IK2213 Network Services and Internet-Based Applications

Peter Sjödin
KTH ICT/ECS/TSLab

Purpose

- The course provides practical and theoretical knowledge about
  - Design and implementation of protocols and services
  - Design and implementation of Internet-based applications
- A main goal is that you should apply your knowledge about networking and programming in practice, through programming assignments
- Is this a programming course?
  - No, not in the sense that there are teaching activities related to programming
  - But you will do quite a lot of programming, and learn a lot about network programming!
    - Should have previous experiences

Organization

- The course is organized as a number of lectures and assignments
  - Introduction lectures
    - UNIX networking programming
  - Guest lectures
    - Invited lectures
  - Four assignments
    - Introduction
    - Supervision occasions

Examination

- To successfully complete the course, you should
  1. Attend the compulsory lectures
  2. Complete the assignments within the given time limits

People

- Course responsible
  - Peter Sjödin
- Course assistants
  - Raul Jimenez
  - Dan Kopparhed
  - Erik Eliasson
- Lectures
  - Peter and invited guest lecturers
- Assignments
  - Raul, Erik, Peter, Dan
- You can contact us through course web (more later)

Lectures

1. Introduction: Protocol, services and implementations (Peter)
2. UNIX networking and protocols (Peter)
3. Using Subversion for efficient collaborative software development (Dan)
4. VoIP and SIP implementations (Erik)
5. Guest lectures (TBD)*

* Compulsory

Course Material

- No course book
  - Although you will have much use of a basic text book on Internetworking
    - Kurose & Ross, Forouzan, Comer, etc
- Web links and references on the assignments

Assignment Organization

- Basic principles:
  - Each assignment is presented at an "Assignment introduction" class
  - Around two weeks for each assignment
    - Deadline announced at assignment introduction
    - Submit your application through course web
  - About a week after assignment introduction, there is a supervision occasion
    - Instructors are there to discuss and answer questions
    - (See schedule for time and place)

Deadline and Make-up

- Deadline is strict!
  - No solution, no grade…
  - Course web submission closes on "Due date"
- You need to demonstrate that you have made a serious attempt at solving the assignment
- If you hand in a solution but it does not work as expected, you will get an opportunity to fix it
  - Typically two more weeks
    - You receive information about this when you get feedback from us
  - And that's it!
    - No further possibilities for make-ups

Communication

- Web pages
  - URL=http://www.ict.kth.se/courses/IK2213
    - Write it down! (Or bookmark)
  - Basic information about the course
  - Together with a link to the course web forum ("eLearning portal")

Course Web Forum

- Moodle course management system
- Linked from the main web page
  - "eLearning portal"
- Create your own account
  - Register as a user for the course "IK2213 Network Services and Internet-Based Applications"
  - Use your genuine name and identity
    - Phonies and fakes are expelled
  - Make sure to keep your profile up to date
- You can find all the instructors there, with email addresses

Assignments

- Network programming assignments
- Performed in groups of two students
  - If you don't have someone to work with, use the forum on the course web to advertise for a team mate
- Working code and short written report
  - 2-3 pages

Grading

- Each assignment has different levels of difficulties, for different grades
  - BASIC, MEDIUM, ADVANCED
- Your solution gets points
  - The number of points depends on
    - Level of difficulty
    - If you have to do a make-up
  - Your final grade is calculated from the total number of points
    - More information on course web
- You must pass every assignment

Hints and Tips

- Plan your assignment work!
- Start early!!!
  - We really, really mean that…
- Don't plan on using the make-up time for final fixes
  - You will need that time for the next assignment
  - Your grade won't be as good

Collaboration

- We encourage collaboration
  - So it is perfectly OK to collaborate and use information that you may find on the Internet
- BUT
  - Each group should produce its own solution
  - This course is about **learning by doing**
    - You can learn a lot
    - But if you don't do, you won't learn…
- If you get stuck or need help, use the discussion forum!
  - You will learn more this way

The Dull Part

- To plagiarize and duplicate is cheating
  - And will be treated as such
  - There are tools to detect this
- Both members of a group should share the work and be able to answer questions about the solution

Programming Environments

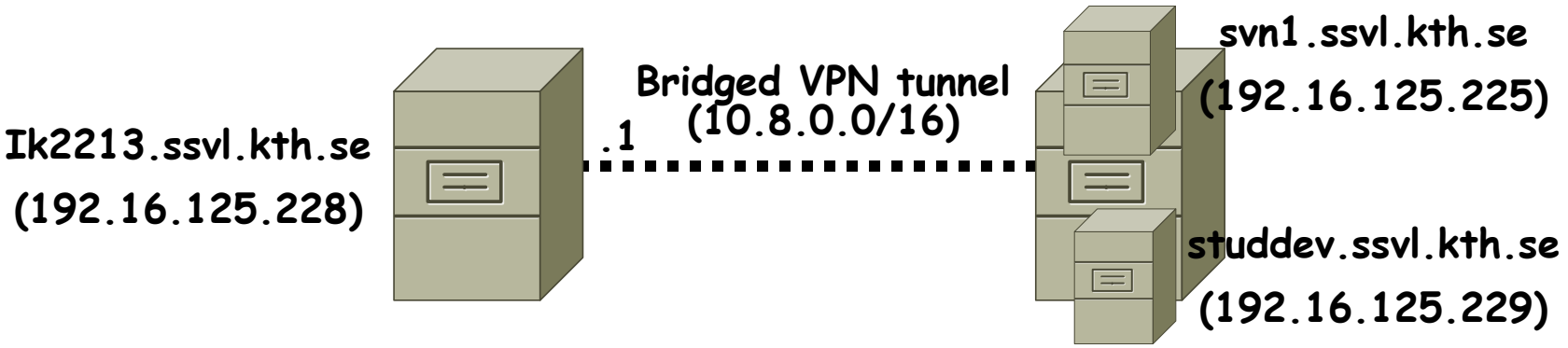
- Each assignment is designed with a specific **reference environment** in mind
  - Typically C or Java on Linux (OpenSUSE 10.3)
  - If we provide include files and library functions, it is for this environment (if any)
  - There is a **reference machine** where you can test your solutions
- You have some freedom to make your own choices
  - But you should check with us first!
  - Windows
  - Your solutions will be tested against the **reference machine**
- If you want to use external libraries and packages, it must be approved by the instructors!
  - Default is "No"!
    - You are here to learn network programming

Subversion

- We encourage you to use Subversion for version control and software development collaboration
  - subversion.tigris.org
- Subversion server
- More later…

Implementation Environment

- Server (Debian Etch) - ik2213.ssvl.kth.se
  - SMTP (Postfix) (Assignment 1)
  - VPN (openvpn) (Assignment 3)
  - WWW (apache2 – accessiable only through VPN) (Assignment 1, 2)
- **Reference machine** – openSUSE + Xen virtual machine
  - studdev.ssvl.kth.se (Xen guest OS - openSUSE)
    - Development utilities
    - Openvpn client (connected to ik2213 server)
  - You all get accounts on this machine

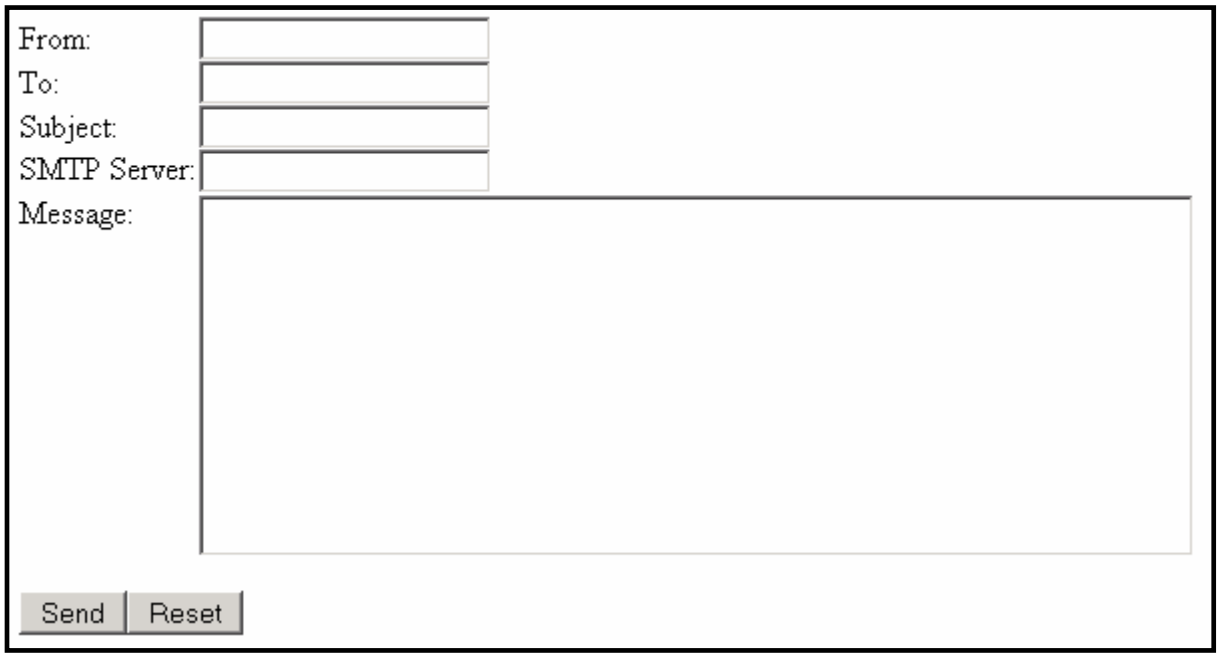


Assignments

1. WebMail
2. SIP Speaker
3. Reliable UDP
4. Packet bouncer

WebMail

- Web service for sending e-mails
- A web server, which
  - Opens up a (very simple) form, where a mail can be filled in
  - Contacts an SMTP server to deliver the mail



WebMail Implementation

- Small catch
  - This is intended for a small-scale system
    - Can't afford to use a large, existing web server, such as Apache
  - You will write the web server yourself
- You will learn about
  - Socket programming, HTTP, HTML, SMTP, SSL, and DNS

SIP Speaker

- SIP UA that waits for incoming calls and answers them when received
- VoIP answering service
- You will learn about
  - SIP, RTP and RTCP
  - Multi-user state management with UDP
  - Multi-protocol servers
  - TTS (Text to speech) systems

Reliable UDP

- UDP datagram service
  - Unreliable
  - Connectionless
  - The idea is that UDP is for applications that want to implement their own error and flow control
    - (Or that don't need it)
- But why must connectionless delivery be unreliable?
- Implement flow control and error control for UDP
- You will learn about
  - UDP socket programming
  - Event-driven programming
  - Window-based flow control and error control

Packet Bouncer

- Packet relaying
- Basic idea: modify IP packets "on the fly" so that they
  - are redirected to a different destination
  - appear to come from a different source
- Many application areas
  - NATs and NAT traversal, IP sprayers, proxies
- User space bouncer application
- Implementation issues
  - "raw" packets to/from the bouncer
  - Packet header modifications