An open ECN service in the IP layer

19 Mar 2001
Bob Briscoe, BT & UCL
Jon Crowcroft, UCL

M3I - Market Managed Multi-service Internet
IST Project No 11429 under the
EU Vth Framework Information Society Technologies Programme

motivation

• Q: why add to ECN at this late stage?
• A: ensure space for ECN research
  (A2: + clarifications for implementors)

• fully support ECN to standards track ASAP
• deeply grateful for many years of work behind this from KKR/SF/DB etc.
ECN in IETF tsvwg

• “TCP/ECN” I-D Ramakrishnan, Floyd, Black
draft-ietf-tsvwg-ecn-02.txt
  – “The Addition of Explicit Congestion Notification (ECN) to IP”
  – standards track (last call before proposed standard)

• “ECN nonce” I-D Wetherall, Ely, Spring
draft-ietf-tsvwg-tcp-nonce-00.txt
  – “Robust ECN Signaling with Nonces”

• “IP/ECN” I-D Briscoe, Crowcroft
draft-ietf-tsvwg-ecn-ip-00.txt
  – “An Open ECN Service in the IP layer”

“IP/ECN” status

• review comments on –01 of “TCP/ECN”
  • intended for incorporation in –02
  • not intended to go anywhere itself

• off-line discussions
  • digests on tsvwg list

• few of our words used in –02, but sufficient
  • we’re happy :–)

• 3 aspects where minor disagreement remains
  • ...agreed to “take to tsvwg”
  • otherwise ‘broadly’ happy with –02 as it stands
“IP/ECN” contents

- highlighted issues with “TCP/ECN” at the IP layer
  - code-points not bits ➔ standards track
  - diffserv interactions ➔ standards track
  - multicast interactions ➔ no conflict with stds track
  - other transport protocols than TCP ➔ a later RFC
    - IP ECN service interface
  - access semantics to ECN field ➔ a later RFC
    - congestion ctrl proxies
  - fragmentation interactions ➔ standards track

ECN code-points, not bits

- **TCP/ECN was:**
  - ECT = ECN capable transport
  - CE = congestion experienced

- **IP/ECN suggests:**
  - separate bits meaning nothing, only whole ECN code-point
    - unmarkable \(<ECT=0, CE=0>\)
    - markable \(<ECT=1, CE=*>\), \(<ECT=0, CE=1>\)
    - marked \(<ECT=1, CE=1>\)
    - unmarked \(<ECT=1, CE=0>\), \(<ECT=0, CE=1>\)
      - potentially marked = \(<ECT=0, CE=1>\)

- **TCP/ECN now agrees, but using own terminology**
buffer filling vs. starving
(background to ECN/diffserv discussion)

keep queue empty for low latency
keep queue full for high utilisation

from load to queue length

expected queue length 100%
marking probability 1

buffer starving
buffer filling

ave. output load/ %
marking probability

ave queue length 100%

fact

goal
### ECN mark/drop equivalence

<table>
<thead>
<tr>
<th>Buffer Type</th>
<th>non-ECN-capable</th>
<th>ECN-capable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td><img src="image1.png" alt="Graph" /></td>
<td><img src="image2.png" alt="Graph" /></td>
</tr>
<tr>
<td>Starving</td>
<td><img src="image3.png" alt="Graph" /></td>
<td><img src="image4.png" alt="Graph" /></td>
</tr>
</tbody>
</table>

- **ECN mark/drop equivalence**
- **1 probability drop average queue length**
- **“mark ≡ drop”**
- **1 probability drop average queue length**
- **“drop ≡ drop”**

### ECN interactions with diffserv

- **TCP/ECN – 01**
  - no explicit mention of diffserv marking behaviours
- **TCP/ECN – 02**
  - “mark ≡ drop” defined as default for all PHBs
  - if don’t want default...?
  - PHB definitions MAY include marking behaviour
- **clarification**
  - definition of marking behaviour
  - diffserv already provides framework
  - part of queuing behaviour (like discard behaviour)
  - per PHB
  - no change to who defines each: standards /operators
  - above statement in TCP/ECN updates *informational* diffserv architecture guidelines
**implementation advice**

**mark/drop equivalence**

- **TCP/ECN said “mark ≡ drop”**
  - decide to notify *then* decide how (by ECN capability)
  - embedded this assumption in implementation advice
- **IP/ECN has future-proofed implementation advice:**
  - may decide marking/discard *behaviour* by ECN capability
    - *then* marking & discard behaviours MAY be same
      - (e.g. for buffer filling behaviours)
    - “mark ≡ drop” doesn’t make sense for buffer starving
    - “mark < drop” & “drop ≡ drop” allowed
  - ECT code-points like a 2-state extension to DSCP

---

**ECN mark/drop equivalence**

- *default in “TCP/ECN” is sufficient for now*
- *except...*
  - where future research allowed, constraint needed:
    - within each PHB, definition of equivalence between marking and discard behaviours needs to be consistent
    - ...for all routers & host protocols using that PHB

- **if research shows value of buffer starving...**
  - ...take up in a diffserv w-g
multicast forwarding of ECN

**IP/ECN suggests:**

1. Data duplicated
2. Mark randomly selected (per packet) unicast
3. Mark becomes potential mark for remainder

**Legend:** XX = <ECT, CE>

- motivation
  - Duplicating congestion indication was incorrect, but unavoidable with loss-signalled congestion
- Congestion control protocol can choose meaning of ‘potential mark’ <ECT=0, CE=1>
  - Multi-rate schemes (e.g. layered multicast) treat it as unmarked
  - Single rate schemes (e.g. pgmcc) treat it as marked
- May not be necessary - research issue
- ECN nonce is compatible (see IP/ECN I-D)
  - No need to mention multicast in TCP/ECN stds track
**IP’s ECN service to layer 4**

- **“IP/ECN”:**
  - documents service interface that IP provides
  - not just for TCP
  - potentially for UDP, IGMP, ICMP, RSVP, RIP
- **“TCP/ECN” says nothing**
  - don’t want to encourage UDP/ECN anarchy until most routers are ECN-capable
- **“IP/ECN” forms basis of future RFC on this?**
  - silence won’t stop UDP apps using ECN-capable routers
  - banning contraceptive advice doesn’t prevent pregnancy

---

**UDP/ECN unsafe?**

- does “mark ≡ drop” give wrong incentives?
- “drop ≡ drop” gives ECN capable flows:
  - no delivery advantage (functional)
  - latency advantage (non-functional)
    - ...through network supporting co-operation

<table>
<thead>
<tr>
<th></th>
<th>non-ECN-capable</th>
<th>ECN-capable</th>
</tr>
</thead>
<tbody>
<tr>
<td>buffer starvag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>probability</td>
<td>drop</td>
<td>mark</td>
</tr>
<tr>
<td>probab.</td>
<td>drop ≡ drop</td>
<td>mark</td>
</tr>
<tr>
<td>ave queue length</td>
<td></td>
<td>ave queue length</td>
</tr>
</tbody>
</table>
ECN & IP fragmentation

- **IP/ECN says:**
  - IPv4 MUST set don’t fragment (DF) flag
  - best practice (path MTU discovery)
  - IPv6: don’t fragment is implicit

- **TCP/ECN −01 said nothing**

- **TCP/ECN −02 now says:**
  - TCP/IPv4 SHOULD set don’t fragment
  - if not set & fragments arrive, receiver uses logical OR

- **argument...**
  - SHOULD leaves doubt, so all implementers MUST add complex re-assembly code that will never be used
ECN & IP fragmentation solution

- **what “TCP/ECN” –02 says, another way:**

- **don’t fragment MUST be set...**
  - ...UNLESS the sending TCP knows the receiving IP will not ignore CE on any fragment
  - this document doesn’t describe negotiation of such a capability

- **old ECN implementations not compatible**
  - bug fix for something we didn’t notice

summary

- **we’re happy with standards track I-D as it is, but...**

- **3 wishes**
  1. add explicit guideline on marking/discard equivalence being consistent within a PHB
  2. define IP’s ECN interface to higher layers (soon)
  3. don’t fragment: best as a MUST...UNLESS

- **nothing worth fighting about**
- **what does the w-g think?**