



Assignment 1: fip

Design a protocol
Specify the protocol
Implement the protocol

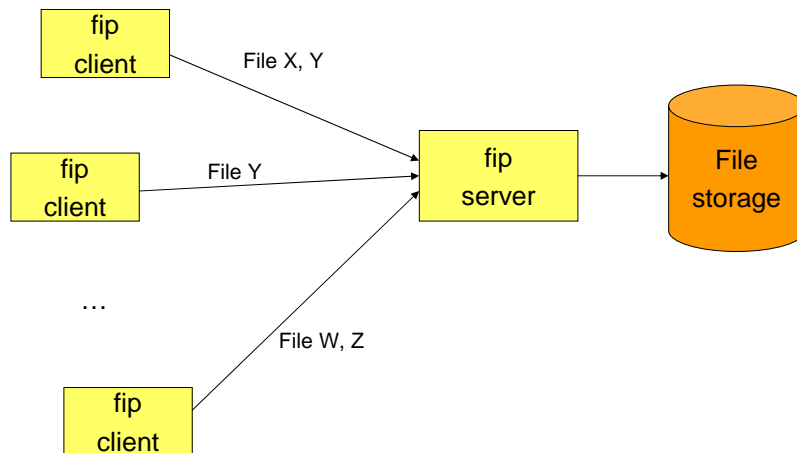


File Push: fip

- ▶ Scenario: news agency gathering information from large crowd of free contributors on different subjects
 - Examples: pictures of famous people, gathering pricing information, short stories about certain places
 - Contributors upload files per subject and “item”
 - Example: Subject: picture of person; item: Tarja Halonen
 - Represented by naming conventions for file names (for simplicity)
 - First come — win: only one article is allowed per subject
 - Many people may submit the files the same time
 - Several people may submit files with the same name at roughly the same time
 - Make sure that only one gets through and contents is not overwritten
 - Those who lose get error messages (“better luck next time”)
 - Let’s not worry about security for the time being...



File Push: fip



File Push: fip

- ▶ “Reliable” transfer of a file from many endpoint to one target
- ▶ Client mode operation
 - Initiate a transfer of one or more files to a server: send data
 - Sequentially or in parallel
- ▶ Server mode of operation
 - Wait for incoming files from a client
- ▶ File transmission shall take place in chunks of 1024 bytes
- ▶ File transmission should **not** be just lock-step
- ▶ File identification to be conveyed (i.e., the file name)
- ▶ File size to be included
- ▶ File checking information (e.g., a checksum)
- ▶ Other information conceivable (e.g., the sender identity for the award)
 - Choose where to put it depending on whether or not persistence is needed
 - E.g., protocol header or separate resource file
- ▶ Support “simulated” packet loss
 - Independently on both sender and receiver side



Some Issues to Consider

- ▶ How to do flow control?
 - There may be many senders at the same time.
- ▶ How to do error handling?
 - File exists already?
 - Receiver busy?
- ▶ How to deal with failed file transfers?
 - What is a failed file transfer?
 - How and when do you declare something failed?
- ▶ How do achieve fairness (and what is your definition of fairness)?
 - When is a file actually submitted?
 - Upon start or completion of the upload or anywhere in-between?
 - How to prevent Denial-of-File attacks?
 - E.g., somebody just starting to upload the most interesting 200 file names?



fip: Design and Specification

- ▶ Document (and motivate!) your design decisions
 - There are many possible approaches
- ▶ Write up a short specification for your protocol
 - Include sufficient detail so that one can understand and implement from it
 - Litmus test
 - Design together in your group
 - One or two of your group writes part of the spec
 - The other(s) try to understand it
 - Be critical: ask yourself what is really written there (as opposed to what might be meant)
 - No need to exaggerate on the spec though
- ▶ Hand in the spec by 2 April 2007
 - Complete your implementation by 5 April 2007
 - You will need to build on it in the second assignment



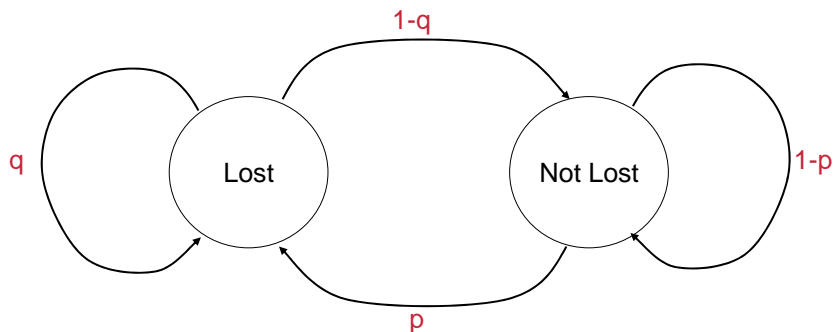
fip: Implementation

- ▶ Realize your protocol specification in some language
- ▶ Write a single program that can act as both sender and receiver
 - Distinguished by command line options
- ▶ Simulate your own packet losses
 - Trashing packets in your code before sending or after receiving
- ▶ Test it!
 - Does it “comply” with your spec
- ▶ Document what you did and what you learned
 - How is your program structured?
 - Which were the major implementation issues?
 - Did you have to adjust your spec during the implementation?
 - What would you do differently if you started all over again?



Packet loss simulation

- ▶ Choose a simple Markov chain
 - Then, we can play with dependent and independent losses





```
fip [-s] [-t <port>] [-p <p>] [-q <q>]  
fip <host> [-t <port>] [-p <p>] [-q <q>] -b <bitrate>  
    <file> [<file>*]
```

- s: server mode: accept incoming files from any host
Operate in client mode if “-s” is not specified
- <host> the host to send to or request from (hostname or IPv4 address)
- t: specify the port number to use (use a default if not given)
- p, -q: specify the loss probabilities for the Markov chain model
if only one is specified, assume p=q; if neither is specified assume no loss
- b: transmission bitrate for the file (gross transmission rate)
- <file> the name of the file(s) to send

Further options may be useful; up to you.

Remember to do report errors (locally and across the network) as needed.

You may want to do something useful if the user aborts either process (Ctrl-C).