



inf

INF3190 – Mandatory Assignment



UNIVERSITY
OF OSLO

INF3190 – Mandatory Assignment

- Must be approved prior to the submission of HE1
 - Individual work
 - The goal is to learn about *abstraction* and layering in networking stack
- ⇒ focus on the **link layer** and implementing the **flow control** feature

Learning:

1. UNIX socket programming in C
2. Sliding window (chapter 3, Tanenbaum's book)
3. Linux shell commands

Program's Structure (UI)

- A **client-server** program (tested on IFI machines) for “file transfer”
- **Usage:** `./main <port> <id>`
 - `<port>` is the UDP port used
 - `<id>` is the fake MAC address

SEND <filename>
QUIT
- **Step #1)** PHY is “UP”:
 1. Client **sends** a UDP packet containing “**CONNECT**” (a **unique number** for PHY link)
 2. Server **responds** with “**UP**” in a UDP packet (l1_linkup)
 3. Client assumes that PHY link is up now (l1_linkup)

... sending frames

Program's Structure

- **Step #2)** Flow control implementation
 - e.g. **Stop-and-Wait** (*acceptable* for this assignment)
 - Recommended to implement a **Go-Back-N** or **Selective Repeat** => 😊 with HEs
 - LL provides **full-duplex** communication for the higher layers

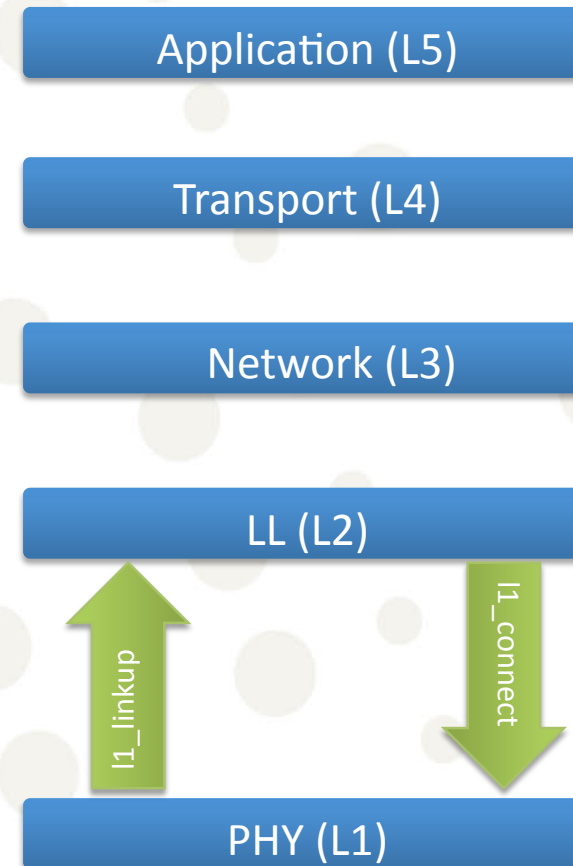
Pre-code

- You'll be given a pre-code with plenty of files (feel free to modify them):

delayed_dropping_sendto.c
delayed_dropping_sendto.h
delayed_sendto.c (designed to make the delay => to trigger the flow control mechanism)
delayed_sendto.h
irq.c
irq.h
l1_phys.c
l1_phys.h
l2_link.c
l2_link.h
l3_net.c
l3_net.h
l4_trans.c
l4_trans.h
l5_app.c
l5_app.h
main.c
slow_receiver.c (designed to make the delay => to trigger the flow control mechanism)
slow_receiver.h

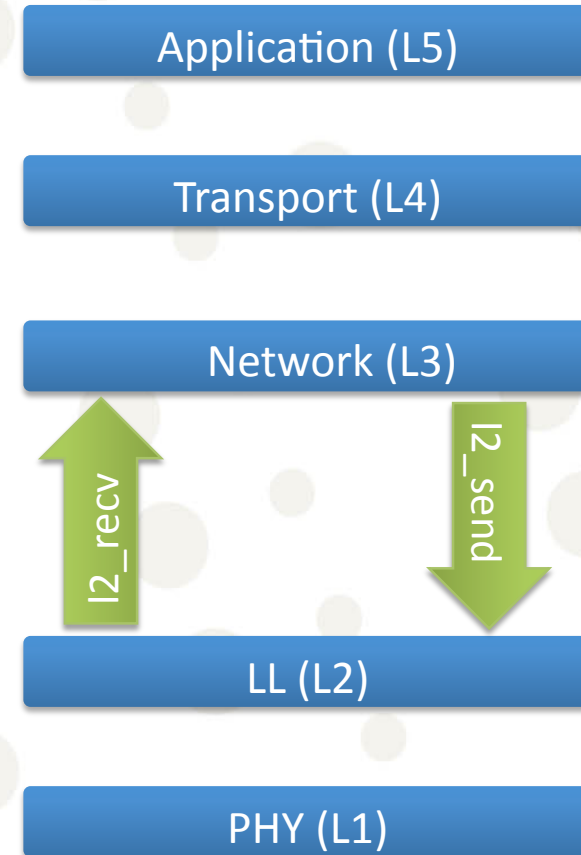
Program's Structure (PHY)

- On the top of existing *IP layer on linux* machines
- Other L1 functions provided to LL:
 1. ***int l1_send (int device, const char * buf, int length);***
 2. ***void l1_handle_event ()*** that uses *recvfrom ()* which calls either *l1_linkup ()* or *l2_rcv ()*



Program's Structure (Link Layer)

- Maximum **100 bytes** frames! (both DATA and ACKs)
- L2 functions:
 1. ***l2_init ()*** to support multiple simultaneous links
 2. ***int l2_send (int dest_mac_addr, const char * buf, int length)*** If sending frame is within the sliding window, save the frame of the window and return 1 If not then return 0
 3. ***void l2_rcv (int device, const char * buf, int length)*** received frames to be delivered to higher layer protocols without LL header by calling *l3_rcv ()*



(Delivery == Devilry)

– **What to include in the delivery:**

A. Design document (written with editors e.g. LaTeX; pdf-only submission)

1. A discussion about flow control, and the advantages of the more advanced sliding window protocols
2. How the program is designed?
3. Function calls map and details about headers, formats, etc
4. Consisting files (C files, header files, etc)

B. Program code (with inline documentation)

1. Documents all variables and definitions
2. What each function does?
3. Input and output parameters
4. Global variable changes
5. Returns values
6. Other characteristics that are important to know about (e.g. error situations)

(Delivery == Devilry)

When to deliver? Before Friday 17 February 2012 23:59

Q&A?