

Construction of a Formal Validation

- Build a high level prototype and verify that the design criteria are met
- Protocol design is an iterative process. Not likely to be correct first time around
- Each time a design phase is completed, be convinced it is error-free

George Blankenship

3

George Blankenship

Formal Validation

File Transfer Protocol

- A point-to-point protocol
- One sender and one receiver
- Provides *end-to-end* service between two users on two different machines

George Blankenship

4

5

6

4 Elements of the Protocol

- 1. Service Specification
 - a. connection establishment
 - b. termination

Formal Validation

Formal Validation

- c. recovery from transmission errors
- d. flow control strategy
- 2. Transfer ASCII text files
- 3. Low undetected bit error probability
- 4. User able to abort a file transfer in progress and protocol able to recover from message loss

George Blankenship

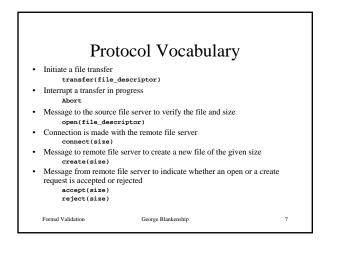
Channel Assumptions

- Full-duplex transfer
- voice grade telephone linesIgnore networking issues
 - i.e. routing

Formal Validation

• Minimal time for message to travel is approx 0.15 seconds

George Blankenship



Transfer Process Phases

- **1. Establishment of a connection** with the local file server
- **2. Establishment of a connection** with the remote file server
- 3. Transfer of data

Formal Validation

4. Orderly termination of the connection

George Blankenship

8

9

Connection Between File Servers

Two steps

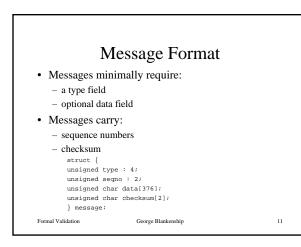
- 1. Initialization of the flow control protocol
- 2. A handshake with the remote system using the **connect** and **accept** or **reject** message
- The synchronization of local and remote flow control protocols is necessary to guarantee that they agree on the initial sequence numbers to be used
- A handshake using the message **Sync** and its acknowledgment **sync_ack** Formal Validation George Blankenship

ack Formal Validation

Data Transfer We need messages for retrieving the data from the file server and transmitting them to the remote system data(cnt, ptr) To signify the end of the transmission eof Completion of a file transfer close A simple flow control discipline that acknowledges correctly received data

George Blankenship

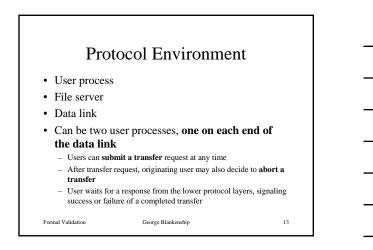
10



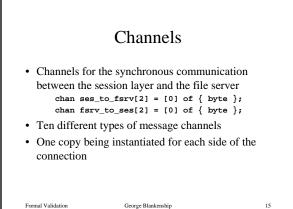
Procedure Rules

- Consider only the semantics of the protocol ignoring the syntax
- Layers
- Design divided into several layers:
 - Presentation layer: the user interacts with this layer
 - Session layer: controls the transfer itself
 - Data link layer: assumed that it can lose
- messages but not distort them Formal Validation George Blankenship

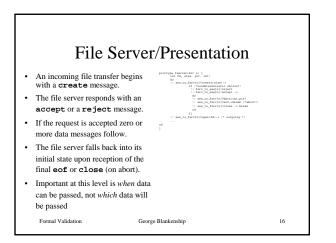
12

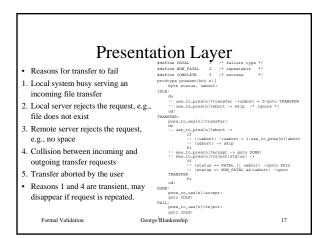


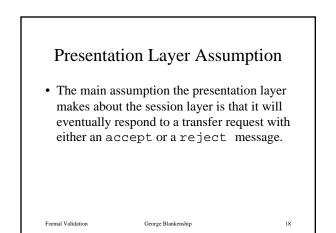
	User Layer	
<pre>P proctype user_proce user_to_pres[n]!t if</pre>		
:: pres_to_u :: pres_to_u	use[n]?accept -> goto Done use[n]?reject -> goto Done res[n]!abort->goto Aborted	
Aborted: if :: pres_to_u	use[n]?accept -> goto Done	
:: pres_to_u fi; Done: skip }	use[n]?reject -> goto Done	
n - identifies the user and the o rransfer - a message ordin tdefine OSZ N /* gu	narily carry a parameter that points to the file transf	erred
chan use_to_pres[2] chan pres_to_use[2] chan pres_to_ses[2]	= [QSZ] of { byte }; = [QSZ] of { byte };	
chan ses_to_pres[2] chan ses_to_flow[2]	<pre>= [QSZ] of {byte,byte}; = [QSZ] of {byte,byte};</pre>	
chan dll_to_flow[2]	<pre>= [QSZ] of {byte,byte}; = [QSZ] of {byte,byte}; = [QSZ] of {byte,byte};</pre>	
Formal Validation	George Blankenship	1

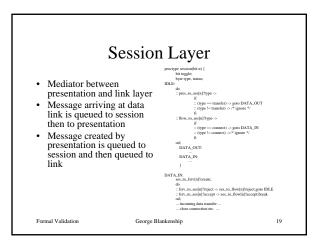


15

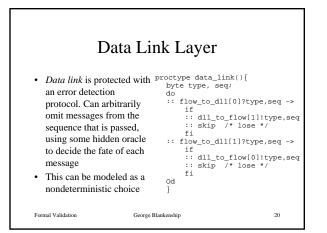


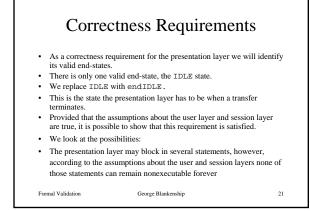












George Blankenship