Motivation and Requirements for Datagram Control Protocol

Mark Handley mjh@aciri.org http://www.aciri.org/kohler/dcp/

Question

- How best to deploy congestion-controlled applications that require flow-like unicast semantics, but don't want absolute reliability?
 - Streaming Media
 - Internet Telephony
 - Unicast Multimedia Conferencing
 - Games (eg position updates)
 - probably many others we haven't thought of yet

Unicast Congestion Control

- End-to-end congestion control requires a flow of packets across which to perform congestion control.
 - Implies some sort of flow setup
 - Requires feedback messages.
- For an unreliable data flow, to do good congestion control you need feedback messages that reliably convey the feedback.

Unicast Congestion Control

- Different applications want different congestion control dynamics (eg AIMD, TFRC).
- Different congestion control algorithms need different feedback messages and feedback reliability.
 - AIMD needs to convey precisely which packets arrived, were ECN marked, etc.
 - TFRC only needs to convey congestion information based on averages.
 - Also need simple congestion control for the feedback channel.

Security Issues

- Flow setup involves getting someone to hold state.
 - We need to pay great attention to avoiding DoS attacks on the passive end of a connection ("servers").
- Unreliable flows may be harder to defend against packet spoofing.
 - It's harder to say what a valid sequence number is than with TCP.
 - Bursts of packet loss mean its not trivial to say whether a new packet is valid.

Ancilliary Issues

- Concern about poor application-level congestion control implementations.
- Firewalls and NATs don't co-exist very well with UDP.
 - Lack of explicit flow setup and teardown.
 - Lack of well-known ports because UDP flow setup is usually server-to-client.
- Concerns about deploying ECN with UDP applications.

Required Features of DCP

- Flow setup/teardown.
- Negotiation of congestion control algorithm and feedback.
- ECN capable.
- Sequence space and reliable acknowledgment mechanisms.
- Secure against spoofing and DoS.

Scope

• We'd like DCP to be extremely general purpose.

- Don't want to go through the pain of deployment more than once (especially in NATs, etc).
- A new transport protocol is not a short-term solution to anyone's problem.

• We'd like DCP to be very lightweight.

- Especially DCP Data packets.
- Otherwise people will use UDP where they care about efficiency.

Interaction with other layers

- Given that DCP supports unreliable data delivery, you can layer pretty much anything over it that:
 - ▶ is unicast,
 - ▶ is flow-based,
 - needs congestion control.
- Unlike TCP, there's no problem interleaving multiple data streams over one flow.

• What can't you layer over DCP?

- Basic Security (needs to be integrated).
- Mobility (?).

Mobility/Multihoming.

- You could layer DCP over Mobile IP.
 - Potential efficiency issues.
- You could add Mobility/Multihoming support to DCP.
- You could do Mobility at a higher layer.
 - As it's unreliable, you can just spawn a new connection.

Summary

- E2E Congestion Control needs to be performed on flows.
- If we want a congestion control layer for unreliable data, it needs to be closely coupled to flow setup and flow feedback.
- Flow setup has security implications.
- Nothing else is closely coupled.
 - Except possibly mobility/multihoming?