consistent.

The following gives one way that goal 2 might be achieved, but it is not intended to be the only way:

- For each of the streams of frames that encapsulate the IP packets of each IP-ECN codepoint and follow the same path through the subnet, a counter ('in') tracks octets arriving within the payload of marked L2 frames and another ('out') tracks octets departing in marked IP packets. While 'in' exceeds 'out', forwarded IP packets are ECN-marked. If 'out' exceeds 'in' for longer than a timeout, both counters are zeroed to ensure that the start of the next congestion episode propagates immediately. The 'out' counter includes octets in reconstructed IP packets that would have been marked, but had to be dropped because they were Not-ECN-PDUs (by Decapsulation Guideline 1 in Section 4.4).

Generally, relative to the number of IP PDUs, the number of L2 frames may be higher (e.g., ATM), roughly the same, or lower (e.g., 802.11 aggregation at an L2-only station). This distinction may influence the choice of mechanism.

5. Feed-Up-and-Forward Mode: Guidelines for Adding Congestion Notification

The guidance in this section is applicable, for example, when IP packets:

- are encapsulated in Ethernet headers, which have no support for congestion notification;
- are forwarded by the eNode-B (base station) of a 3GPP radio access network, which is required to apply ECN marking during congestion [LTE-RA] [UTRAN], but the Packet Data Convergence Protocol (PDCP) that encapsulates the IP header over the radio access has no support for ECN.

This guidance also generalizes to encapsulation by other subnet technologies with no built-in support for congestion notification at the lower layer, but with support for finding and processing an IP header. It is unlikely to be applicable or necessary for IP-in-IP encapsulation, where feed-forward-and-up mode based on [RFC6040] would be more appropriate.

Marking the IP header while switching at layer 2 (by using a Layer 3 switch) or while forwarding in a radio access network seems to represent a layering violation. However, it can be considered as a benign optimization if the guidelines below are followed. Feed-up-and-forward is certainly not a general alternative to implementing feed-forward congestion notification in the lower layer, because:

- IPv4 and IPv6 are not the only Layer 3 protocols that might be encapsulated by lower-layer protocols.
- Link-layer encryption might be in use, making the Layer 2 payload inaccessible.
- Many Ethernet switches do not have 'Layer 3 switch' capabilities, so the ability to read or modify an IP payload cannot be assumed.
- It might be costly to find an IP header (IPv4 or IPv6) when it may be encapsulated by more than one lower-layer header, e.g., Ethernet MAC in MAC ([IEEE802.1Q]; previously 802.1ah).

Nonetheless, configuring lower-layer equipment to look for an ECN field in an encapsulated IP header is a useful optimization. If the implementation follows the guidelines below, this optimization does not have to be confined to a controlled environment, e.g., within a data centre; it could usefully be applied in any network -- even if the operator is not sure whether the above issues will never apply:

1. If a built-in lower-layer congestion notification mechanism exists for a subnet technology, it is safe to mix feed-up-and-forward with feed-forward-and-up on other switches in the same subnet. However, it will generally be more efficient to use the built-in mechanism.
2. The depth of the search for an IP header **SHOULD** be limited. If an IP header is not found soon enough, or an unrecognized or unreadable header is encountered, the switch **SHOULD** resort to an alternative means of signalling congestion (e.g., drop or the built-in lower-layer mechanism if available).
3. It is sufficient to use the first IP header found in the stack; the egress of the relevant tunnel can propagate congestion notification upwards to any more deeply encapsulated IP headers later.

6. Feed-Backward Mode: Guidelines for Adding Congestion Notification

It can be seen from Section 3.3 that congestion notification in a subnet using feed-backward mode has generally not been designed to be directly coupled with IP-layer congestion notification. The subnet attempts to minimize congestion internally, and if the incoming load at the ingress exceeds the capacity somewhere through the subnet, the Layer 3 buffer into the ingress backs up. Thus, a feed-backward mode subnet is in some sense similar to a null mode subnet, in that there is no need for any direct interaction between the subnet and higher-layer congestion notification. Therefore, no detailed protocol design guidelines are appropriate. Nonetheless, a more general guideline is appropriate:

> A subnetwork technology intended to eventually interface to IP **SHOULD NOT** be designed using only the feed-backward mode, which is certainly best for a stand-alone subnet, but would need to be modified to work efficiently as part of the wider Internet because IP uses feed-forward-and-up mode.

The feed-backward approach at least works beneath IP, where the term 'works' is used only in a narrow functional sense because feed-backward can result in very inefficient and sluggish congestion control -- except if it is confined to the subnet directly connected to the original data source when it is faster than feed-forward. It would be valid to design a protocol that could work in feed-backward mode for paths that only cross one subnet, and in feed-forward-and-up mode for paths that cross subnets.

In the early days of TCP/IP, a similar feed-backward approach was tried for explicit congestion signalling using source-quench (SQ) ICMP control packets. However, SQ fell out of favour and is now formally deprecated [RFC6633]. The main problem was that it is hard for a data source to tell the difference between a spoofed SQ message and a quench request from a genuine buffer on the path. It is also hard for a lower-layer buffer to address an SQ message to the original source port number, which may be buried within many layers of headers and possibly encrypted.

QCN (also known as Backward Congestion Notification (BCN); see Sections 30-33 of [IEEE802.1Q], previously known as 802.1Qau) uses a feed-backward mode that is structurally similar to ATM's relative rate mechanism. However, QCN confines its applicability to scenarios such as some data centres where all endpoints are directly attached by the same Ethernet technology. If a QCN subnet were later connected into a wider IP-based internetwork (e.g., when attempting to interconnect multiple data centres) it would suffer the inefficiency shown in Figure 3.

7. IANA Considerations

This document has no IANA actions.

8. Security Considerations

If a lower-layer wire protocol is redesigned to include explicit congestion signalling in-band in the protocol header, care **SHOULD** be taken to ensure that the field used is specified as mutable during transit. Otherwise, interior nodes signalling congestion would invalidate any authentication protocol applied to the lower-layer header -- by altering a header field that had been assumed as immutable.

The redesign of protocols that encapsulate IP in order to propagate congestion signals between layers raises potential signal integrity concerns. Experimental or proposed approaches exist for assuring the end-to-end integrity of in-band congestion signals, such as:

- Congestion Exposure (ConEx) for networks:
  - to audit that their congestion signals are not being suppressed by other networks or by receivers; and
  - to police that senders are responding sufficiently to the signals, irrespective of the L4 transport protocol used [RFC7713].
- A test for a sender to detect whether a network or the receiver is suppressing congestion signals (for example, see the second paragraph of Section 20.2 of [RFC3168]).

Given these end-to-end approaches are already being specified, it would make little sense to attempt to design hop-by-hop congestion signal integrity into a new lower-layer protocol because end-to-end integrity inherently achieves hop-by-hop integrity.

Section 6 gives vulnerability to spoofing as one of the reasons for deprecating feed-backward mode.

9. Conclusions

Following the guidance in this document enables ECN support to be extended consistently to numerous protocols that encapsulate IP (IPv4 and IPv6) so that IP continues to fulfil its role as an end-to-end interoperability layer. This includes:

- A wide range of tunnelling protocols, including those with various forms of shim header between two IP headers, possibly also separated by an L2 header;
- A wide range of subnet technologies, particularly those that work in the same 'feed-forward-and-up' mode that is used to support ECN in IP and MPLS.

Guidelines have been defined for supporting propagation of ECN between Ethernet and IP on so-called Layer 3 Ethernet switches using a 'feed-up-and-forward' mode. This approach could enable other subnet technologies to pass ECN signals into the IP layer, even if they do not support ECN.

Finally, attempting to add congestion notification to a subnet technology in feed-backward mode is deprecated except in special cases due to its likely sluggish response to congestion.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[RFC3819] Karn, P., Ed., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, DOI 10.17487/RFC3819, July 2004, <https://www.rfc-editor.org/info/rfc3819>.

[RFC4774] Floyd, S., "Specifying Alternate Semantics for the Explicit Congestion Notification (ECN) Field", BCP 124, RFC 4774, DOI 10.17487/RFC4774, November 2006, <https://www.rfc-editor.org/info/rfc4774>.

[RFC5129] Davie, B., Briscoe, B., and J. Tay, "Explicit Congestion Marking in MPLS", RFC 5129, DOI 10.17487/RFC5129, January 2008, <https://www.rfc-editor.org/info/rfc5129>.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <https://www.rfc-editor.org/info/rfc6040>.

[RFC7141] Briscoe, B. and J. Manner, "Byte and Packet Congestion Notification", BCP 41, RFC 7141, DOI 10.17487/RFC7141, February 2014, <https://www.rfc-editor.org/info/rfc7141>.

[RFC9600] Eastlake 3rd, D. and B. Briscoe, "TRILL (TRansparent Interconnection of Lots of Links): ECN (Explicit Congestion Notification) Support", RFC 9600, DOI 10.17487/RFC9600, June 2024, <https://www.rfc-editor.org/info/rfc9600>.

10.2. Informative References

[ATM-TM-ABR] Cisco, "Understanding the Available Bit Rate (ABR) Service Category for ATM VCs", Design Technote 10415, June 2005, <https://www.cisco.com/c/en/us/support/docs/asynchronous-transfer-mode-atm/atm-traffic-management/10415-atmabr.html>.

[Buck00] Buckwalter, J.T., "Frame Relay: Technology and Practice", Addison-Wesley Professional, ISBN-13 978-0201485240, 2000.

[Clos53] Clos, C., "A Study of Non-Blocking Switching Networks", The Bell System Technical Journal, Vol. 32, Issue 2, DOI 10.1002/j.1538-7305.1953.tb01433.x, March 1953, <https://doi.org/10.1002/j.1538-7305.1953.tb01433.x>.

[GTPv1] 3GPP, "General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface", Technical Specification 29.060.

[GTPv1-U] 3GPP, "General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)", Technical Specification 29.281.

[GTPv2-C] 3GPP, "3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3", Technical Specification 29.274.

[IEEE802.1Q] IEEE, "IEEE Standard for Local and Metropolitan Area Network--Bridges and Bridged Networks", IEEE Std 802.1Q-2022, DOI 10.1109/IEEESTD.2022.10004498, December 2022, <https://doi.org/10.1109/IEEESTD.2022.10004498>.

[INTAREA-GUE] Herbert, T., Yong, L., and O. Zia, "Generic UDP Encapsulation", Work in Progress, Internet-Draft, draft-ietf-intarea-gue-09, 26 October 2019, <https://datatracker.ietf.org/doc/html/draft-ietf-intarea-gue-09>.

[ITU-T.I.371] ITU-T, "Traffic control and congestion control in B-ISDN", ITU-T Recommendation I.371, March 2004, <https://www.itu.int/rec/T-REC-I.371-200403-I/en>.

[Leiserson85] Leiserson, C.E., "Fat-trees: Universal networks for hardware-efficient supercomputing", IEEE Transactions on Computers, Vol. C-34, Issue 10, DOI 10.1109/TC.1985.6312192, October 1985, <https://doi.org/10.1109/TC.1985.6312192>.

[LTE-RA] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2", Technical Specification 36.300.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <https://www.rfc-editor.org/info/rfc2003>.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[RFC2637] Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little, W., and G. Zorn, "Point-to-Point Tunneling Protocol (PPTP)", RFC 2637, DOI 10.17487/RFC2637, July 1999, <https://www.rfc-editor.org/info/rfc2637>.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, DOI 10.17487/RFC2661, August 1999, <https://www.rfc-editor.org/info/rfc2661>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC2884] Hadi Salim, J. and U. Ahmed, "Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks", RFC 2884, DOI 10.17487/RFC2884, July 2000, <https://www.rfc-editor.org/info/rfc2884>.

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, DOI 10.17487/RFC2983, October 2000, <https://www.rfc-editor.org/info/rfc2983>.

[RFC3931] Lau, J., Ed., Townsley, M., Ed., and I. Goyret, Ed., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, DOI 10.17487/RFC3931, March 2005, <https://www.rfc-editor.org/info/rfc3931>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, DOI 10.17487/RFC4380, February 2006, <https://www.rfc-editor.org/info/rfc4380>.

[RFC5415] Calhoun, P., Ed., Montemurro, M., Ed., and D. Stanley, Ed., "Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification", RFC 5415, DOI 10.17487/RFC5415, March 2009, <https://www.rfc-editor.org/info/rfc5415>.

[RFC6633] Gont, F., "Deprecation of ICMP Source Quench Messages", RFC 6633, DOI 10.17487/RFC6633, May 2012, <https://www.rfc-editor.org/info/rfc6633>.

[RFC6660] Briscoe, B., Moncaster, T., and M. Menth, "Encoding Three Pre-Congestion Notification (PCN) States in the IP Header Using a Single Diffserv Codepoint (DSCP)", RFC 6660, DOI 10.17487/RFC6660, July 2012, <https://www.rfc-editor.org/info/rfc6660>.

[RFC7323] Borman, D., Braden, B., Jacobson, V., and R. Scheffenegger, Ed., "TCP Extensions for High Performance", RFC 7323, DOI 10.17487/RFC7323, September 2014, <https://www.rfc-editor.org/info/rfc7323>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7567] Baker, F., Ed. and G. Fairhurst, Ed., "IETF Recommendations Regarding Active Queue Management", BCP 197, RFC 7567, DOI 10.17487/RFC7567, July 2015, <https://www.rfc-editor.org/info/rfc7567>.

[RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <https://www.rfc-editor.org/info/rfc7637>.

[RFC7713] Mathis, M. and B. Briscoe, "Congestion Exposure (ConEx) Concepts, Abstract Mechanism, and Requirements", RFC 7713, DOI 10.17487/RFC7713, December 2015, <https://www.rfc-editor.org/info/rfc7713>.

[RFC7780] Eastlake 3rd, D., Zhang, M., Perlman, R., Banerjee, A., Ghanwani, A., and S. Gupta, "Transparent Interconnection of Lots of Links (TRILL): Clarifications, Corrections, and Updates", RFC 7780, DOI 10.17487/RFC7780, February 2016, <https://www.rfc-editor.org/info/rfc7780>.

[RFC8084] Fairhurst, G., "Network Transport Circuit Breakers", BCP 208, RFC 8084, DOI 10.17487/RFC8084, March 2017, <https://www.rfc-editor.org/info/rfc8084>.

[RFC8087] Fairhurst, G. and M. Welzl, "The Benefits of Using Explicit Congestion Notification (ECN)", RFC 8087, DOI 10.17487/RFC8087, March 2017, <https://www.rfc-editor.org/info/rfc8087>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", RFC 8257, DOI 10.17487/RFC8257, October 2017, <https://www.rfc-editor.org/info/rfc8257>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[RFC8311] Black, D., "Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation", RFC 8311, DOI 10.17487/RFC8311, January 2018, <https://www.rfc-editor.org/info/rfc8311>.

[RFC8926] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", RFC 8926, DOI 10.17487/RFC8926, November 2020, <https://www.rfc-editor.org/info/rfc8926>.

[RFC9300] Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, Ed., "The Locator/ID Separation Protocol (LISP)", RFC 9300, DOI 10.17487/RFC9300, October 2022, <https://www.rfc-editor.org/info/rfc9300>.

[RFC9331] De Schepper, K. and B. Briscoe, Ed., "The Explicit Congestion Notification (ECN) Protocol for Low Latency, Low Loss, and Scalable Throughput (L4S)", RFC 9331, DOI 10.17487/RFC9331, January 2023, <https://www.rfc-editor.org/info/rfc9331>.

[RFC9601] Briscoe, B., "Propagating Explicit Congestion Notification Across IP Tunnel Headers Separated by a Shim", RFC 9601, DOI 10.17487/RFC9601, June 2024, <https://www.rfc-editor.org/info/rfc9601>.

[UTRAN] 3GPP, "UTRAN overall description", Technical Specification 25.401.

Acknowledgements

Thanks to Gorry Fairhurst and David Black for extensive reviews. Thanks also to the following reviewers: Joe Touch, Andrew McGregor, Richard Scheffenegger, Ingemar Johansson, Piers O'Hanlon, Donald Eastlake 3rd, Jonathan Morton, Markku Kojo, Sebastian Möller, Martin Duke, and Michael Welzl, who pointed out that lower-layer congestion notification signals may have different semantics to those in IP. Thanks are also due to the Transport and Services Working Group (tsvwg) chairs, TSV ADs and IETF liaison people such as Eric Gray, Dan Romascanu and Gonzalo Camarillo for helping with the liaisons with the IEEE and 3GPP. And thanks to Georg Mayer and particularly to Erik Guttman for the extensive search and categorization of any 3GPP specifications that cite ECN specifications. Thanks also to the Area Reviewers Dan Harkins, Paul Kyzivat, Sue Hares, and Dale Worley.

Bob Briscoe was part-funded by the European Community under its Seventh Framework Programme through the Trilogy project (ICT-216372) for initial drafts then through the Reducing Internet Transport Latency (RITE) project (ICT-317700), and for final drafts (from -18) he was funded by Apple Inc. The views expressed here are solely those of the authors.

Contributors

Pat Thaler
Broadcom Corporation (retired)
CA
United States of America

Pat was a coauthor of this document, but retired before its publication.

Authors' Addresses

Bob Briscoe
Independent
United Kingdom
Email: ietf@bobbriscoe.net
URI: https://bobbriscoe.net/

John Kaippallimalil
Futurewei
5700 Tennyson Parkway, Suite 600
Plano, Texas 75024
United States of America
Email: kjohn@futurewei.com