Router participation in Congestion Control

Motivation

Techniques
- Partial Packet Discard
- Random Early Detection
- Explicit Congestion Notification

Router support for congestion control

Recall that TCP's congestion control was implemented in end-systems for convenience: it would have been difficult to update routers.

If you could update routers, what would you change?
Constraint: Per-flow isolation may be too complex (amount of state and determining when flows begin/end) to be feasible.
We'll consider per-flow scheduling in a later lecture.

Possibilities:
- Discard long packets (allow pure acks to pass)
- Partial Packet Discard
- Random Early Discard
- Explicit Congestion Notifications
  All create dependencies between network layer and transport layer.

References

Partial Packet Discard: Keshav § 9.7.2
Random Early Detection:
  - Keshav § 9.7.3
  - RFC: 2309
Explicit Congestion Notification†:
  - Keshav § 13.4.9
  - RFC 3168

Overview:

† Known as Explicit Forward Congestion Indication (EFCI) in ATM, Forward ECN (FECN) in Frame Relay.

Outline
Review: Segmentation and Reassembly

Many layers segment the Service Data Unit (SDU) that they receive from the higher layer into multiple smaller Protocol Data Units (PDUs) e.g. 1: Network layer may fragment a large amount of data (e.g. 9KB) supplied by the transport layer so that fragments can be transmitted over a link with a smaller Maximum Transmission Unit (e.g. Ethernet with 1500B maximum frame length needs 9KB to be fragmented into 6 parts).

2: Asynchronous Transfer Mode: Uses short (48B of data) cells in order to limit voice packetisation delays. If Ethernet frame is to be sent over an ATM link connecting routers then it needs to be segmented (e.g. 1500B Ethernet frame → 32 cells)

We’ll refer to the large SDU as the “whole” and the smaller PDUs as “parts”.

For generality, without confusion about segments (=parts in ATM, = whole for TCP/IP), fragments, packets, frames, cells, ...

† TCP may apply a Path MTU discovery process, which may render network layer fragmentation unnecessary.

Partial Packet Discard

The smaller the data unit, the higher the percentage overhead for encapsulation fields such as sequence numbers and checksums

⇒ “parts” often just include addressing, and omit reliable transfer fields.

⇒ Reliable transfer only operates on the “whole”. If any “part” is lost, must retransmit the “whole” e.g. TCP retransmits “whole” (segment) when one IP “part” (fragment) is lost

⇒ If router discards one part of the whole, then it might as well discard all ensuing parts: they will be retransmitted anyhow.

- Good policy for applications requiring complete transfer: Releases space in router’s buffer
- Bad policy for applications that tolerate small loss – magnifies the loss

Outline

1. Early congestion notifications

“Early” notifications inform end-systems that the network is “congested” before buffers overflow in network elements (routers).

- Excessive delay may be considered to reflect congestion, not just packet loss
- Sources will take some time to respond to notifications

⇒ early notifications may enable congestion (buffer overflow) “avoidance”

Two forms of early notifications:

- Discard: “Active Queue Management (AQM)”
- Explicit Congestion Notifications
Active Queue Management (AQM)

- Notification takes same form as inevitable buffer overflow. ⇒ can exploit existing end-system responses.
- The hope is that the early discard will send a signal to sources to slow down, and they will respond in time to avoid buffer overflowing.
- The assumption is that sources slow down in response to packet loss. i.e. layer interdependence: the network layer depends on certain transport protocols being used.

Details in RFC 2309

Random Early Detection† (RED)

Discard arriving packets with a probability that increases linearly with the average queue length. Discarding packets before queue overflows provides an early warning to sources.

1. Increasing discard rate with queue length indicates severity of congestion and penalises sources that don’t slow down.

2. Averaging the queue length rather than using instantaneous measurement allows limited bursts to pass, accommodating TCP’s Slow Start

† Sometimes called “Random Early Discard”

Outline
Explicit Congestion Notification

Motivation:
- Early discard is crude: wastes bandwidth; packets propagate from source to congested router, only to be discarded.
- Increases packet loss
- May be better for routers to explicitly inform sources (through an ECN signal rather than packet loss) that they are getting congested.

Limitation of ECN:
- Presence of ECN indicates congestion
- Absence of ECN doesn’t prove non-congestion (congestion might have resulted in ECN signal being lost)
  ⇒ sources still have to respond to loss as a congestion indicator
  ⇒ still have ambiguity between congestion/transmission error

ECN mechanisms

Router could directly tell the source:
- Send a new packet to the source
- But that would create extra network traffic during times of congestion (Early Internet had “Source quench”, but use has been deprecated.)
- Set a field in a packet already going to the source (“Backward ECN”)
  - Router has to wait for the packet
  - Delay in signalling a source response
  - Router is complicated by having to tell particular sources.
  - Source transmitting to multiple destinations won’t know which need to slow down
  ⇒ Used by Frame Relay etc, but not Internet
  - Router could tell destination, and destination (if it receives the packet) tells source.
    “Forward ECN”, used by Frame Relay, and now the Internet.

ECN in the Internet

IP header was: (CU=Currently Unused)

Becomes:

<table>
<thead>
<tr>
<th>PRECEDENCE</th>
<th>TOS</th>
<th>DECF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ECT  CE</th>
<th>Not-ECT</th>
<th>ECT(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1</td>
<td>ECT(1)</td>
<td></td>
</tr>
<tr>
<td>1 0</td>
<td>ECT(0)</td>
<td></td>
</tr>
<tr>
<td>1 1</td>
<td>CE</td>
<td></td>
</tr>
</tbody>
</table>

ECN in the Internet Protocol

2 ECN bits (bits 6&7 of the TOS/DS field of the IP header indicate:

- Traditional Not-ECT
  - Routers only changes ECT field if set to ECT(1) or ECT(0)
  - Traditional receivers check that reserved bits are set to 0
- Congestion Experienced (CE)
- ECN-Capable Transport (ECT)
  - If ECT is set and router is experiencing congestion, router may change ECT field to CE, rather than discarding the packet.
  - Why 2 codepoints? See slide “Detect lying with 2 ECT values”

RFC 3168 (obsoletes RFC 2481)
Details: Transport reaction to ECN

"an ECT codepoint MUST NOT be set in a packet unless the loss of that packet in the network would be detected by the end nodes and interpreted as an indication of congestion." [3168] e.g. don't set it in pure (not piggybacked) ACKs, since ACK rate isn't reduced in response to ACK loss.

Rate of reaction: "end-systems should react to congestion at most once per window of data (i.e., at most once per round-trip time), to avoid reacting multiple times to multiple indications of congestion within a round-trip time." [3168]

Form of reaction to CE packet: "the congestion control algorithms followed at the end-systems MUST be essentially the same as the congestion control response to a *single* dropped packet." [3168]

Traditional sources could receive unfair treatment if routers perform early discard to signal congestion rather than marking CE:

End systems need to negotiate whether to use ECN. Source that mistakenly thinks that destination is ECN-capable may penalise dest'n for making false reports about no CE issues arising from multiple versions during incremental deployment.

ECN and TCP

Reserved field shrinks by 2 bits to indicate:

- **ECN-Echo (ECE):** Echoes to source receipt of CE.
  - Destination continues setting ECE in ACKs until it receives CWR from source (in case ACKs are lost).
- **Congestion Window Reduced (CWR):** Source informs destination that window has been reduced. Destination need no longer echo previous CE.

In SYN segments: ECE&CWR ⇒ ECN capable

Typical series of events

1. Source establishes connection with destination, and negotiates to use ECN.
2. Source sends segments with IP=ECT, CWR=0
3. Router is "congested". Changes IP=ECT to CE. Router:
   - Doesn’t process TCP fields
   - Likely marks all segments received while congested with IP=CE. Source will only reduce rate for one segment, but marking all segments simplifies router and expedites signal in case only segment with IP=CE is lost.
4. TCP destination receives segment, sends ACK, ECE=1
   - Keeps setting ECE=1 for all acks until step 6.
5. Source receives ACK with ECE=1, reduces congestion window. Sends segment with CWR=1 (IP=ECT).
6. Destination receives segment with CWR=1, and sends ACK with ECE depending on IP field of segment:
   1. IP=CE ⇒ router still congested ⇒ ECE=1
   2. IP=ECT ⇒ router no longer congested ⇒ ECE=0

Detect lying with 2 ECT values

- One ECT value is enough s.t. a router can determine whether end-systems support CE markings.
- But uncooperative routers/destinations might suppress CE markings (e.g. greedy destination that wants to maximise its rate without regard to network congestion)
- Two ECT values allow a source to check the destination’s reporting of CE markings:
  - Source varies ECT 0/1 value according to a pattern that it alone knows (e.g. pseudorandom) – “nonce”
  - Destination must return the pattern back to the source, e.g. using a new “Nonce Sum” TCP header field (sum of ECT(x) fields received).

For details, see RFC 3540
ECN summary

- Reduces waste from loss
- Can’t entirely replace loss as a congestion indicator, since during extreme congestion loss may prevent ECN carriage.
- Router marks packets as having experienced congestion; destination signals to source to reduce rate.
- IP’s coding (using 2b)
  - provides backwards compatibility with non-ECN capable transport.
  - enables a source to detect false reporting.