Explicit Congestion Notification - ECN
Birds of a Feather Session

Sally J. Floyd & K. K. Ramakrishnan
Lawrence Berkeley National Lab AT&T Labs. Research
41st IETF
Los Angeles, CA.
Agenda

- Introductions and Agenda planning
- Current plan:
  - Introductory material - K. K. Ramakrishnan
  - Important details and issues related to cooperation - Sally Floyd
  - Security and issues related to IPSEC - Steve Bellovin
  - Experimental Results - report from UCLA
  - Other items?
  - Planning for the future
Proposal to add Explicit Congestion Notification

- TCP - the dominant transport protocol in the Internet
  - embodies a set of mechanisms for congestion control and avoidance

- Congestion is detected by packet loss
  - mechanisms exist to adapt to congestion while minimizing throughput degradation due to packet loss
    » fast retransmit and fast recovery

- Depending on packet loss by queue overflow as congestion indication
  - builds up queueing delay; variability in delay
  - undesirable unfairness due to “tail drops”
  - synchronization of loss across multiple flows
Random Early Detection

◆ Active queue management methods (RED)
  – can be used to maintain a smaller average queue
    » reduces delay and its variability
◆ RED intended also to improve fairness and avoid synchronized losses due to “tail drops”
◆ RED mechanisms being actively considered for implementation
◆ But, *loss* is still the primary indication of congestion
◆ Philosophically:
  – desirable to separate policies of queueing (or dropping) packets at a router from policies for indicating congestion
  – some applications are sensitive to loss: audio/video; interactive traffic (telnet, web-browsing). Sensitivity may also be to latency for retransmission.
Explicit Congestion Notification

- Routers provide an indication of incipient congestion through marking of packets

![Diagram showing IP Packet with ECT bit 1, CE bit 1, ECN Echo, and TCP Acknowledgment]

- Avoids packet losses
- When average queue size, using RED algorithms, exceeds a threshold, router *probabilistically* sets the Congestion Experienced (CE) bit

  » expectation: packet losses would be infrequent
Router Actions

◆ Router computes exponential weighted average for queue length as specified for RED

◆ Following rules for RED marking, if average queue length exceeds threshold, router determines it is congested, packet is selected:
  ◆ if ECT (ECN Capable Transport) bit is set for packet, packet’s CE bit is set probabilistically
    – if packet’s ECT bit is not set, packet is dropped

◆ Under more severe congestion, router drops incoming packet
  – when queue is full, or average queue size exceeds an upper threshold

◆ A packet with its CE bit set: leave it unchanged
Transport Protocol Actions

- Source and destination TCPs exchange desire to use and/or capability to use ECN at connection setup
- If the entities agree
  - ECT bit is set on transmitted packets
  - Receiver would look at CE bit
- Originating packets have the CE bit clear
- When ECN-Echo bit in Ack. is received set
  - treat just as a congestion loss was detected
    - halve congestion window & reduce “ssthresh”
    - do not increase “cwnd” in response to Ack. if ECN-Echo bit set
Transport Protocol Actions (contd…)

◆ Do not react to Explicit Congestion indication more than once every window of data

◆ At “t”, source TCP reacts to ECN-Echo
  – note packets outstanding (sent, acks. awaited)
  – react to ECN-Echo only after those packets are acked at time “u”

◆ Also, during interval “t” to “u”, do not reduce congestion window in response to duplicate acks.
TCP Receiver

◆ When TCP receives a packet with CE-bit set, set the ECN-Echo in returning Ack packet
◆ If Ack. Withholding implemented:
  – “OR” all CE bits of packets being acknowledged
◆ Generate duplicate Acks. as usual

◆ Congestion in the Ack. path is subject of further work
Applications that might benefit from ECN

- Very short web transfers: prefer to not have to wait for TCP’s retransmit timer to expire
  - to detect loss of single packet in very short transfer

- Realtime flows with (fixed or adaptive) playback times
  - user would rather receive the packet

- Reliable multicast - where overhead costs of retransmitting dropped packets can be expensive

- Low bandwidth telnet connections

- Hopefully, future applications where congestion notification is useful - and prefer to not have packet dropped.