UDP File Transporter

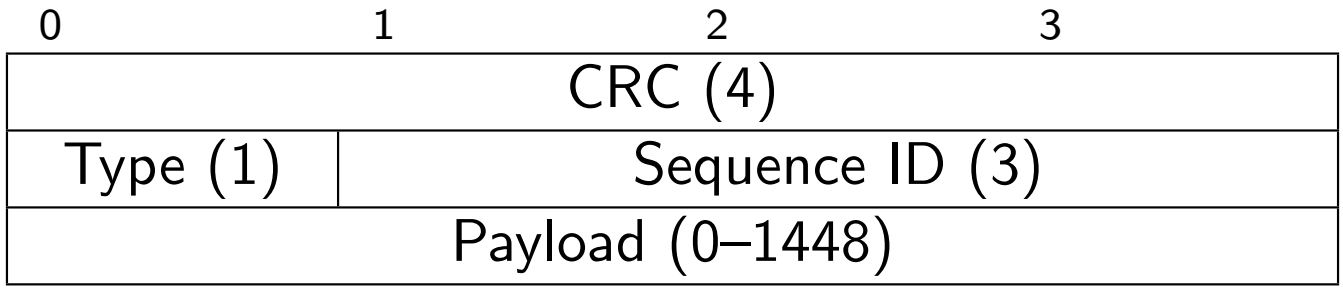
Trevor Spiteri
tspiteri@ieee.org

3 February, 2006

Outline

Outline

The Structure of the Datagram



CRC covers Type, Sequence ID, Payload

Type required operation and flags

Sequence ID identifies successive datagrams

Payload data required by operation

Mechanism for Downloading a File

- The client pulls data from the server.
- The client asks for each segment using one datagram.
- The server responds each datagram with required segment.

Mechanism for Downloading a File

- The client pulls data from the server.
- The client asks for each segment using one datagram.
- The server responds each datagram with required segment.

Mechanism for Downloading a File

- The client pulls data from the server.
- The client asks for each segment using one datagram.
- The server responds each datagram with required segment.

Datagrams Used for Downloading a File

Client request for downloading a file

	Type	Payload	Flags
(a)	4	offset, filename	none

Server reply for downloading a file

	Type	Payload	Flags
(a)	4	contents	EOF (optional)
(b)	4	empty	Error

Mechanism for Uploading a File

- The client pushes data onto the server.
- The client places a segment into each datagram.
- The server acknowledges each datagram.

Mechanism for Uploading a File

- The client pushes data onto the server.
- The client places a segment into each datagram.
- The server acknowledges each datagram.

Mechanism for Uploading a File

- The client pushes data onto the server.
- The client places a segment into each datagram.
- The server acknowledges each datagram.

Datagrams Used for Uploading a File

Client request for uploading a file

	Type	Payload	Flags
(a)	5	0 (32-bit), filename, contents	EOF (optional), Resume (optional)
(b)	5	upload id, offset, contents	EOF (optional)

Server reply for uploading a file

	Type	Payload	Flags
(a)	5	upload id, complete offset, capacity offset	none
(b)	5	empty	Error

Outline

Timeout Calculation Using Round Trip Time Estimate

RTT and RTT deviation estimate:

$$\begin{aligned} \textit{RTT difference} &= \textit{actual RTT} - \textit{RTT estimate} \\ \textit{new RTT estimate} &= \textit{RTT estimate} + \tfrac{1}{8}\textit{RTT difference} \\ \textit{new RTT deviation} &= \tfrac{3}{4}\textit{RTT deviation} + \tfrac{1}{4}\left|\textit{RTT difference}\right| \end{aligned}$$

The timeout is then calculated to be:

$$\textit{Timeout} = \textit{RTT estimate} + 3\textit{RTT deviation}$$

How the Client Avoids Network Congestion

- The client is solely responsible for congestion avoidance.
- The window size is initially set to one datagram.
- Window grows when no datagrams are lost.
- Window size halved for every lost datagram.

How the Client Avoids Network Congestion

- The client is solely responsible for congestion avoidance.
- The window size is initially set to one datagram.
- Window grows when no datagrams are lost.
- Window size halved for every lost datagram.

How the Client Avoids Network Congestion

- The client is solely responsible for congestion avoidance.
- The window size is initially set to one datagram.
- Window grows when no datagrams are lost.
- Window size halved for every lost datagram.

How the Client Avoids Network Congestion

- The client is solely responsible for congestion avoidance.
- The window size is initially set to one datagram.
- Window grows when no datagrams are lost.
- Window size halved for every lost datagram.

Summary

We have seen

- The datagram format.
- The protocol used for downloading and uploading files.
- The flow control mechanisms involved.