An Open ECN Service in the IP layer

Bob Briscoe
<bob.briscoe@bt.com> <www.btexact.com/people/briscorj/>
BT Research, B54/130, Adastral Park, Martlesham Heath, Ipswich, IP5 3RE, England
Tel. +44 1473 645196

29 Dec 2001

1 Abstract

This document contributes to the effort to add explicit congestion notification (ECN) to IP. In the current effort to standardise ECN for TCP it is unavoidably necessary to standardise certain new aspects of IP. However, the IP aspects will not and cannot only be specific to TCP. We specify interaction with features of IP such as fragmentation, differentiated services, multicast forwarding, and a definition of the service offered to higher layer congestion control protocols. This document only concerns aspects related to the IP layer, but includes any aspects likely to be common to all higher layer protocols. Any specification of ECN support in higher layer protocols is expected to appear in a separate specification for each such protocol.

2 Introduction

{Update this to reflect the new purpose}

This document is intended to improve the specifications incorporating explicit congestion notification (ECN) into IP. It is intended to complement the existing Internet Draft on addition of ECN to TCP/IP [13] in order to hasten the proposal to the IETF standards track. We envisage the current document being absorbed into that I-D where agreement is reached on the issues discussed. Therefore, for brevity, we will not make this document stand alone; we hope the authors of that I-D will not be offended if we presume to write an addendum to the above I-D. We have tried to avoid conflicts with that I-D, wherever possible suggesting additions rather than changes.

In this document we only focus on issues with ECN at the IP layer (v4 & v6). In order to standardise ECN behaviour in TCP it is unavoidably necessary to standardise certain aspects in IP. However, the IP aspects will not and cannot only be specific to TCP. We believe the introduction of ECN into TCP/IP is best achieved in two documents, one on IP and the other on TCP. Therefore, in this document we solely discuss aspects of ECN that will be common to all protocols layered over IP.

For the history and status of the endeavour to add ECN to the Internet, also refer to [13]. We share the desire of that work to ensure backwards compatibility, and offer this work with the aim of also ensuring forwards flexibility.

In the remainder of this document we first define our terms. Then we focus on router behaviour, in particular differentiated queuing and multicast forwarding. Next we move the focus to host behaviour, particularly clarifying the ECN support that any congestion control protocol should be able to expect from the IP layer. We also give general requirements on such congestion control algorithms. Next we discuss fragmentation and re-assembly issues specific to IPv4. Finally we clarify access rights to ECN fields and discuss other security issues.

3 Conventions, definitions and acronyms

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [5]. Of course, readers should note that this is an Internet draft, and such keywords have no force unless the status of the document moves beyond draft.

We use the tuple (ECT, CE) to represent the settings of the flags in the ECN field of the IP packet

header. When set, [13] defines them to mean respectively ECN capable transport, and congestion experienced.

These two flags in the ECN field can currently be treated separately. If, on the other hand, the two bit field is considered as four code points, currently only three have unanimously proposed uses. The fourth (ECT=0, CE=1) remains undefined, but with four speculative uses proposed from various quarters (we agree with one — see later).

For clarity, where appropriate, the terms ECT and CE are used for the ECN flags (bits), while the succinct terms below will always be used in this document to refer to packets with the given code-points:

- markable (ECT=1, CE=*);
- unmarkable (ECT=0, CE=0).
- marked (ECT=1, CE=1);
- unmarked (ECT=1, CE=0);

Note that markable traffic includes marked traffic but that unmarkable traffic does not include unmarked traffic. If required, the intention is to allow these definitions to include more code-points in the future without rewriting the whole document. For instance, in the future, both 'markable' and 'marked' might be redefined to include (ECT=0, CE=1).

The terms marker, marking, pre-marking and re-marking are already defined concerning the setting of the diffserv code-point [3]. If it is ever not clear from the context whether we are discussing diffserv marking or ECN marking, we will use the terms congestion markable, congestion marked etc.

4 ECN router marking algorithms and differentiated services

4.1 Specification of marking behaviour

The ECN specification for TCP/IP [13] expects the random early detection (RED) algorithm [6, 4] to be used to mark traffic that is markable. It also accepts that other active queue management mechanisms may be developed and used. For instance, a virtual queue has been suggested to trigger marking even before queuing starts [7]. In this proposal, as packets enter the real queue a reference to them is also placed in the virtual queue. But the virtual queue has a smaller buffer and is emptied at a slower rate than the real one. Whenever the virtual queue is in an overflow state, all packets leaving the real queue are marked.

We believe that it is important for the marking behaviour of routers to be predictable for the hosts using them. As the art of active queue management evolves, it should not be possible for completely different marking behaviours to be invoked at each router along a path. We wish to point out that a framework for experimentation with and competition between queuing behaviours already exists: the differentiated services architecture [3]. The per hop behaviour (PHB) associated with each diffserv code point (DSCP) can already be specified. The guidelines on PHB specification in the diffserv architecture include the discard behaviour [3, Section 3].

Fig 1 shows how traffic classes only distinguished by ECN marking algorithm could simultaneously offer both a low latency service (buffer starving) and an improved best effort service (buffer filling). Based on DSCP traffic classification, one class would be marked early in the onset of congestion, while the other class would be marked later. If the end-systems reacted to these congestion signals, the end-systems would create the required service differentiation by their behaviour, without a need for one class to be over-provisioned in the network.

In future, PHBs MUST also define the congestion marking behaviour[1] of markable traffic if they define the discard behaviour of unmarkable traffic. Where appropriate, of course, such a definition MAY simply state that markable traffic is treated as if it were unmarkable. The addition of a need to define marking behaviour UPDATES the guidelines in the diffserv architecture referred to above. In the absence of descriptions of discard and marking behaviour, the implementation will determine the default marking behaviour.

Whether the definition of a PHB MUST be through standardisation or MAY be by local definition depends on which pool its code-point falls within [12, Section 6]. The same would obviously apply to the marking behaviour.

[1] Note that congestion marking behaviour is distinct from traffic contract policing behaviour. The former doesn't discriminate flows or customers, as distinct from the latter which identifies out of contract traffic on a per-customer basis at the network interface with that customer.

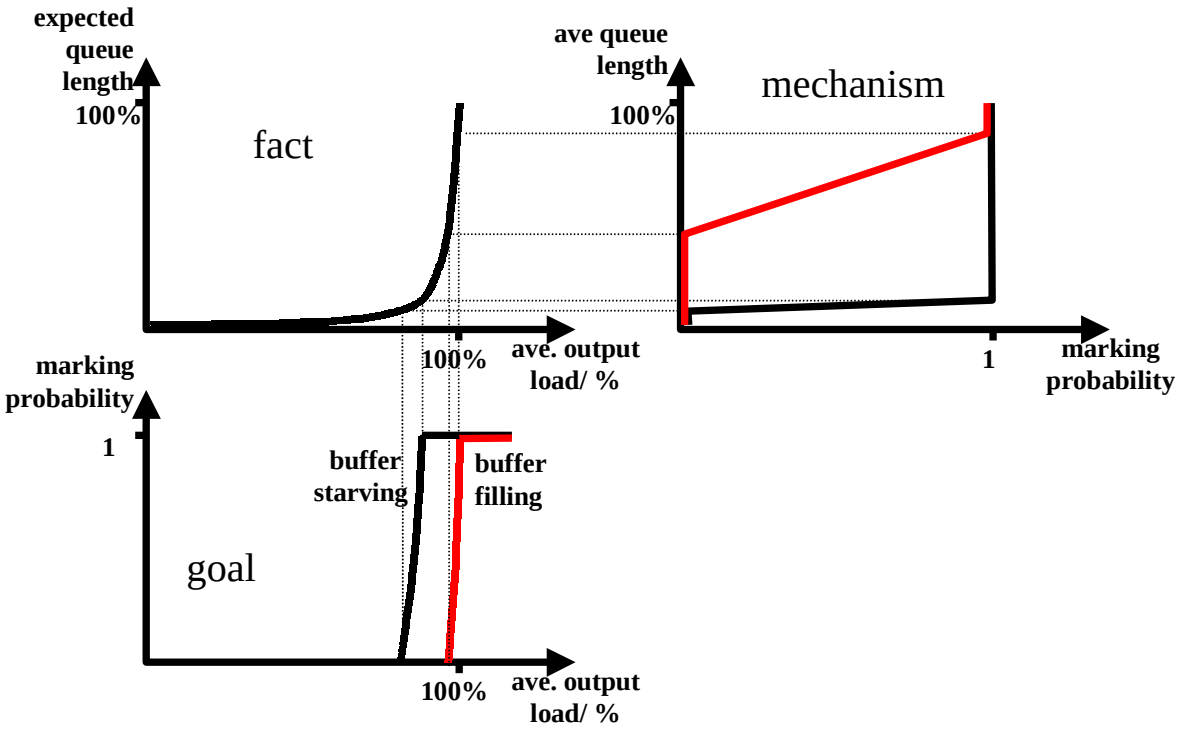

Figure 1: Buffer filling vs. starving

Thus, the best effort (default) PHB might be standardised by specifying its congestion marking behaviour as the RED algorithm and by giving its parameters. Other PHBs might be offered by network operators each using a different algorithm to trigger congestion notification, such as a virtual queue.

4.2 Equivalence between marking and drop behaviour

The ECN specification for TCP/IP [13] stipulates that a packet should only be congestion marked if it would have been dropped, were it unmarkable. It is even stipulated that this assumption should be embedded in implementations, by stating that the ECT flag should only be checked after the decision has been made to drop a packet. Exactly mimicking drop behaviour is motivated by the need to provide incentives for hosts to switch to ECN capability when competing with unmarkable flows. Indeed, [13] accepts that research into new criteria will be necessary for environments where all end-nodes are ECN-capable.

It is perfectly possible that future end-to-end congestion control protocols may be developed in conjunction with new router behaviours. For such a new service treatment, the router might be required to drop markable packets under the same conditions as unmarkable packets (the buffer starving discipline in Fig 2). However, markable packets would have to be marked at a far lower level of utilisation. In these new protocols, hosts would then be required to react far less severely to a marked packet than to a dropped one.

The incentive for sending markable packets into such a service discipline would be the extra feedback from the network, which would make applications of this service behave far more smoothly. Such a service would be valuable for applications that benefitted from rate stability. Another perfectly reasonable possibility is that the incentive to send markable packets into the network will be provided by a lower charge than for unmarkable packets. Such incentives are not appropriate for the best effort service which best serves its relatively elastic data applications by keeping queues relatively full. However, these incentives make sense for applications requiring the low latency of empty queues.

Thus, there is clearly a need to ensure space for future experimentation. Each approach would have to define the standard point of equivalence between the behaviours for markable and unmarkable packets. Nonetheless, it is perfectly reasonable to restrict all

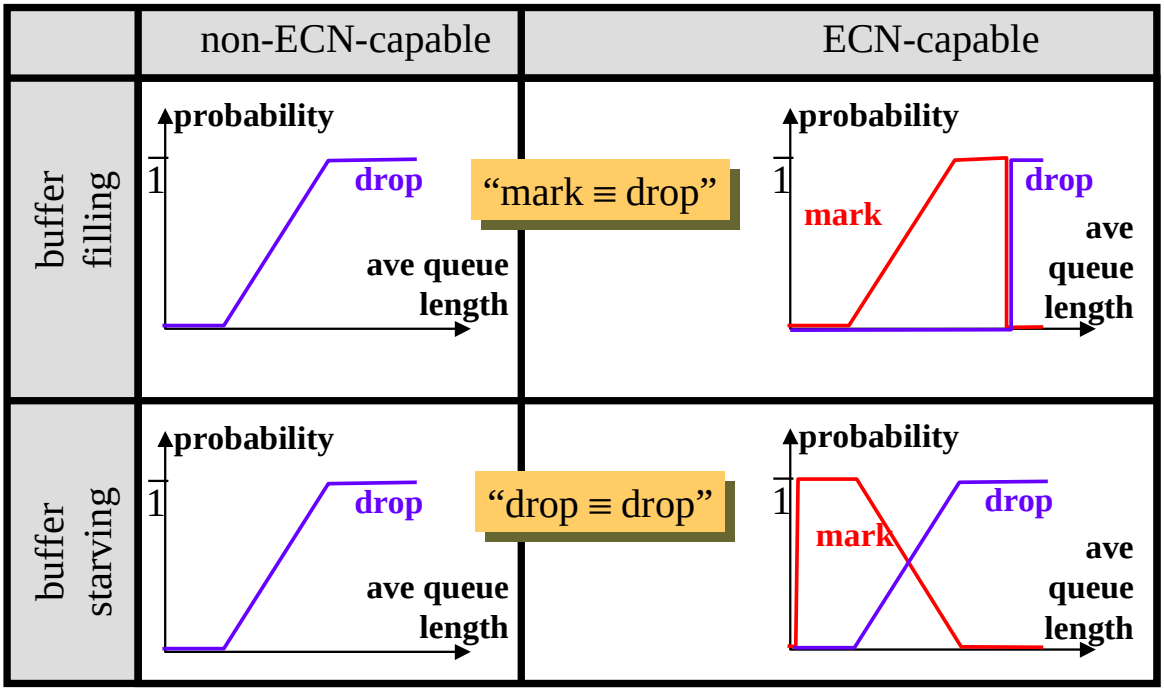

Figure 2: ECN mark/drop equivalence

protocols within a service treatment to the same standard. Otherwise, routers would have to examine the protocol field to determine the queuing behaviour.

Therefore, the equivalence proposed in [13] is appropriate for all protocols using the best effort service. However, it would be unnecessary and probably incorrect to make such a sweeping restriction across every differentiated service. Indeed, it will often be meaningless to mimic the drop behaviour of a PHB that never existed before ECN. In fact, it is perfectly possible that some operators might deny unmarkable traffic access to certain service treatments in the future.

To summarise, the point of equivalence between marking behaviour for markable packets and discard behaviour for unmarkable packets MUST be defined, but it MAY be different for each different service treatment.

4.3 Dependence on ECN-enabled routers

If a differentiated service is offered that depends on its marking behaviour for optimal functioning, it must also depend on how many and which routers are ECN-enabled. There may be good reason why certain routers cannot be upgraded cost-effectively, or why a neighbouring domain may choose not to upgrade any routers to ECN-capability. Thus, statistics describing the distribution of ECN-enabled routers SHOULD be part of future service level agreements.

5 Forwarding of ECN for multicast

To the author's knowledge there are no known research papers let alone proposals in the IETF, specifically on multicast congestion control using ECN. Informal discussions in the research community have only recently started on this subject. A brief, provisional summary of the relevant state of the art from these discussions is given below.

The issue with multicast and ECN solely concerns multicast duplication of the ECN field. Multicast active queue management will be no different to unicast — being dealt with at the egress interfaces of a multicast router, after multicast duplication and forwarding from the ingress.

With loss-based multicast congestion control, there

are two main arrangements for where the reaction to congestion occurs in a multicast group:

- Single rate:

  Sender takes account of all receivers :
  Each receiver feeds back congestion levels to the sender, with suitable controls on implosion, then the sender alters the rate of the group taking all feedback into account. Such approaches tend to suffer from the loss path multiplicity problem, finding more bottlenecks as the group size scales, and consequently causing the rate to 'drop to zero' [2]. Hence the next approach is preferred over this;

  Sender chooses representative receiver :
  Each receiver feeds back congestion levels to the sender, with suitable controls on implosion, then the sender nominates one receiver (typically the one that would run the slowest independent unicast session). This 'acker' runs a tight rate control feedback loop with the sender [14];

- Multi-rate:

  Receiver vary rate independently :
  The sender may arrange for data to be spread across multiple multicast groups with essential data in the 'base' group, slightly less essential data in a second and so on (layering). Each receiver may then independently leave the least essential groups while remaining joined to the rest until the point where congestion on their leg is reduced to acceptable levels. This is termed receiver-driven layered multicast (RLM [10]);

Currently, multicast duplication doesn't treat any fields in the header distinctively. It is often assumed that the ECN field should simply be duplicated in this way to every egress interface at a multicast router. However, there is concern that simple duplication would multiply the level of congestion seen by the session. This would result in as much congestion marking arriving at receivers as for multiple unicast flows. Where each receiver independently varies its rate (multi-rate), each misses out on the benefit it should derive from joining a multicast group. A multicast group should share the congestion it imposes on competing flows across its membership.

Due to this concern, it has been informally proposed (by Kelly) that when a marked packet is duplicated, all but one randomly chosen copy at each router is reverted to unmarked. The random choice is made for each packet arrival. Unmarkable and unmarked packets are duplicated unchanged, of course. For brevity, we will term this proposal 'randomly selected ECN'. The advantage of such an arrangement is that each congestion event is notified to a single receiver. Of course, implementation would be slightly more complex than simple duplication.

When used in multi-rate schemes, randomly selected ECN tends to treat multicast fairly with respect to unicast. However, problems surface if it is used for single-rate schemes. In simple single rate schemes based on randomly selected ECN, if feedback to the sender is triggered on the arrival of each congestion mark, the scheme still suffers from the loss path multiplicity problem. Selection of a representative receiver is the current preferred way to solve this problem. However randomly selected ECN results in such a low rate of marking at any one receiver that it would be very slow to converge on a suitable choice of acker. The round trip time of each feedback message varies dramatically, but has a mean value of all the congested paths weighted by the congestion on each. Therefore, over time, more marks arrive at receivers closer to bottlenecks. But it takes a lot of time for a large group.

It appears that, if a single rate is in use, simple duplication of ECN marking would be more useful, giving richer information to each receiver. Where rate control is co-ordinated by the sender (single rate), allowance can be made for duplication of the marking in the downstream direction during aggregation of congestion feedback in the upstream direction. It is 'only' necessary for the level of aggregation to mimic the tree topology, whether exactly or approximately.

Therefore it appears that the two types of congestion control scheme require different multicast duplication of the ECN field. Rather than require hosts to control multicast duplication, we propose a third 'hybrid ECN duplication' technique. In this hybrid scheme, when a marked packet is duplicated, all but one randomly chosen copy at each router is changed to be 'potentially marked', denoted by the remaining unused code-point (ECT=0, CE=1) (Fig 3). The random decision is made for each new packet. Unmarkable, unmarked and potentially marked packets themselves would all be duplicated unchanged. With this hybrid congestion notification, members of a group could extract the information they needed for either the single-rate or the multi-rate approaches. This would avoid having to add a signalling mechanism to request the network to choose one or the

other approach, also saving having to secure the signalling. Implementation would be slightly more complex again, of course.

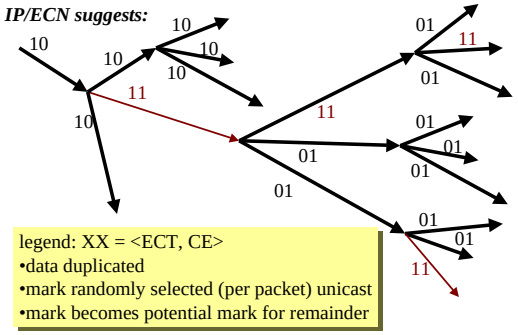

Figure 3: multicast forwarding of ECN

If the hybrid scheme were used, we would have to redefine our definitions of terms in section 3, as follows:

- markable (ECT=1, CE=*) or (ECT=0, CE=1);
- unmarkable (ECT=0, CE=0).
- marked (ECT=1, CE=1);
- potentially marked (ECT=0, CE=1);
- unmarked (ECT=1, CE=0) or (ECT=0, CE=1);

Active queue managment would be as before, with markable packets being chosen to be marked. Of course, the implication is that potentially marked packets might be changed to marked packets (ECT=0, CE=1) -> (ECT=1, CE=1) if they hit congestion more than once.

[13] suggests three alternative uses for the extra code-point we require for our hybrid ECN duplication scheme (ECT=0, CE=1):

1. Some other non-ECN-related function;
2. ECN-capable but for alternative semantics to the marked code-point (e.g. 'slightly' marked).
3. The extra code-point could be given an identical meaning to the marked code-point so that the two could be alternated randomly throughout a flow depending on a nonce at the sender, allowing the sender to detect 50% of any changes along the path from marked packets to unmarked;

At least for multicast, our proposal rules out categories 1 and 2, which we assume were speculative anyway. For instance, one would imagine that an alternate semantic (e.g. slightly congested) could be implied by a lower marking rate. It is likely that if either of these schemes was needed for unicast, it would also be needed for multicast.

Our scheme is compatible with the nonce scheme (guessing the details which haven't been published yet). The motivation for the nonce scheme is primarily for the sender to detect receivers that under-report congestion feedback. This assumes large senders may wish to act as policers on behalf of the network (their incentive may not be a natural one). It only works with positive acknowledgements (acks) where the sender can compare the ECN field in the ack with that it sent. The sender accepts that the network destroys the nonce information when it marks a packet, so nacks would not be comparable. To avoid implosion, multicast feedback schemes never use acks. So only nacks are seen by the sender. Therefore a multicast sender might as well originate all packets as (ECT=1, CE=0). And fortunately, in our scheme, the network treats (ECT=0, CE=1) as effectively unmarked when it arrives at congestion downstream of previous congestion.

A multicast sender could *even* arrange to use the nonce scheme in conjunction with our multicast duplication scheme. The motivation might be that some multicast congestion control schemes involve at least one receiver giving ack feedback (e.g. pgmcc [14]). The sender would guarantee an equal ratio of (ECT=1, CE=0) to (ECT=0, CE=1) over a moving window of n packets. It would affect say n/2 randomly selected packets with the nonce and insert padding into the remainder to balance the nonce packets. The multicast router would behave no differently from the description above. Each receiver could then detect the difference between the number of (ECT=1, CE=0) and (ECT=0, CE=1) packets over a moving window of n packets. This would be the level of 'potentially marked' traffic. This level would be fairly slow to emerge, and noisy if there were losses too[2]. But it may be enough to decide to drop a layer (in for example RLM [10]) or select the 'slowest' receiver (in for example pgmcc) both of which have some hysteresis anyway.

Until this last code point is defined, it is advisable for

[2] n would have to be large enough for there to be a high chance of more than two marks in any n packet window. The sender may have to adapt n and re-announce it depending on current feedback, which in itself is a potential security flaw if not done carefully.

implementations of ECN on both hosts and routers to avoid optimisations that would make it difficult to treat the two bit ECN field as four code points.

The uncertainty over multicast duplication of the ECN field need not hold up standardisation of other aspects of ECN in the IP layer. The default behaviour of all existing routers is to dumbly duplicate the ECN field along with the rest of the packet. Whatever the status of the rest of the ECN standardisation effort, simple duplication of the ECN field on multicast routers SHOULD be considered experimental.

6 Anycast forwarding of ECN

Anycast forwarding of the ECN field is no different from unicast.

7 ECN service to higher layer protocols

The IP service layer provides the following three ECN services to any upper layer protocol:

- The data sender MAY request that the packet is treated as markable by the IP layer. Nodes on an end-to-end path MAY honour such a request. If any node on the path cannot honour the request, it silently services the packet as if it were unmarkable.
- The IP layer forwards a request to treat a packet as markable without alteration (with the exception of congestion control proxies - see section 11).
- The IP layer only notifies receiving hosts of congestion experienced by each markable packet through the average marking rate apparent in a flow. The total congestion experienced is roughly the sum of congestion experienced at nodes along the packet's path. There is no guarantee that all or even any nodes on the path will be capable of contributing to this signal. The average marking rate is the ratio of marked packets to the total number of packets in a sample period. The meaning of a certain average marking rate and the sample period are defined by the marking behaviour in the service definition relevant to the packet's diffserv code-point[3].

[3]Of course, the host may operate a congestion control algorithm that tends to respond to the average marking rate without directly calculating it (e.g. [13]).

The IP layer offers these three services to all higher layer protocols, whether or not they use them, simply to avoid having to inspect the protocol field of the IP header to establish whether the ECN service is appropriate. Thus, any higher layer protocol MUST be able to assume these services will be available to it, whatever protocol it is. Of course, this is aside from any access control to this service interface on the host, which may deny access to a capability of this interface dependent on the user running the higher layer protocol.

8 Host congestion control algorithms for ECN

All new or updated congestion control protocols standardised through the IETF SHOULD state their applicability for markable as well as unmarkable packets.

The ECN specification for TCP/IP [13] stipulates that the congestion control algorithm followed by an ECN-capable data receiver on receipt of a marked packet must be essentially the same as that following a dropped packet.

As discussed in section 4.2, router and host algorithms are mutually dependent but need not be cast in stone. The point of equivalence between behaviour for markable and for unmarkable packets on a router will reflect that on a host. If markable traffic is marked at a router when unmarkable traffic is dropped, a mark should be treated like a drop at a host. If on the other hand the two types of traffic are both dropped in the same circumstances at the router, a drop for one should be treated like a drop for the other at the host. It was argued that this latter example seemed likely to be a useful one. A suitable wording for standardisation was given in that earlier section, which would allow room for experimentation across different service treatments.

9 Host requirements for ECN

Where a host protocol layer does not implement congestion control (e.g. UDP), it SHOULD offer ECN services to higher layers that are equivalent to those defined in section 7 for the IP layer. Specifically, a sending protocol SHOULD honour requests to send markable datagrams; and a receiving protocol should

allow higher layer protocols to determine whether received datagrams were markable and to determine whether each is marked.

Note that a markable packet is generally a signal to the *network* to enable ECN behaviour. As new congestion control protocols are defined, it is possible this signal to the network will be overloaded as an end-to-end signal from the data sender to the data receiver to request ECN behaviour. Because multicast protocols generally have to support 'late join', it is likely that data receivers may need to determine whether any datagram in a flow is markable. Due to these general requirements, a receiving application MUST be able to determine whether any arriving datagrams is markable.

However, future ECN-based congestion control protocols MUST NOT use markable packets before ECN-capability has been established. The only exception would be if the protocol were designed to ensure congestion control worked correctly even if such a marked packet arrived at a non-ECN-capable receiver.

Until a use for the (ECT=0, CE=1) code point is defined, host implementations of ECN SHOULD be able to request and to pass on any of the four code-points of the ECN field, rather than just each flag (bit) separately.

Whether hosts SHOULD or MUST implement an ECN version of each particular congestion control protocol (e.g. TCP) is not the concern of this document, which only covers aspects of ECN common to all protocols over IP.

10 ECN and fragmentation

For IPv4, markable traffic MUST have the don't fragment (DF) flag set. Setting the DF flag and using path maximum transmission unit (MTU) discovery [11] is current best practice anyway [9]. Hence it is not a problem to mandate its use with a new feature of IP. This is not an issue for IPv6, where there is no DF flag because not fragmenting is the only supported behaviour.

The rationale for not allowing fragmentation when ECN is enabled is to avoid complications on re-assembly of fragmented datagrams. Some fragments could be marked and others not, making it necessary to decide the marking of the re-assembled datagram before passing it to the congestion control protocol. To use the logical OR of the marking of all fragments might be a pragmatic solution, particularly for congestion control protocols like TCP where one loss per round trip is treated identically to many. However, it is becoming more common to see large numbers of packets per round trip time as data rates increase while packet sizes and the speed of light haven't increased for many years. Therefore it is to be expected that newer congestion control protocols might take more accurate account of the number of packets marked in a round trip. Hence, the inaccuracy of a logical OR during re-assembly at the IP layer is best avoided. A logical OR would also confound the accuracy of congestion avoidance charging [8], if it were shown to be necessary.

If an IPv4 packet contains a markable code-point but does not have the DF flag set (an illegal combination), it SHOULD be silently forwarded unless fragmentation is required. If fragmentation is required, an ECN capable router MUST discard it and return an ICMP Destination Unreachable error to the data sender. It MAY contain a code meaning "fragmentation needed and DF set". Alternatively it SHOULD contain a new ICMP code meaning "fragmentation needed but markable code-point used". If such an illegal datagram reaches the data receiver in fragments (perhaps due to a non-ECN-capable router or due to a bug in a node on the path), the receiver MUST discard the datagram and return a similar ICMP message to the data sender, as this may imply an unknown upstream problem. If such an illegal IPv4 datagram arrives at the data receiver intact, there is no need to take corrective action. The datagram should be silently handled in the normal fashion.

11 Access to the ECN field

This section clarifies exactly what types of node are expected to read or write the flags in the ECN field.

In [13] it has been proposed or implied that:

- the ECT flag SHOULD be set by the data sender if it has been established that all ends have ECN capability;
- routers MAY read ECT (we cannot say MUST, because not all routers will be ECN-capable) but MUST NOT alter it;
- whether the data receiver should read ECT once a session is in progress depends on the transport

protocol in use[4].

Also, it has been proposed or implied that:

- the CE flag SHOULD be clear when it leaves the data sender (excepting for random security checks);
- routers MAY set CE but MUST NOT clear it;
- the receiving host MAY read CE (it may not be ECN-capable), but certainly MUST NOT alter it.
- if the receiving host has enabled an ECN-capable session, it MUST read CE during that session;

This is, of course, quite apart from the discussions on what each node could do, if it chose to misbehave.

We wish to differ on the implied rules concerning what an intermediate node might be allowed to do to these flags. Our goal is to allow future flexibility where there is no reason not to.

The rules on changing the ECT flag at an intermediate point have not been explicitly stated, except in the context of tunnels, which we will discuss presently. Therefore, we will now propose rules for changing the ECN capability of a packet at intermediate nodes, in the most general form we can.

An unmarkable code-point MUST NOT be changed to a markable one by an intermediate node unless that node is able to control congestion on behalf of the data sender in response to ECN signalling and it has established that a downstream node has an ECN-capable transport (sender congestion control proxy).

Changing a markable code-point to unmarkable turns on drop behaviour in downstream routers. This capability may be used by a policer to 'punish' packets outside a contracted or reserved profile. Such packets would no longer be protected by ECN capability, so would be dropped while other packets within profile would merely be marked.

Changing a markable code-point to unmarkable would not generally disable ECN at the data receiver, as it is expected that the markable code-point depends on ECN capability, not the other way round. However, the markable code-point MAY be used as part of the negotiation of ECN capability between data sender and receiver in future congestion control protocols. If the network happened to change a packet being used for this negotiation from markable to unmarkable, this might result in ECN being disabled for a whole session.

An intermediate node MUST NOT change *all* packets with a markable code-point to unmarkable unless it is either able to handle ECN signalling on behalf of the data receiver (receiver congestion control proxy) or has arranged to reinstate the markable code-point with a node further downstream (effectively a limited functionality tunnel).

Congestion control proxies may help with the introduction of ECN into the core of the network, even where hosts are not ECN capable. A proxy to transform an intserv reservation at one or many ends of a flow into ECN behaviour in the core has been proposed in [1]. If appropriate, such proxies SHOULD ensure account is taken of the reduction in path length they have introduced.

To recap the position stated in [13] concerning the ECT flag and tunnels, a markable code-point MUST only not be copied to the active outermost header of a packet at tunnel ingress if it has also been arranged to reinstate it at tunnel egress. If the full-functionality tunnel behaviour is the considered normal, this constraint on limited functionality tunnels is effectively a specific case of the above rule concerning changing markable to unmarkable.

12 Security considerations

Authentication of the ECN field depends on whether it is treated as two flags or four code points. This further depends on whether the last undefined code-point (ECT=0, CE=1) is defined to relate to marking capability or to marking itself. Therefore authentication will not be discussed in this draft until the fate of this last code-point is clearer.

Firewalls SHOULD NOT discard packets simply because the ECN field has a non-zero value. In the past, while the currently unused (CU) field of the diffserv field (which phrase includes its previous uses) was truly unused, some firewalls treated any non-zero values as suspicious and discarded such packets.

Note that the requirement in [13] for ECN to be backward compatible is not met for 'simple tunnels'. This is because tunnel end-points MUST implement either

[4] in TCP the data receiver MUST still read ECT once a session is in progress (even though ECN capability has been negotiated for the session, some acks will be for re-transmitted packets), and in other transport protocols a data receiver MAY be required to read ECT to determine the ECN capability of the session at any point in a session (e.g. to cater for late joins).

the limited or the full functionality options, neither of which is the case with a simple tunnel.

Security is also the main subject of section 11. It is also discussed in subsection 4.2 and section 5 where fairness and incentives to use congestion avoidance are considered.

13 Further work

Considerable further research is required to establish the need for the 'potentially markable' ECN code-point for multicast duplication.

Once the fate of the fourth code-point is decided, authentication can be finalised.

Feedback is particularly requested concerning the relative merits of a new ICMP destination unreachable code (section 10), rather than overloading an old one. The argument for taking the approach adopted is that the purpose of an error message should be to identify the error, not identify that one of two errors has occurred. It is assumed that legacy host implementations will report an ICMP error code that is unknown to them opaquely, but such an assumption may be dangerous.

The approach to fragmentation in section 10 effectively gives IPv4 another set of code points for markable datagrams with DF=0, as long as path MTU discovery has been done. However, the extra space is fairly useless, as the DF flag should remain set during a session to allow discovery to detect changes to the path MTU involving non-ECN capable routers. The extra IPv4 code-point will be slightly more useful as a greater proportion of Internet routers become ECN-capable. No such extra code-point is possible with IPv6.

Many of the proposals in this document have not undergone a full security analysis to check for new denial of service threats, etc.

14 Conclusions

This document includes the necessary words to ensure that interactions with more aspects of the IP layer have been specified than in previous Internet drafts. It is believed that every aspect of this document is additive to [13]. The ability to define new marking behaviours and new host behaviours has been added using the diffserv architecture. This has been achieved without affecting the behaviours already defined for TCP. Similarly, a forward looking approach to fragmentation has been defined.

A stake has been placed in the ground warning that multicast duplication of ECN may not be as straightforward as some believed, and allowing room for experimentation.

Finally, requirements have been set to ensure that all new standardisation work will promote the use of ECN in preference to loss as a congestion signalling mechanism.

15 Acknowledgements

Arnaud Jacquet (BT), Sally Floyd (ACIRI), David Black (EMC) and Martin Karsten (TU Darmstadt) each for their help and constructive review comments.

References

[1] Ragnar Andreassen (Ed.). M3I; Requirements specifications; reference model. Deliverable 1, M3I Eu Vth Framework Project IST-1999-11429, URL: http://www.m3i.org/, July 2000.

[2] Supratik Bhattacharyya, Don Towsley, and Jim Kurose. The loss path multiplicity problem in multicast congestion control. In *Proc. IEEE Conference on Computer Communications (Infocom'99)*, URL: http://www.ieee-infocom.org/1999/papers/06c_04.pdf, March 1999.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. Request for comments 2475, Internet Engineering Task Force, URL: http://www.ietf.org/rfc/rfc2475.txt, December 1998.

[4] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang. Recommendations on queue management and congestion avoidance in the internet. Request for comments 2309, Internet Engineering Task Force, URL: http://www.ietf.org/rfc/rfc2309.txt, April 1998.

[5] Scott Bradner. Key words for use in RFCs to indicate requirement levels. BCP 14, Internet Engineering Task Force, URL: http://www.ietf.org/rfc/rfc2119.txt, March 1997. (RFC 2119).

[6] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[7] Richard J. Gibbens and Frank P. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35, 1999.

[8] Frank P. Kelly, Aman K. Maulloo, and David K. H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49, 1998.

[9] C. Kent and J. Mogul. Fragmentation considered harmful. In *Proc. SIGCOMM '87 Workshop on Frontiers in Computer Communications Technology*, August 1987.

[10] Steven McCanne, Van Jacobson, and Martin Vetterli. Receiver-driven layered multicast. *Proc. ACM SIGCOMM'96, Computer Communication Review*, 26(4), October 1996.

[11] J. Mogul and S. Deering. Path MTU discovery. Request for comments 1191, Internet Engineering Task Force, URL: http://www.ietf.org/rfc/rfc1191.txt, November 1990.

[12] K. Nichols, S. Blake, F.Baker, and D. Black. Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers. Request for comments 2474, Internet Engineering Task Force, URL: http://www.ietf.org/rfc/rfc2474.txt, December 1998.

[13] K. K. Ramakrishnan, Sally Floyd, and David Black. The addition of explicit congestion notification (ECN) to IP. Internet draft, Internet Engineering Task Force, URL: http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-ecn-01.txt, January 2001. (Work in progress) (expires Jul 2001).

[14] Luigi Rizzo. pgmcc: A TCP-friendly single-rate multicast congestion control scheme. *ACM SIGCOMM Computer Communication Review*, 30(4):17–28, September 2000.

[15] IETF secretariat. Transport area working group (tsvwg). Working group charter, Internet Engineering Task Force, URL: http://www.ietf.cnri.reston.va.us/html.charters/tsvwg-charter.html, Continuously updated.

Document history

Version	Date	Author	Comments
A	23 Feb 2001	Bob Briscoe	Internet Draft on which this is based published
B	29 Dec 2001	Bob Briscoe	Started to removed I-D specific material and added figures.