

On the Limited Usefulness of the Datagram Congestion Control Protocol (DCCP)

I like the protocol. What I have to say makes me sad.

DCCP design motivation

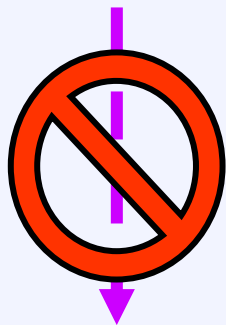
- Some apps want **unreliable, timely delivery**
 - e.g. VoIP: significant delay = ☹️ ... but some noise = 😊
- Unresponsive applications
 - endanger others (congestion collapse)
 - may hinder themselves (queuing delay, loss, ..)
- Implementing congestion control is difficult
 - illustrated by lots of faulty TCP implementations
 - should use precise timers ⇒ should be placed in kernel
- **DCCP** = e2e transport protocol for unreliable flows, well-defined framework for congestion control mechanisms
 - E.g. TCP-like congestion control or TFRC (smoother rate)

Classifying DCCP applications

- Congestion control trade-off (selfish single-flow view):

- + reduced loss
- necessary to adapt rate
 - Use sender buffer, drain it with varying rate
 - Change encoding

Trade-off: sender buffer size (=delay) vs. frequency of encoding changes



VoIP,
Games

videoconf.

Sweet spot?

Streaming Media



Delay sensitive

Delay insensitive

Is TCP the ideal protocol for one-way streaming media?

- Perhaps! Let's consider what happens...
- Remember: we're at the "buffering" side of the spectrum
 - Buffers (delay) don't matter
 - User perception studies of adaptive multimedia apps have shown that users dislike permanent encoding changes (big surprise :-)

⇒ no need for a smooth rate!
- **Little loss case**: TCP retransmissions won't hurt
- **Heavy loss case**:
- DCCP: 1, ~~2~~, ~~3~~, 4, ~~5~~, ~~6~~, 7, ~~8~~, ~~9~~, 10...
- TCP: (assume window = 3): 1, ~~2~~, ~~3~~, 2, ~~3~~, ~~4~~, 3, ~~4~~, ~~5~~, 4...
 - Application would detect: 4 out of 10 expected packets arrived
⇒ should reduce rate
 - Is receiving 1, 4, 7, 10 instead of 1, 2, 3, 4 really such a big benefit?
 - Or is it just a matter of properly reacting?
 - In RealPlayer and MediaPlayer, TCP can be used for streaming... seems to work well

DCCP usage: incentive considerations

- Benefits from DCCP (perspective of a single application) limited
- Compare them with reasons not to use DCCP
 - programming effort, especially if updating a working application
 - common deployment problems of new protocol with firewalls etc.
- What if dramatically better performance is required to convince app programmers to use it?
- Can be attained using “penalty boxes” - but:
 - requires such boxes to be widely used
 - will only happen if beneficial for ISP:
financial loss from unresponsive UDP traffic > financial loss from customers whose UDP application doesn't work anymore
 - requires many applications to use DCCP
 - chicken-egg problem!

Please tell me I'm wrong!

Thanks! :-)