# Assignment 1: frp

Design a protocol
Specify the protocol
Implement the protocol

© 2009 Jörg Ott & Carsten Bormann

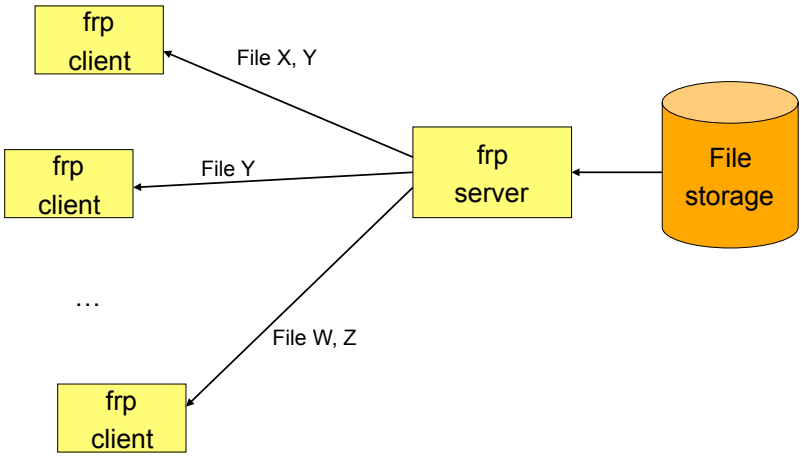



# File Pull: frp

- Scenario: news agency offers information to a large crowd of free writers on different subjects
  - Examples: pictures of famous people, gathering pricing information, short stories about certain places
  - Writers can requests files per subject and "item"
    - Example: Subject: picture of person; CV; birthday item: Tarja Halonen
    - Represented by naming conventions for file names (for simplicity)
  - Clients can requests multiple files
  - Let's not worry about security for the time being…
    - This is a public service
    - No username / password identification is needed

© 2009 Jörg Ott & Carsten Bormann

# File Pull: frp




© 2009 Jörg Ott & Carsten Bormann




# File Pull: frp

- "Reliable" transfer of a file from one source to many endpoints
  - Individual transfers
- Client mode operation
  - Initiate a transfer of one or more files from a server: receive data
    - Sequentially or in parallel
- Server mode of operation
  - Wait for requests 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 checking information (e.g., a checksum)
- Other information?
- Support "simulated" packet loss
  - Independently on both sender and receiver side

© 2009 Jörg Ott & Carsten Bormann

# Some Issues to Consider

- How to do flow control?
  - There may be many receivers at the same time. How to organize serving them?
- How to do error handling?
  - File does not exist?
  - Sender too 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)?
- How to prevent Denial-of-File attacks?

© 2009 Jörg Ott & Carsten Bormann




# frp: 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 5 April 2009
  - Ideally complete your implementation by 7 April 2009
  - You will need to build on it in the second assignment

© 2009 Jörg Ott & Carsten Bormann

# frp: Implementation

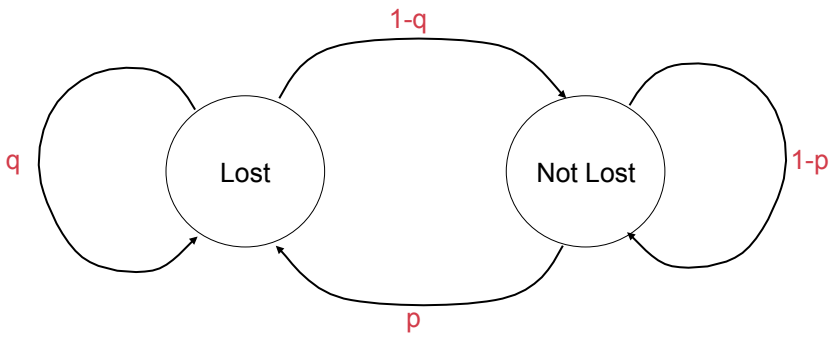
- Realize your protocol specification in some language
- Write a single program that can act as both sender and receiver
  - Distinguished by command line options
- Simulate your own packet losses
  - Trashing packets in your code before sending or after receiving
- Test it!
  - Does it “comply” with 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?

© 2009 Jörg Ott & Carsten Bormann




# Packet loss simulation

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




© 2009 Jörg Ott & Carsten Bormann

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

| | |
|---|---|
| -s: | server mode: accept incoming files from any host<br>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<br>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).

© 2009 Jörg Ott & Carsten Bormann