Position paper for ISOC Workshop on Reducing Internet Latency

ECN & Early ECN Marking

Michael Welzl, Gorry Fairhurst, Naeem Khademi

End-to-End latency is a becoming key performance metric, and is increasingly more relevant to users than "headline" performance metrics, such as "speed". There are a range of factors that impact latency, but this paper focuses on router Active Queue Management (AQM), and more specifically on enabling use of Explicit Congestion Notification (ECN).

Introduction

Router buffering can significantly impact the end-to-end latency observed across the Internet. Router queues grow when the rate of incoming traffic exceeds available capacity. This leads to congestion, and ultimately packet loss. By its nature, TCP congestion control needs to create congestion to discover the available capacity and then react by changing its transmission rate. How much congestion is needed depends on the type of feedback signal: with a FIFO DropTail queue and standard loss-based TCP, a queue needs to overflow or an AQM mechanism needs to decide it is time to send a congestion signal (drop a packet).

Typical AQM methods can reduce the size of router queues. Most methods react to congestion, rather than seeking to avoid inducing latency. A key concern for bulk applications is throughput across the network, but for other applications there may be a different trade-off between loss, latency and throughput (e.g. interactive video sensitivity to loss and latency). Although AQM can reduce latency, the original use-case for AQM was different – and has lead to patchy appreciation of the incentives for ISP deployment. Awareness of "bufferbloat" is increasing understaning of the benefits of using AQM.

Need for enabling ECN

Theoretically, AQM could be much better when an explicit signal, such as Explicit Congestion Notification (ECN), is present. This avoids the requirement that a transport/application has to experience loss to discover congestion. ECN allows routers to mark packets *earlier* than the point where packets would normally be dropped by AQM, e.g., right at the point where capacity is first exceeded. Responding to ECN marks could cause senders to back-off earlier and reduce queuing delay.

So why do we not see wide-spread deployment of ECN? ECN relies both on ISPs enabling AQM to mark packets and on enabling the user stack to process ECN markings. Most mainline operating systems now implement ECN, but while being enabled at the server-side, this is typically disabled in clients. Awareness of the issues incurred by latency helps provide an important incentive for users to enable ECN, if network operators deploy ECN-enabled routers.

We assert that now is the time to consider seriously an ECN roadmap to deployment, because this can directly leverage the benefis of AQM for latency-sensitive applications. In doing so, we need to plot an incentive-compatible way forward to affirm user and network benefits from ECN. As in AQM, ECN needs to seek solutions that maximize performance benefits across different kinds of applications, without incurring penalties in explicit configuration.

A proposal for building better ECN

The incentives for ECN deployment will be assisted by defining a marking method that potentially improves and never degrades the performance experienced by users, while reducing the queuing delay (latency) in routers.

How and how early should AQM mark? RFC 3168 says that AQM should only mark packets as Congestion-Experienced (CE) when a router would drop them in the absence of ECN. This ensures compatibility between ECN-capable and non-ECN-capable flows: if an AQM router would mark ECN-capable traffic earlier than it would drop non-ECN-capable traffic, the ECN-capable flows would receive lower throughput than a non-ECN-capable flows. This present requirement does not encourage use of ECN.

We assert that routers should be encouraged to support an earlier ECN marking, in a method we call Early ECN (EMM). This can provide significant benefit, with obvious performance improvement for applications that are both loss and latency-sensitive such as video. If early marking (based on low latency thresholds) is not the default method – assuming that some flows want a behaviour closer to current Internet trade-offs, then we need a way to introduce an early marking method.

We suggest considering exploring a change to existing standards with the following requirements:

1. Routers become able to identify packets from an "new" ECN-capable transport and early-mark only such packets.

2. If both types of traffic share the same queue, "new" ECN-capable traffic should exhibit a more aggressive form of transport congestion control to compensate for the presumed early ECN-marking. For instance, in response to a "new" CE signal, instead of halving the sending rate, a new ECN-capable TCP could reduce it by a smaller amount.

Item 2) is an open transport area research question. *Item 1)* could be addressed by re-defining the ECT(1) codepoint. ECT(1) was introduced in RFC 3168, with one use to support the ECN nonce. This nonce is only useful if it is deployed in at both an ECN sender and receiver, and has not seen widepread deployment. ECT(1) has also been used to provide alternate ECN markings. We suggest to use the ECT(1) codepoint to let a "new" ECN-capable sender signal an EEM-capable router (on a side note, it is possible to realise a weaker variant of the nonce with the mechanism presented here, but we skip these details for the sake of brevity). The CE mark remains the same as with standard-ECN.

Backward compatibility

Methods have been proposed that provide flow-isolation – reducing the impact of flows sharing a common queue with other flows that have different application goals. Diffserv provides one architecture for doing this, there are also AQM-variants that attempt to isolate flows. Combining this with AQM and ECN could prevent bulk flows impacting latency (or loss) sensitive flows. Mechanisms are also needed to isolate traffic from flows that do not appropriately respond to CE-markings or loss, to prevent increasing the latency for other flows.

However, several questions arise when considering EEM in a network with a mix of routers and end hosts that may or may not support EEM and cannot separate the traffic flows:

- Should we keep separate virtual queues for standard-ECN and EEM flows?

- If standard-ECN and EEM flows share an EEM-enabled router queue, how should we tune the EEM sender's congestion response to be compatible with standard-ECN flows?

- If standard-ECN and EEM flows share the same queue in a traditional ECN router, EEM flows may be marked the same as standard-ECN flows and then push these flows aside – but is this bad?, or can it motivate deployment? e.g., is it worse than current behavior with CUBIC vs. standard-TCP, or would it just give users the right incentives to upgrade their systems?

Table 1 provides an overview of all combinations of "old" and "new" hosts and routers, assuming that traffic from "old" and "new" end systems share a common router queue.

Conclusion

We have looked at the motivation to encourage deployment of ECN from two perspectives, noting the opportunity to introducs ECN when deploying ACM. We note that using ECN in hosts and routers can yield significant benefit for latency-sensitive applicatons. The additional infromation provided by ECN feedback can enable a range of applications to derive useful congestion feedback, rather than relying soley on TCP-like loss-based congestion detection. We also motivate the need for research to develop techniques for early ECN marking. This can substantially improve the latency experienced by applications when congestion is experienced. We suggest that ECN methods can emerge that will directly support latency-sensitive applications.

This work was funded by the European Community under its Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project (ICT-317700). The views expressed are solely those of the author(s).

Table 1. Summary of possible deployment combinations for 2 TCP flows

Router Type	*No AQM*	*Old AQM* *ECT(0)*	*New AQM* *ECT(1) +ECT(0)*
Sender Type			
Old (no ECN) + Old (no ECN)	Default behavior	Default AQM behavior	Default AQM behavior
Old (no ECN) + Old (ECT(0))	Default behavior	Default AQM behavior and saves some packets for ECN	Default AQM behavior and saves some packets for ECN
Old (no ECN) + New (ECT(1))	Default behavior	Default AQM behavior and, if router supports ECT(1): saves some packets for ECN, ECT(1) flow more aggressive	Default AQM behavior for old flow, early marking for new ECT(1) may need to be aggressive to compensate
Old (ECT(0)) + Old (ECT(0))	Default behavior	Default AQM behavior and saves some packets for both flows	Default AQM behavior and saves some packets for both flows, may be almost equal
Old (ECT(0)) + New (ECT(1))	Default behavior	Default AQM behavior and, if router supports ECT(1): saves some packets for ECN, ECT(1) more aggressive	Default AQM behavior for old flow, early marking for new ECT(1), may need to be aggressive to compensate
New (ECT(1)) + New (ECT(1))	Default behavior	If router supports ECT(1): new, aggressive host reaction for both flows; else, default behavior.	EEM behavior

Table Terminology
Sender: 1. Old (no ECN): TCP flow with ECN disabled 2. Old (ECT(0)): TCP flow with standard ECN enabled 3. New (ECT(1)): TCP flow with Early ECN Marking (EEM) enabled
Router: 1. No AQM: FIFO/DropTail 2. Old AQM ECT(0): Any AQM with standard CE marking on ECT(0) packets
New AQM ECT(1)+ECT(0): A hypothetical AQM with EEM-marking on ECT(1) packets and standard CE marking on ECT(0) packets. When exactly these markings should happen is an open research question.