Protocol Design

Concluding remarks
Exam hints

Summary…

- There are many tools in the toolkit
- The wise craftsman knows which tool to choose

- Protocol design still is more an art than a science
- It does have sound engineering fundamentals
  - No matter how hard you push […] you can't increase the speed of light
- Forgetting the humans in the loop is a big mistake
  - Decision makers, bean counters
  - Implementers, operators
  - Users

Some of the things we did not talk about…

- Math & proofs
- Performance (some exceptions)
- Packet and XML design practices (syntax)
- Testing methods and tools, torture tests (too much of its own)
- Code generators and protocol compilers (deliberately left out)
- Techniques for efficient implementations (beyond scope)

- Could have added many more examples, further case studies, excruciating detail, …

Feedback

- Many ways
  - Official feedback website operational as of now
    http://palaute.ee.hut.fi/lomake.php?id=542
  - Talk to us: now, after the lecture, in the small group review
  - Send us email: cabo@tzi.org jo@netlab.tkk.fi
- We are interested in
  - What you liked and what did not like
  - What was unnecessarily included and (particularly) what was missing
  - What should be emphasized and what should be reduced
  - What type of assignments do come to mind
  - In general: what we could do differently (and why)

  Specific feedback is preferred by mail or personal discussions because this allows us to ask if something is unclear

Assignment finalization

- Short assignment review discussion with the groups
- 20min per group

- First opportunity: 12 May, 8 – 10, SE 324
  - Please send mail before
- Further opportunities to follow until the end of May and in June

Notes on Assignment 2

- Some textual descriptions were rather short
- At least one documentation file was not readable
- At least one textual description of the theory parts appears to be missing

- On stress testing
  - Did anyone try 64 KB packets?
  - Very nice and elaborate descriptions returned
  - Good analyses
- On uft2
  - Variety of different congestion mechanisms discussed
  - Partly even included evaluation based upon test runs

Questions on Assignment 3?

- Well, this is up to you…

Exam hints

Exam logistics

Thursday, 11 May 2006, 13 – 16, S3

Different types of questions (we will note which type which one is)

a) Short knowledge questions regarding the course contents
   - Probably 3 – 6 (or so)
   - May be free text, may be multiple choice
b) Mid-size "mini design" or more elaborate text questions
   - Probably 1 – 2 (or so)
c) One knowledge "transfer" question
   - May be an analysis, may be a small design

Basic idea: provide enough time to think and write, avoid rush

Entire course contents is relevant including assignments

Type A Questions

- Discuss two advantages and two disadvantages of FEC schemes. Give an example where using FEC makes sense and state why.
- Which problems for protocol design arise from NATs? Discuss two ways to address the problems. What are there pros & cons?
- Describe the soft state approach including its basic idea, advantages, and disadvantages.
- How do long fat pipes interact with (TCP) congestion control? Sketch a possible solution to address this problem.
- Which design considerations may affect the choice of a protocol syntax? Give two examples for such choice and motivate them.
- Which three aspects of robustness does a protocol design need to address?

Type B Questions

- Discuss the reasoning in favor of the end-to-end principle. Where do today's networks and applications deviate from this?
- Describe two protocol design aspects that are complicated by introducing (multicast-based) group communications. Sketch a possible approach to address each.
- Discuss advantages and disadvantages of link layer repair schemes, taking into account that the particular link(s) in question form only part(s) of a longer end-to-end path.
- Why are options often a bad idea?
- Assume a multicast-based web distribution scheme in which clients can ask a set of servers for contents—the delivery of which is then scheduled according to the relative demand. Sketch a simple DoS attack on this system and describe an approach how this could be fixed.

Type C Questions (1): Analysis

- Given the following fragment of a protocol specification that is supposed to update a database on a laptop of the sales force, which issues do you see (there are at least three)?
  - After initialization, the mobile node M compares the obtained address with its known home address space and, if they match (i.e., if in the office or after a VPN tunnel has been set up), it validates this by sending UDP multicast queries to its known servers in 1s intervals. The first server to respond is elected as "update server" for the subsequent data exchange. The response contains an address for sending updates to and for retrieving all changes to the database from. M uses TCP to connect to both these addresses and transmits and retrieves the latest data. M is notified asynchronously via the TCP connection about newly available data.

Type C Questions (2): Design

- Assume a simple (UDP-based) file transfer protocol between two nodes that allows to SEND file attributes, to send file DATA, REQUEST a file and/or its attributes, and allows to exchange CONTROL messages. Assume further that a daemon is running on each side that supervises a single directory (no subdirectories) and assume that the directory content is identical when they start.
- Describe a small application protocol on top of the file transfer protocol that keeps these two directories synchronized when a file is added, modified, or deleted on either side. Consider conflicting operations and propose some (possibly trivial but deterministic) resolution mechanism.
- How do you use which operations? What do you use control for? Which file attributes do you use?
- Would your protocol scale to more than two clients? Assess its complexity with respect to the number of peers (O (…)).