ECN Background

- Explicit Congestion Notification (ECN) in IP [RFC3168]
  - Mark, not drop, packet at congestion onset
  - used extensively in data centres
    - usually for low queue delay with Data Centre TCP (DCTCP)
  - L3 switches mark the IP header
- Enduring incremental deployment problem
  - if legacy receiver or sender would not understand marking
    - congested buffer MUST drop not mark
    - otherwise legacy hosts would black-hole congestion signals
- Solution for e2e transports over IP (eg. TCP, RTP)
  - e2e ECN capability negotiation at flow set-up
  - IP-ECN codepoints for “ECN-capable transport” (ECT)
  - then congested buffers:
    - mark ECT packets or
    - drop Not-ECT packets

<table>
<thead>
<tr>
<th>IP-ECN codepoint</th>
<th>value</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not-ECT</td>
<td>00</td>
<td>Not ECN-capable transport</td>
</tr>
<tr>
<td>ECT(0)</td>
<td>10</td>
<td>ECN-Capable Transport</td>
</tr>
<tr>
<td>ECT(1)</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>CE</td>
<td>11</td>
<td>Congestion Experienced ('marked')</td>
</tr>
</tbody>
</table>
Adding ECN to TRILL: the problem

- Similar incremental deployment problem
  - if legacy egress does not understand ECN
  - will not propagate upward to forwarded IP inner header
  - would black-hole congestion signals

- Three potential types of solution:
  1. require TRILL R Bridges to act as “L3 switches” and mark IP-ECN field
  2. require ingress-egress ECN capability advertisement (via routing)
  3. use a critical ingress-to-egress flag

- rest of talk
  - #1: set aside as unreasonable
  - #2: described first and in [draft-eastlake-trill-ecn-support-00]
  - #3: described second but not in draft yet (developed after draft deadline)
Approach #2: Requires ingress-egress ECN negotiation

- **Non-Critical Hop-by-Hop Bits**

<table>
<thead>
<tr>
<th>ECN codepoint</th>
<th>value</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not-ECT</td>
<td>00</td>
<td>Not ECN-capable transport</td>
</tr>
<tr>
<td>ECT(0)</td>
<td>10</td>
<td>ECN-Capable Transport</td>
</tr>
<tr>
<td>ECT(1)</td>
<td>01</td>
<td>ECN-Capable Transport</td>
</tr>
<tr>
<td>CE</td>
<td>11</td>
<td>Congestion Experienced</td>
</tr>
</tbody>
</table>
Approach #2: ECN negotiated between ingress and egress

- Ingress RBridge
- Transit
- Egress RBridge
- End Station

Possible ECN marking

TRILL Header ECT marking if (a) traffic was marked as ECT & (b) ingress and egress support ECN

Congestion experience in TRILL path combined with tunneled ECN indication [RFC6040] Egress announced it would support this

"Tunnel"
Approach #3: Defer mark / drop decision to egress

Non-Critical Hop-by-Hop Bits

<table>
<thead>
<tr>
<th>Crit.</th>
<th>CHbH</th>
<th>NCHbH</th>
<th>CRSV</th>
<th>NCRSV</th>
<th>CItE</th>
<th>NCItE</th>
</tr>
</thead>
<tbody>
<tr>
<td>......</td>
<td>......</td>
<td>-------</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>-------</td>
</tr>
</tbody>
</table>

| C | C | C | Ext |
| R | R | R | Hop |
| H | I | R | Cnt |
| b | t | s | Ext |
| H | E | v | Cnt |

Critical Ingress-to-Egress Bits

<table>
<thead>
<tr>
<th>ECT codepoint</th>
<th>value</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not-ECT</td>
<td>00</td>
<td>Not ECN-capable transport</td>
</tr>
<tr>
<td>ECT(0)</td>
<td>10</td>
<td>ECN-Capable Transport</td>
</tr>
<tr>
<td>ECT(1)</td>
<td>01</td>
<td></td>
</tr>
<tr>
<td>CS</td>
<td>11</td>
<td>Congestion Signalled (Reserved for future use)</td>
</tr>
</tbody>
</table>

Congestion Experienced CE flag
Approach #3: Defer mark / drop decision to egress

Possible ECN marking

End Station

Ingress RBridge

Transit

Congested transit RBridge that supports ECN marks using CE flag (Critical Ingress-to-Egress)

No checking for ECT

Egress RBridge

End Station

Copy traffic ECN field to Non-Critical Hop-by-Hop TRILL ECN field (not necessary for ‘Classic’ [RFC3168] ECN, but needed for variants.

No ingress or transit dependence on Egress capabilities

Two cases:
1. Egress supports ECN decap, correctly combines outer TRILL ECN with inner IP ECN [RFC6040]
2. Egress doesn't understand ECN, drops any frame with a CltE bit set (Default behavior, which is desired)
Recap: ECN tunneling rules at egress

[ RFC6040 ]

- TRILL egress same as [ RFC6040 ] but 3 ECN bits in proposal #3 so map 3 bits to the 4 codepoints as shown in table:
Next Step

- Ask for comments on the mailing list
- Call for WG Adoption
SUPPLEMENTAL SLIDES
Recap: ECN tunneling rules at ingress [RFC6040]

<table>
<thead>
<tr>
<th>incoming header (also = outgoing inner)</th>
<th>outgoing outer</th>
<th>Compatibility mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal mode (default)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not-ECT</td>
<td>Not-ECT</td>
<td>Not-ECT</td>
</tr>
<tr>
<td>ECT(0)</td>
<td>ECT(0)</td>
<td>Not-ECT</td>
</tr>
<tr>
<td>ECT(1)</td>
<td>ECT(1)</td>
<td>Not-ECT</td>
</tr>
<tr>
<td>CE</td>
<td>CE</td>
<td>Not-ECT</td>
</tr>
</tbody>
</table>
Adding support to TRILL for Low Latency Low Loss Scalable throughput (L4S)

- for background on L4S see:
  - draft-briscoe-aqm-duaq-coupled, draft-briscoe-tsvwg-ecn-l4s-id

- On TRILL RBridge, use solution #3 “defer mark/drop decision”
  - plus: classify on NCHbH TRILL ECN field, as shown above

```
Classic queue:
if (p' > max(random(), random()) ) {
    mark(frame, CE(0))
} else mark(frame, CE(1))
```

```
L4S queue:
if (p' > random() ) {
    if (p' > random() ) mark(frame, CE(0))
    else mark(frame, CE(1))
}
```

- then deferring mark/drop decision to egress gives desired outcome
End

Donald E. Eastlake, 3rd
d3e3e3@gmail.com

Bob Briscoe
ietf@bobbriscoe.net