More Accurate ECN Feedback in TCP

tcpm – 84. IETF Vancouver – July 30, 2012

draft-kuehlewind-tcpm-accurate-ecn-01
draft-kuehlewind-tcpm-accurate-ecn-option-01

Mirja Kühlewind <mirja.kuehlewind@ikr.uni-stuttgart.de>
Richard Scheffenegger <rs@netapp.com>
Problem Statement

ECN provides only one congestion feedback signal per RTT ...
    as designed for current congestion control mechanisms
    that will react only once per RTT on congestion

... but new TCP mechanisms need to know how many congestion markings occurred exactly (number of marked packets/bytes).

Scope

• DCTCP: Decrease depended on number of ECN markings
• ConEx: Feedback on expected whole-path congestion
• Congestion control might react differently on ECN in future (ICCRG and LEDBAT ML)
1. Accurate ECN Feedback Option in TCP
   (draft-kuehlewind-tcpm-accurate-ecn-option-01)
   – TCP Option for accurate ECN feedback (AccECN) can provide more information
   – Can be used in addition to the classic ECN or the new scheme below
   – Not proposed as default mechanism because of
     • Middlebox issues with new TCP Options
     • Overhead in TCP header

2. More Accurate ECN Feedback in TCP
   (draft-kuehlewind-tcpm-accurate-ecn-01)
   – Mechanism to retrieve more accurate ECN feedback (more than one signal per RTT) by
     reusing the 3 ECN/ECN-Nonce bits in the TCP header
   – New version now proposes one specific scheme
1. Accurate ECN Feedback Option in TCP
   (draft-kuehlewind-tcpm-accurate-ecn-option-01)
   – TCP Option for accurate ECN feedback (AccECN) can provide more information
   – Can be used in addition to the classic ECN or the new scheme below
   – Not proposed as default mechanism because of
     • Middlebox issues with new TCP Options
     • Overhead in TCP header

2. More Accurate ECN Feedback in TCP
   (draft-kuehlewind-tcpm-accurate-ecn-01)
   – Mechanism to retrieve more accurate ECN feedback (more than one signal per RTT) by
     reusing the 3 ECN/ECN-Nonce bits in the TCP header
   – New version now proposes one specific scheme
Overview ECN in IP

Terminology from [RFC3168]

The ECN field in the IP header
- ECT(0)/ECT(1): either one of the two ECN-Capable Transport codepoints
- CE: the Congestion Experienced codepoint
TCP Accurate ECN Feedback Option Negotiation

<table>
<thead>
<tr>
<th>Kind</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
</table>

→ Only valid in a <SYN> or <SYN,ACK> during a three way handshake
  
  - MAY send the AccECN negotiation option in an initial SYN
    and MAY send a AccECN option (see next slide) in other segments only if it received this option negotiation in the initial <SYN> or <SYN,ACK>
  
  - MAY send an <SYN,ACK> segment with the AccECN negotiation option in response to a AccECN negotiation option in the <SYN>
  
  - MUST only negotiate for the AccECN option if ECN is negotiated as well

→ Further Option Space could be used to indicate
  
  - if an end point wants get AccECN feedback or not
  
  - how often the AccECN option should be sent (e.g. for at least every 10th data segment)
TCP Accurate ECN (AccECN) Feedback Option

Kind: TBD (same as above)
Length: 12 bytes

<table>
<thead>
<tr>
<th>Kind</th>
<th>12</th>
<th>ECT(0)</th>
<th>ECT(1)</th>
<th>CE</th>
<th>non-ECT</th>
<th>loss</th>
<th>CE in bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- SHOULD be included in every ACK
  
  To reduce network load the AccECN option MAY not be send in every ACK

- Counter wrap-aroounds
  
  - MUST send at least three ACKs/segments with the AccECN option between counter wrap around
  
  - Whenever an AccECN option is received with smaller counter value than in the previous one and the respective ACK acknowledges new data, a wrap around MUST be assumed
1. Accurate ECN Feedback Option in TCP
   (draft-kuehlewind-tcpm-accurate-ecn-option-01)
   - TCP Option for accurate ECN feedback (AccECN) can provide more information
   - Can be used in addition to the classic ECN or the new scheme below
   - Not proposed as default mechanism because of
     • Middlebox issues with new TCP Options
     • Overhead in TCP header

2. More Accurate ECN Feedback in TCP
   (draft-kuehlewind-tcpm-accurate-ecn-01)
   - Mechanism to retrieve more accurate ECN feedback (more than one signal per RTT) by reusing the 3 ECN/ECN-Nonce bits in the TCP header
   - New version now proposes one specific scheme
Overview ECN and ECN Nonce in TCP

Terminology from [RFC3168] and [RFC3540]

The ECN field in the IP header
- ECT(0)/ECT(1): either one of the two ECN-Capable Transport codepoints
- CE: the Congestion Experienced codepoint

The ECN flags in bytes 13 and 14 of the TCP Header

```
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|               |           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| N | C | E | U | A | P | R | S | F |
| S | W | C | R | C | S | S | Y | I |
| R | E | G | K | H | T | N | N |
```

- CWR: the Congestion Window Reduced flag
- ECE: the ECN-Echo flag
- NS: ECN Nonce Sum
Codepoint Coding for More Accurate ECN Feedback

- Use of these 3 header flags as one 3-bit more Accurate ECN (AcE) field

  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |               |           |           | U | A | P | R | S | F |  
  | Header Length | Reserved  |   AcE    | R | C | S | S | Y | I |  
  |               |           |           | G | K | H | T | N | N |  
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

→ 8 possible codepoints: 5 for "congestion indication" (CI) counter; 3 undefined

  +---------------------------------------------+  
  | AcE | NS | CWR | ECE | CI (base5) |  
  +---------------------------------------------+  
  |  0  |  0 |  0  |  0  |      0     |  
  |  1  |  0 |  0  |  1  |      1     |  
  |  2  |  0 |  1  |  0  |      2     |  
  |  3  |  0 |  1  |  1  |      3     |  
  |  4  |  1 |  0  |  0  |      4     |  
  |  5  |  1 |  0  |  1  |      -     |  
  |  6  |  1 |  1  |  0  |      -     |  
  |  7  |  1 |  1  |  1  |      -     |  
  +---------------------------------------------+
More Accurate ECN Feedback in TCP

More Accurate ECN TCP Receiver

- congestion indication (CI) counter: # of CE markings during a connection
  → For each incoming segment with a CE marking increase CI by 1

- With each ACK calculate CI modulo 5 and set the respective codepoint in the AcE field
  - SHOULD switch from a delayed ACK behavior to send ACKs immediately in a high congestion situation (to avoid counter wrap-arounds)
  - MUST send value of CI counter by default; send other (undefined) codepoints if needed
  - MUST echo the CI in the next ACK whenever a CE is received
  - MUST repeat each CI codepoint directly on the subsequent ACK
  → Cope with a certain amount of ACK loss and avoid counter wrap-arounds

More Accurate ECN TCP Sender

- Congestion indication received (CI.r) counter: # of CE markings as signaled by the receiver, and reconstructed by the sender
  → For each ACK increase CI.r by D (= difference between local CI.r value modulo 5 and the signaled CI value of the codepoint in the ACK)
Use with ECN Nonce (Appendix)

Codepoints with Dual Counter Feedback

One field in TCP ACK but encoding 2 counters in 8 codepoints

1. Congestion Indication (CI) counter: number of CE marks
2. ECT(1) (E1) counter: number of ECT(1) signals (ECN Nonce Sum)

+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+
| ECI | NS | CWR | ECE | CI (base5) | E1 (base3) | ... | 1 | ... | 1 | ... | 1 | ... | 1 | ... |
+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+
| 0 | 0 | 0 | 0 | 0 | 0 | - | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | - | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 0 | 2 | - | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 1 | 0 | 1 | 3 | - | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4 | 1 | 0 | 4 | - | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 1 | 1 | 0 | 5 | - | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 1 | - | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 2 | - | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

- By default an accurate ECN receiver MUST echo the CI counter (modulo 5)
- The receiver MUST repeat the codepoint directly on the subsequent ACK
- Whenever ECT(1) occurs, E1 will be echoed; expect CI has increase before next ACK
Review Comments

1. Needs to be clearer what an implementer must do
   → We will address this

2. Relation to RFC3168?
   – Does not obsolete RFC3168 but does it update RFC3168?

3. Single repeat of each codepoint unnecessary?
   – Proposal to calculate whether wrap could have happened,
     and to calculate the worst-case count of CE marks
   → We will add this heuristic
   → A repeat can still improve robustness (e.g. if less than every second segment get ack’ed)

4. ECT in the IP header of the SYN and SYN-ACK
   – In SYN optional! We will address this
   – In SYN-ACK optional or mandatory? I’d say optional

5. ECN Nonce support needs to be mandatory (only for receiver feedback)
   – Working group feedback?

6. Advanced Compatibility Mode
   – Should we remove this?

7. Alternative proposal
Adoption as working group document?

1. Accurate ECN Feedback Option in TCP
   (draft-kuehlewind-tcpm-accurate-ecn-option-01)
   → Is the working group interested in adopting this document to specify a new TCP Option for accurate ECN feedback (AccECN)?
      Fields and size of the option are open to changes....

2. More Accurate ECN Feedback in TCP
   (draft-kuehlewind-tcpm-accurate-ecn-01)
   → Is the working group interested in adopting this document to specify for a more accurate ECN feedback mechanism which is reusing bits in the TCP header and as such can (only) be used as an alternative to the classic ECN?
      → Does the working group see the codepoint coding scheme as a good starting point for a more accurate ECN feedback mechanism?
Question?
Backup
Negotiation in the TCP Handshake

1. Host A indicates a request to get more accurate ECN feedback by setting \( \text{NS}=1, \text{CWR}=1 \) and \( \text{ECE}=1 \) in the initial SYN

   Classic ECN will still be negotiated (with CWR=1 and ECE=1)

2. Host B returns a SYN ACK with flags \( \text{CWR}=1 \) and \( \text{ECE}=0 \)

   Broken receiver that just reflect SYN bits get detected

---

Ac: *A*ccurate ECN Feedback, N: ECN-*N*once (RFC3540), E: *E*CN (RFC3168),
I: Not-ECN (*I*mplicit congestion notification)