# Lecture 9/4/18 – UPD / TCP/ ID

## Protocols

### UDP (User Datagram Protocol)

Unreliable, packets delivered out of order, connectlioness

### TCP (Transmission Control Protocol)

More reliable. Packets are delivered in order. Connection Oriented. Overhead determines place / order.

### IP (Internet Protocol)

IP is in the network later

Essentally responsible for routing (host to host packet delivery)

Translates between multiple data link protocols like ethernet wifi etc

#### IP Datagrams

IP Provides connectionless, unreliable delivery of IP datagrams

Connectionless. All datagrams are independent of each other. Ordering is not guaranteed.

Unreliable: no guarantee datagrams are even delivered, let alone ordered. It makes a “best effort attempt:

#### IP Addresses

Ip at network layer but must be able to talk to other devices on different mediums (ex Iphone to wrired server – also why MAC at different aer tha IP)

IP addresses are logical addreses – four octets (32 bits) IPv4. Not called a byte because not everything used 8-bit bytes. Four numbers one 0 to 225.

Includes a network portion and a host portion. Used because routing is kinda like a hierarchy.

Sometimes a big network will use one IP address but route different packets to different ports.

Class A to D – more HostIDs to less HostIDS

#### Subnet Masks

Lolwhat?

#### Ramifications

Class As were needed for bigger networks but sometimes people got those first??

#### Network / Host IDS

Network IDs are assigned by a central authority – both ICANN and IANA

Host IDs are assigned locally by a systems administrator

Network ID and host ID are used for routing purposes.

#### Format

128.113.0.2 ([www.rpi.edu](http://www.rpi.edu))

Dotted decimal notation – 128.11.0.2 translates to 10000000.01110001.00000000.00000010

#### Subnet Addresses

Organizations are able to further subdivide its avaiale address space into subnets

For example, clump nearby machines into their own subnet

Breaks address into 10 | NetID | Subnet ID | HostID

#### Conclusion

Connectionless delivery – each datagram handled individually.

Unreliable – no guarantee

Fragmentation + reassembly – Hardware MTU (maximum transmissible unit)

Routing

Error Detection

#### Fragmentation Stuff

## UDP

Transport protocol (communication between two processes)

UDP uses IP to deliver datagrams to the proper host

Uses ports to provide additional specification

Format – the only header information is source port, destination port, length, and checksum.

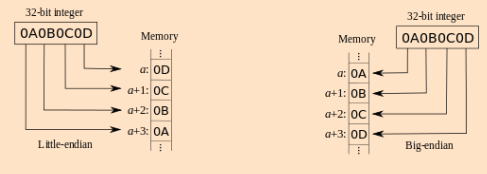
## TCP

Also a transport layer protocol. Adds reliability. Full Duplex (Both sides can talk to one another at same time), connection-oriented, byte-stream. Lots of header

## UDP Sockets

### Endianness

Sometimes right hand bit is the biggest bit. Othertimes it’s the left hand bit.



When hosts exchange single byte data types, no problem. With bigger words, transport between big vs little machine sis problematic.

### Sockets

Building blocks for network enabled programs

Bind() – associates socket to address and port.

Recvfrom() – stops program and waits for data??? Lolidk – can return 0 whch means peer has closed connection.

Sendto()

# Lecture 9/7/18 – Trivial File Transfer Protocols

## Motivation

◼ Suppose you want to send a file

◼ Too big to wrap into one packet

◼ Transport layer doesn’t know your file system

◼ Is it text? Binary? Rare case of Presentation layer being noticeable

## TFTP

TFTP inherently offers very little, the entire RFC is only 10 pages!

Very small header – two bytes of op code. Rest is data.

### Op Codes

Read Request (RRQ) – let me see a file

Write Request (WRQ) – let me send a file

DATA (DATA)

### RRQ/WRQ structure

### Initiating

Server reeives a RRQ/WRQ with the Transfer ID (TID) of the requesting host

Either the server sends…

DATA block 1 with matching destinations TID and its own source TID

Or sends ERROR with the approporiate code

Transfer ID is just the port number

### Modes

Netascii is just ascii

Receiver must translate

Mail sends to a host/relay with the intention of emailing data

WRQ only

We won’t use this – it’s dumb.

### Data Structure

2 bytes for op code and 2 byres for block number

Rest is for data. Data is at max 512 I guess, so if a data packet contains less than 512 bytes, it’s the end

### ACK Structure

2 bytes for op code

2 bytes for block #

### Error Codes

There’s a bunch.

Two bytes for op code/ 2 bytes for error code

String

0 to terminate

# Lecture 9/14/18 – Signals

## Overview

Basically software interrupts I guess?

Platform Agnostic --- ooooo hitting me with dat terminology DAYUM

## Portability

It’s not? Or at least it wasn’t until now ☹

## Handlers

When a singal handler is looking or signal X or handling one, it cannot receive more?

## Terminating Processes

Once a child terminates, it sends a signal to the parent. The parent reviews it before taking action.