



Some Findings from Assignment 1

Wide variety of specifications:

- ▶ Usually a couple of pages
- ▶ More protocol spec vs. more implementation spec
- ▶ More or less complete (at a first glance)



Encodings

- ▶ Binary + mixed text/binary
- ▶ CSV lists for header fields
 - Termination (implicit)?
 - One byte type code
- ▶ Box notation
 - 16 bit, 32 bit alignment
 - Partly reminds you of TCP
- ▶ Fixed frame size
 - 1024 byte frames vs. 1024 bytes data
 - 1024 kbits



Protocol Operation (1)

- ▶ Transport
 - Mostly UDP
 - Once TCP (oops?)
 - Separation into transport (binary) and application (text, HTTP-style)
- ▶ Connection setup
 - Explicit establishment via some handshake mechanism
 - Single handshake
 - Multi-stage process (negotiation, meta-data retrieval)
 - Implicit setup as part of a pull request
- ▶ File transmission
 - Sequence numbers
 - ACK-based, NACK-based, ACK/NACK combined
 - Timeouts + mismatches in sequence numbers
 - Various forms of checksums (e.g., CRC32, MD5, HMACs) but also none
- ▶ Flow control



Protocol Operation (2)

- ▶ Completion
 - Explicit end signaling
- ▶ Errors
 - Access denied, file not found, bad request
 - Negotiation failure (bit rate, etc.)
- ▶ Other
 - Negotiation support
 - Load balancing: option for a busy server to delay
 - Fairness: round robin as one example
 - Not mentioned very often



Some Observations on Possible Constraints

- ▶ Limited sizes (filename < 255 characters)
- ▶ Did you think about sequence number wrap around?
- ▶ Fixed timer values (are these the right ones?)



Protocol Design

Assignment 2: FRP Extensions



Part 2: FRP v2

- ▶ Extend your frp to support an adaptive congestion control scheme
 - Ack-clocking, TFRC-based, or something else deemed useful
 - Motivate and document your choice
- ▶ Your scheme should scale
 - From 10 kbit/s to 1 Mbit/s data rate
 - How about different delays?
- ▶ You obviously need to modify the semantics of “-b”
 - Suggests something useful for this parameter



Testing: udppipe

```
udppipe -l [lhost:]lport -c [chost:]cport -b <bitrate> [-d <delay>]
```

- l: transport address to receive UDP packets on from first uft peer; in the opposite direction, packets are sent to the address, they were received from
- c: transport address to send UDP packets to (the other uft peer which needs to transmit its responses must be taken from rcvfrom ())
- b: bit rate specified in kbit/s
- d: delay specific in milliseconds (default: none)

