DiffServ QoS and Browsers

David L. Black, EMC

Browsers and Web Applications

- Browser = Web Application Framework
  - Web Applications run in Browser
    - Example: JavaScript
  - Multiple web applications at same time
    - Separated (or should be) by web origin
    - Web origin example: http://www.example.com:8080
- Web Application networking
  - Want DiffServ QoS [DSCP/PHB] per application flow
  - It's not that simple: Web RTC is a good example

Interruption - Brought to you by the Acronym Police (DiffServ Branch)

- DSCP = Differentiated Services Code Point
  - Set in IP header to request network QoS
- PHB = Per Hop Behavior
  - What the network does when it sees a DSCP
  - DSCP:PHB mapping is network-specific
- CS = Class Selector (specific DSCP)
  - CS0-CS7 align with former IP Precedence concept
  - CS1 intended for Lower (than Best) Effort service
    - More on this in a few slides

DiffServ and Web RTC (I)

- Web RTC = Web Real Time Conferencing
  - – Audio, video, data between browsers
- NAT Traversal: Has to work
  - – Every home “router” contains a NAT.
  - – Goal: Minimize pinhole punching and maintenance
- Pinhole needed for each local port used
  - – So, run different types of traffic on same port
  - – UDP encapsulation preferred
- But what about QoS per traffic type?

DiffServ and Web RTC (II)

- Q: When is it ok to vary QoS within a 5-tuple?
  - 5-tuple = 2 IP addresses, protocol (e.g., UDP), 2 ports
  - For Web RTC and other real-time applications
- A: Only when transport protocol is UDP, but ...
  - ... even then, only with care (easy to get wrong)
  - Network may ignore or remove QoS differentiation
- Don’t vary QoS (DSCP/PHB) for TCP, SCTP, DCCP
  - Packet Reordering: Misinterpreted as drops
  - Multiple Drop Precedence (PHBs from an AF class)
    - TCP: No useful effects
    - SCTP & DCCP: Interaction not well-understood

Varying QoS within a 5-tuple: What can go wrong?

- Network may manage QoS by 5-tuple, not DSCP
  - Entire Web RTC session lands in one router queue
- Core networks may limit QoS differentiation
  - 3-4 classes are common, e.g., when MPLS is used
  - OTOH, most QoS problems are in edge/access
- Network may remove QoS differentiation
  - E.g., remark (bleach) to best effort at boundary
- TURN may have chosen TCP when you didn’t ask for it
  - That’s it, no Differentiated QoS for you today!
- Lower (than Best) Effort service is a crap-shoot
  - CS1 DSCP may result in Best Effort, or even better than BE

QoS and Web Applications: What to do?

- Insulate web app from network via browser API.
  - Web RTC does this, but ...
  - No interface to ask what services (PHBs) are available
  - Result: Browser use of DSCPs = Set & Pray
- How many QoS service levels? (Good Question)
  - Varies, e.g., between core and enterprise networks
  - Current Web RTC QoS draft uses a lot of levels
    - Goal: Obtain useful differentiation at least at sending edge
- Less (than Best) Effort via DiffServ
  - Only reasonable when best effort is acceptable

To Explore Further

- DART draft (done)
  - draft-ietf-dart-dscp-rtp (at RFC Editor)
  - Source of transport guidance in this talk
- Web RTC QoS draft (work in progress)
  - draft-ietf-tsvwg-rtcweb-qos
  - Browser DSCP/PHB support and web application recommendations
- Diffserv Intercon draft (work in progress)
  - draft-ietf-tsvwg-diffserv-intercon
  - Doing better than “Bleach to Best Effort” at network interconnection points