60th IETF, TCPM WG:

TCP Corruption Notification Options
draft-welzl-tcp-corruption-00.txt

Michael Welzl
http://www.welzl.at , michael.welzl@uibk.ac.at

Distributed and Parallel Systems Group
Institute of Computer Science
University of Innsbruck, Austria
Motivation

- TCP problems with noisy links are well known
  - lots of research efforts
  - lots of possibilities if a sender knows about noise
  - but: no “noise notification”

- Explicit (ICMP style) signaling was unsuccessful up to now
  - Scaling issues, layering problems, etc.

- Partial and separate checksums have gained some IETF acceptance
  - Partial checksums (give erroneous data to apps):
    - UDP Lite - RFC 3828 (Proposed Standard)
    - DCCP
  - Separate checksums (distinguish congestion / corruption):
    - DCCP ("Data Checksum Option")
Proposal

• Introduce separate checksums to TCP
  - Note: partial checksums useless here (reliably transmit erroneous data?)

• How it works: additional (CRC-32c) checksum in TCP option
  - covers relevant header parts (pseudo header)
    • only look at it if TCP checksum fails
    • if it succeeds, we know that packet #XY experienced corruption!
      (otherwise, we don’t even know if the port numbers are correct)

• Notify sender
  - now, it’s an option in the ACK that says “the ACKed packet is corrupt”
  - alternatives:
    • flag in the reserved field of the TCP header
    • SACK-like option format (is this overkill?)

Not a new idea!
e.g., TCP HACK by Balan et al, Infocom 2001
Link layer considerations

- Currently, only few link layers hand over corrupt data
  - and typically must be configured to do so; why do it?
  - similarly, why support corrupt data in transport endpoints?
  - chicken-egg type of problem … someone has to make a start!
  - with UDP Lite (and DCCP), the IETF did (will do)

- Clearly, link layers should not always hand over corrupt data
  - how to signal from transport to link?
  - current strategy (UDP Lite and DCCP): link layer notices transport feature
  - IMO problematic, but lack of better alternative

- Link layers MAY notice this feature and hand over corrupt data
- Transport endpoints MUST stay robust no matter what link layers do
- Note: various reasons for corruption (e.g. broken equipment)
Using this feedback (potential benefits)

<table>
<thead>
<tr>
<th>Feedback usage</th>
<th>Status</th>
<th>Discussions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less severe congestion control reaction</td>
<td>will not be specified in this draft</td>
<td>AWJT</td>
</tr>
<tr>
<td>Earlier retransmission</td>
<td>will be removed</td>
<td>AWJT</td>
</tr>
<tr>
<td>Don’t retransmit if ACK is erroneous</td>
<td>specified</td>
<td>AWJT</td>
</tr>
<tr>
<td>Updating RTT estimate</td>
<td>will not be specified in this draft</td>
<td>DWJT</td>
</tr>
<tr>
<td>Earlier congestion control reaction to ECE=1</td>
<td>specified</td>
<td>ND</td>
</tr>
<tr>
<td>Earlier reaction to control flags (SYN, FIN..)</td>
<td>specified</td>
<td>ND</td>
</tr>
</tbody>
</table>

**Discussion code:**

- **ND** ... Not Discussed
- **AWJT** ... Agreed With Joe Touch :)
- **DWJT** ... Discussed With Joe Touch
Summary

- Research demand + attractive (potential) benefits + some agreement on partial / separate checksums = about time to add this to TCP

- Still lots of open issues
  - How to use this feedback: more discussions required
  - When to include these options?
  - Feedback format
  - Really no better inter-layer communication possible?

- Request
  - Volunteers for implementations and simulations