Support for ECN and PCN in MPLS networks

draft-davie-ecn-mpls-01.txt

Bruce Davie
Cisco Systems

Bob Briscoe
June Tay
BT Research

Problem Overview

- ECN (RFC 3168) encodes 3 states in 2 bits
  - ECT, not ECT, CE
  - ECN nonce uses up the extra codepoint
- MPLS header has only 3 bits (EXP field) suitable for this purpose
- EXP values widely used for Diffserv
- Even stealing one bit for ECN would be tough

Overview of proposal

- Don’t define a bit, use a codepoint (or 2)
  - Given < 8 codepoints in use, can add ECN capability for any single PHB by using one more codepoint
  - “Original” codepoint means “PHB X, not-CE”, new codepoint means “PHB X AND CE”
- Handle ECT at egress
  - If IP header is ECT: Copy MPLS CE state to IP header
  - If IP header is not-ECT: drop packet if MPLS EXP codepoint is CE
- Permissive approach
  - Other uses of EXP permitted

Changes in new (-01) version

- Remove dependency on PCN
  - There as an example only
- Corrected reference to [Shayman00]
  - Our encoding proposal quite similar to his
- Copying ECN information to exposed header on egress (pop) is not mandatory
- Crossing from ECN-enabled to ECN-disabled domain is addressed
- Typos, nits

Summary

- Increased interest in ECN (& PCN) combined with widespread use of MPLS & Diffserv, motivates a solution to ECN support in MPLS
  - Real impediment to ECN deployment otherwise
- One extra codepoint is enough for ECN
- Approach is consistent with prior ECN-MPLS drafts and with RFCs 3168 (ECN) and 3270 (MPLS-Diffserv)
- TSVWG seems appropriate home for this draft
  - Needs ECN expertise
  - ECN deployment benefits from draft

Backup Slides

Issues addressed in -01

- When leaving an MPLS domain, we don't insist the ECN information be propagated back to IP header
  - You could imagine using ECN to control congestion purely in the MPLS cloud - this is up to the operator
- When crossing from ECN-enabled domain to ECN-disabled domain, need to check the ECN state and drop if packet is not-ECT AND congestion-marked
  - This implies peeking below MPLS label at an MPLS-labelled interconnect point

Prior Work

- Floyd, Ramakrishnan & Davie,1999
  - draft-ietf-mpls-ecn-00.txt
  - Encoded 3 states in 1 bit (!) by overloading Not-ECT and CE
  - Would drop ECT packets that experienced congestion marking twice
- Shayman, 2000
  - draft-shayman-mpls-ecn-00.txt
  - Encodes only CE state in EXP (hence may mark non-ECT packets)
  - Figures out the “right thing” at egress
  - Adds explicit signaling from egress to ingress
- RFC 3270
  - Defines usage of 3-bit MPLS EXP field for Diffserv
  - Does not preclude other uses of the field

Example

- Suppose we want to add ECN to just one PHB (e.g. a “premium” data class, AF11)
- Suppose EXP=010 is used for AF11, and that EXP values of 000, 001, 100 are in use for some other PHBs
- We add ECN support to AF11 traffic only, defining EXP=101 to be the “CE” codepoint for AF11
- Encaps/decaps rules on next slide:

Example (cont.)

Ingress (push)

IP	MPLS
AF11 & $\overline{\text{ECT}}$	010 ($\overline{\text{CE}}$)
AF11 & ECT	010 ($\overline{\text{CE}}$)
AF11 & CE	101 (CE)
Not AF11	See RFC 3270

Egress (pop)

MPLS	IP (in)	IP (out)
010 ($\overline{\text{CE}}$)	Any	IP(in)
101 (CE)	ECT	CE
101	$\overline{\text{ECT}}$	drop
101	CE	CE
Other EXP	Any	See RFC 3270

- In this example, 010 is the “Not CE” codepoint and 101 is the “CE” codepoint and all other codepoints/PHBs do not support ECN
- Note that ECN nonce propagates through the MPLS domain

Deployment

- Can create an ECN-enabled MPLS domain by enabling ECN-aware push/pop behavior at ingress/egress
  - All ingress/egress routers MUST be enabled before any ECN core behavior is enabled
- ECN behavior can be added one core router at a time

Tunneling & RFC3168

- Subtle difference between this draft and “full functionality” tunnel mode of RFC3168
  - RFC3168 does not copy CE state to outer header at ingress; this draft does
- We prefer to copy CE state to enable marking that depends on current state (useful for PCN)
- Authors of 3168 agree it makes no difference for ECN
  - If you don’t like copying info to outer header, don’t! (the limited functionality model)

PCN support

- Just like ECN, but more codepoints
- E.g. Add PCN to one PHB by allocating 3 codepoints to that PHB
  - Not marked (NM)
  - Admission-marked (AM)
  - Pre-emption marked (PM)
- Rules for pushing/popping headers are similar to ECN