Template for Alternative Transport Protocols

Title: Datagram Congestion Control Protocol (DCCP)

Contacts (Include institutions):
Eddie Kohler, Mark Handley, Sally Floyd, Jitendra Padhye
ICIR (The ICSI Center for Internet Research)

URLs / RFCs / Papers

DCCP maintainer web site: http://www.icir.org/kohler/dcp/
IETF DCCP working group: http://www.ietf.org/html.charters/dccp-charter.html
Several Internet-drafts; main spec: draft-ietf-dccp-spec-02.txt (TXT)

Principle / Description of Operation

The Datagram Congestion Control Protocol provides a well-defined protocol framework with several options and features (standard ACK format, some security using nonces, Path MTU discovery, identification, negotiation mechanisms, connection set-up and tear-down, ..) for congestion control mechanisms. It does not prescribe a certain congestion control behavior by itself – such specifications are written in separate documents. An additional user guide explains, among other things, how to use of DCCP for real-time media applications. An API to DCCP may, for example, allow an application to specify preferences for negotiable features prior to the initiation of the session – which allows DCCP to perform feature negotiation as per application references without explicit interaction with the application. The idea is to relieve application programmers from the burden of implementing a specially suitable congestion control mechanism and move this logic into the network stack, where it belongs.

Supported operation mode:

memory to memory (general transport)

Authentication:

DCCP has "Identification options", which provide a way for DCCP endpoints to confirm each others' identities, even after changes of address or long bursts of loss that get the endpoints out of sync. DCCP does not provide cryptographic security guarantees, and attackers that can see every packet are still capable of manipulating DCCP connections inappropriately, but the Identification options make it more difficult for some kinds of attacks to succeed.

Implementations / API:

Links to kernel- and user-level Linux implementations are maintained at
http://www.icir.org/kohler/dcp/

Congestion Control Algorithms:

The main DCCP specification does not prescribe any kind of congestion control; the functionality of such mechanisms is defined in special "profile" documents. Presently, there are two profile definitions, Internet-draft *draft-ietf-dccp-ccid2-02.txt* for TCP-like congestion control and Internet-draft *draft-ietf-dccp-ccid3-02.txt* for TFRC congestion control. TFRC is an equation based congestion control mechanism that is TCP-friendly and shows a smoother rate than TCP; more information can be found at http://www.icir.org/tfrc/

Fairness:

The two defined profiles, TCP-like congestion control and TFRC, are TCP-friendly. This defines their fairness. DCCP itself does not define the fairness of mechanisms.

TCP Friendly:

The two defined profiles, TCP-like congestion control and TFRC, are TCP-friendly. This defines their fairness. DCCP itself does not define the fairness of mechanisms.

Predictable Performance Model:

As DCCP itself merely encapsules congestion control mechanisms, it does not have a "performance model".

Results:

As DCCP itself merely encapsules congestion control mechanisms, it does not have performance results. Implementation experience reports are at http://www.cs.berkeley.edu/~laik/projects/dccp/index.html

Target Usage Scenario:

DCCP is a general-purpose protocol for any kind of unreliable data traffic, as encountered with streaming media applications or real-time multimedia Internet applications, for example. Its performance depends on the chosen congestion control mechanism.