

TFTP Assignment - don't worry about extra stuff. Only use RFC1350



UDP is connectionless so packets do not have an inherent order. ~~Any~~ Any ordering of the packets must be done in the application

TFTP was originally designed to be used to boot medialess systems back when these systems could not locally handle a TCP network stack until after boot.

Transport protocol is host-to-host protocol

IP → get to host      Port - get to application  
↳ 16 bits (0 - 65535)

Functional Languages like LISP (List Processing)

- are non mutative
- minimal types
- no declarations
- just symbol processing
- minimal primitive operations
- all user defined operations are functions
- inside out evaluation for nested operations

### List operations

(CAR) ↳ what is the first item in the list

(CDR) ↳ what is the rest of the list

(CONS) ↳ add item to list as the first item on that list

# TCP - Transmission Control Protocol

↳ reliable Stream service

↳ Stream is an ordered collection of bytes

→ end to end

→ in-order Stream

↳ no structure to stream (that's up to the application)

→ flow control end-to-end

→ sliding windows

→ sequenced by bytes

→ sliding window allows pipelining

→ reduce delay in WAN

→ increasing network throughput

→ sliding window units are octets (bytes)

→ acknowledgements used to tell sender the data

has been received, without error, in order

→ acknowledgements are also used as part of  
flow control

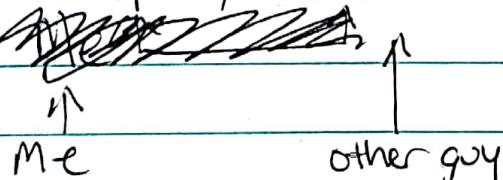
negotiated

~~selective~~

selective  
repeat

goback n

Source port, destination port, 16bit port

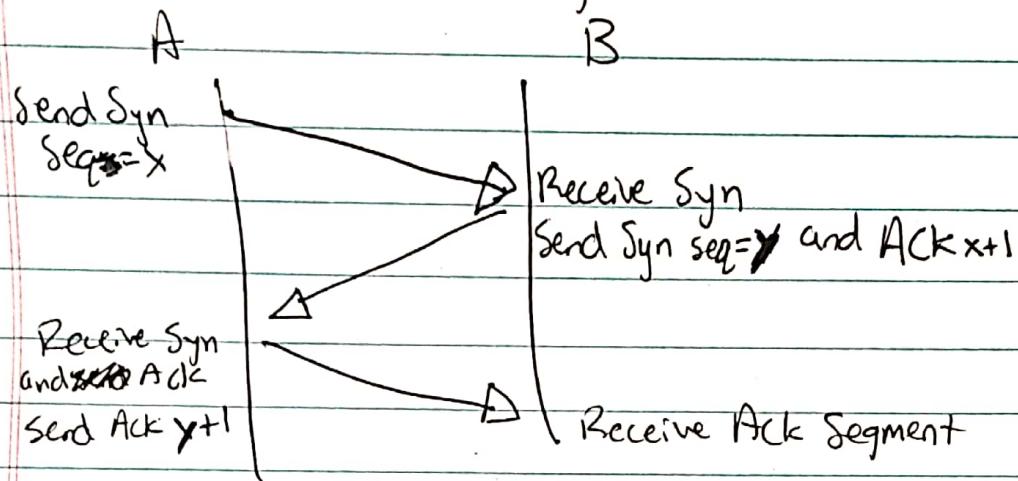


Different connections

Chrome (192.168.1.x, 2000) ↔ (192.168.1.y, 80)

Firefox (192.168.1.x, 2001) ↔ (192.168.1.y, 80)

## Negotiate Starting Sequence number



## TCP Header

Source Port, Dest Port, Sequence #, Ack #,  
Checksum, Window Adv., LEN, Options, Padding, etc.

↳ Code Bits which fields are meaningful?

↳ Urgent Pointer → consume these bytes out of order  
↳ software interrupt

# TCP - Transmission Control Protocol

↳ reliable Stream service

↳ stream is an ordered collection of bytes

→ end to end

→ in-order stream

↳ no structure to stream (that's up to the application)

→ flow control end-to-end

→ sliding windows

→ sequenced by bytes

→ sliding window allows pipelining

→ reduce delay in WAN

→ increasing network throughput

→ sliding window units are octets (bytes)

→ acknowledgements used to tell sender the data

has been received, without error, in order

→ acknowledgements are also used as part of  
flow control

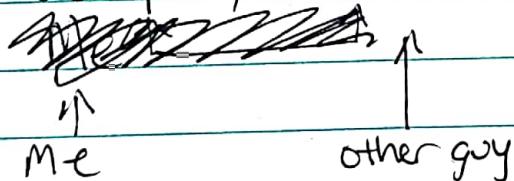
negotiated

~~seq~~

Selective  
repeat

go-back-n

Source port, destination port, 16bit port

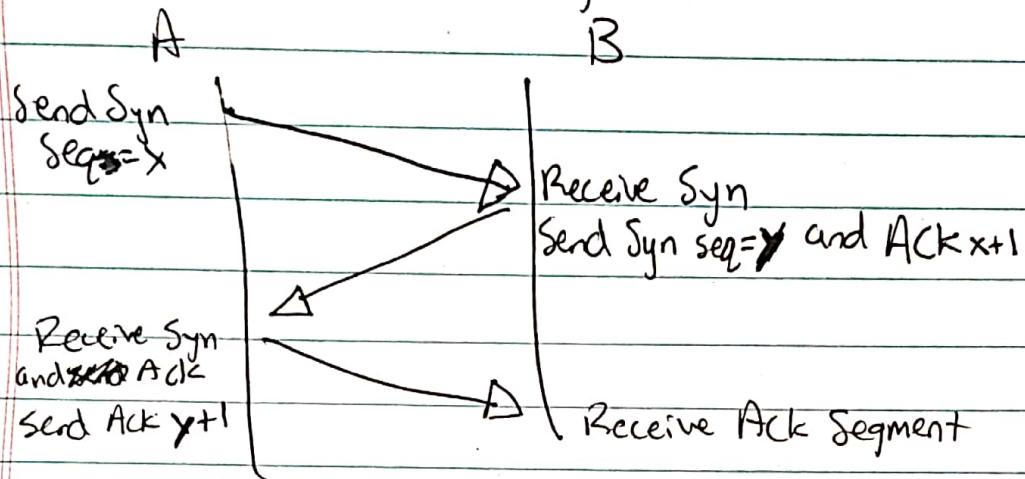


Different connections

Chrome (192.168.1.x, 2000) ↔ (192.168.1.y, 80)

Firefox (192.168.1.x, 2001) ↔ (192.168.1.y, 80)

## Negotiate Starting Sequence number



## TCP Header

Source Port, Dest Port, Sequence #, Ack #,  
Checksum, Window Adv., LEN, Options, Padding, etc

- ↳ Code Bits which fields are meaningful?
- ↳ Urgent Pointer → consume these bytes out of order
  - ↳ software interrupt

- TCP Slow Start - each increases ~~speed~~<sup>window size</sup> 50% each time
- ↳ missed ack, cut back by half
  - ↳ once ack is missed, cut back by half
  - ↳ then only increase window size by 1

Jump to ~94

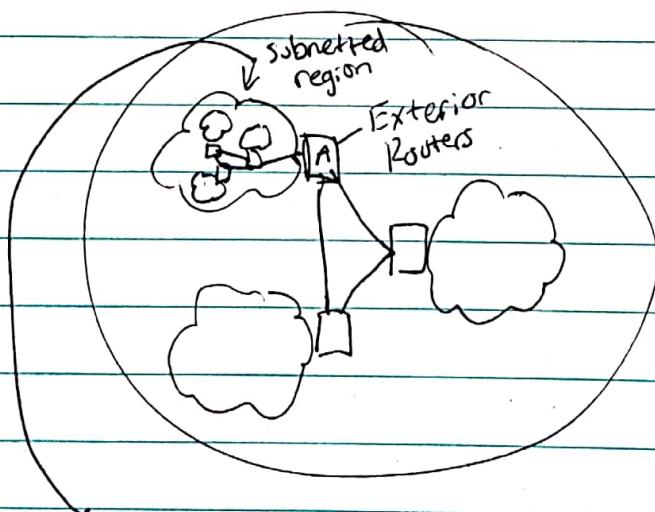
## Internet Routing



~~RIP~~ ~~IGRP~~ OSPF, BGP, GGP, etc

- ↳ each region is only responsible for itself
- ↳ each region is abstracted
- each region can route however it pleases
  - ↳ border points (access points) ~~need~~ need to be understood

~~difference~~ - single entry point is exterior gateway and runs exterior gateway protocol



Exterior routers must speak exterior protocol and interior protocol

Interior to A, but exterior to the subnet

## OSPF (usually interior)

- Can calculate separate routes for different types of services
- netmasks
- multicasting to reduce network load
- equal "cost" routes to distribute traffic
- simple authentication available

## BGP (exterior because it scales well)

- policy based routing
- collection of autonomous systems
- route is a sequence of AS numbers to a system

## AS categories

- Stub AS - single connection to one other AS
  - ↳ carries local traffic only
- Multihomed AS - connection to more than one AS
  - but does not carry transit traffic
- Transit AS - connection to more than one other AS
  - ~~but does not carry~~, but subject to policy restrictions, carries local and transit traffic

## Internet Exchange Point [MAE]-Metro Area Ethernet

- ↳ physical location where two or more network parties connect their networks (wires)
- many different implementations
  - ↳ many racks owned by separate companies
  - facility jointly owned by all parties

CDN - Content Delivery Network

↳ temporary content hosting to distribute content

Internet Exchange Maps can show you where the IEPs and CDNs are

# Declarative Languages

Father(John, Mary)

Build Knowledge Base  
Ask Questions

? - Father(John, Jessica)

No

? - Father(James, Mary)

No

? - Father(John, X)

X = Mary

? - Father(John, Mary)

YES

Unification and Exhaustive search (pattern match)  
on a Conjunctive Normal Form of knowledge Representation

Facts - Father(John, Mary)

Rules - Parent(X, Y) :- father(X, Y)

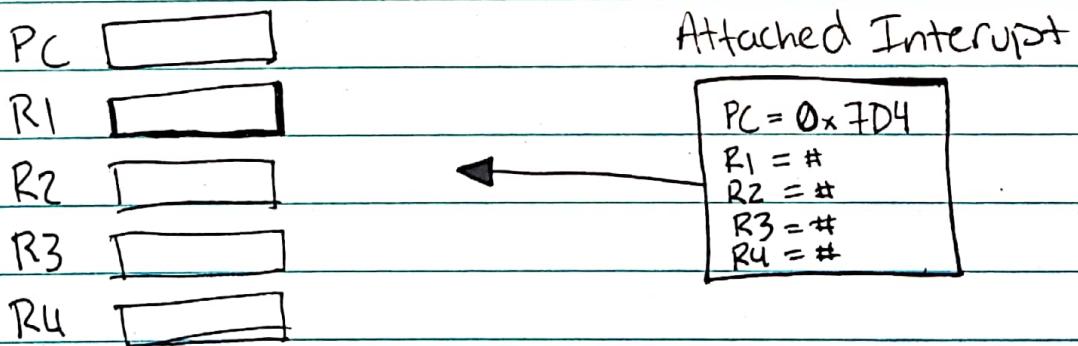
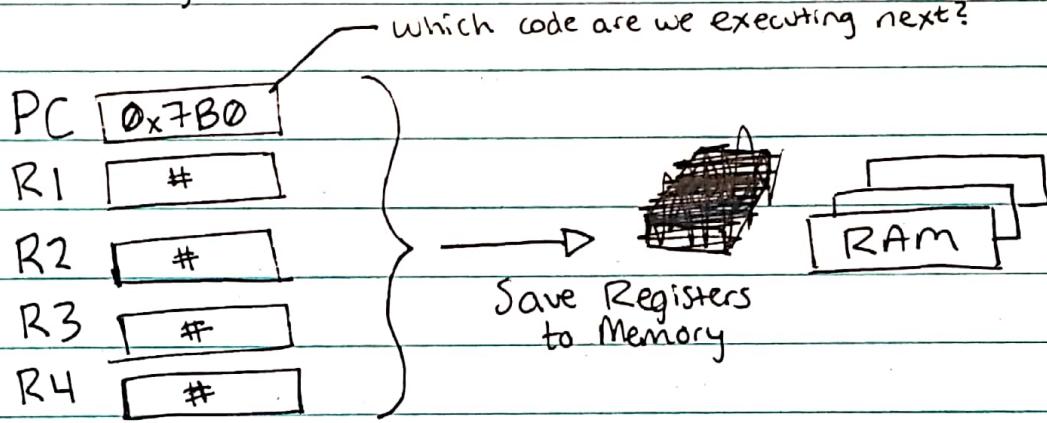
Parent(X, Y) :- mother(X, Y)

GrandParent(X, Y) :- parent(X, Z), parent(Z, Y)

MEMORY LOCATION

0x7A2 drawPattern( ) {  
...  
}

0x7D4 emergencyShutoff( ) {  
...  
}



DHCP largely replaced BOOTP

RARP allowed a node to request IP configuration from a RARP server

BOOTP would also supply location of boot server and the name of the bootfile (TFTP because unbooted nodes didn't have a full network stack (no TCP))

BOOTP doesn't scale well

↳ address is locked to a single device

## Network Management

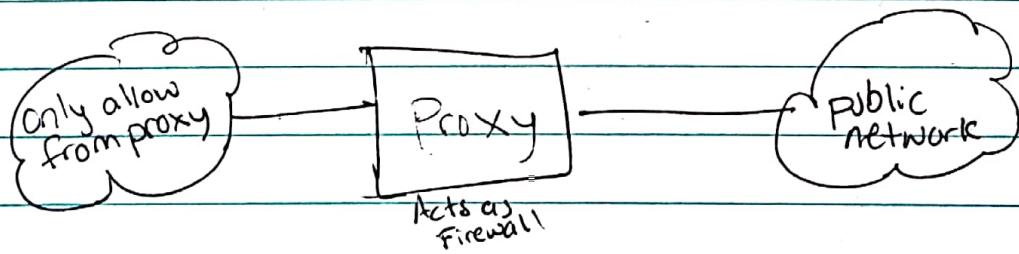
An internet part 7 onto 8

↔ Proxy ↔

Deep Packet Inspection

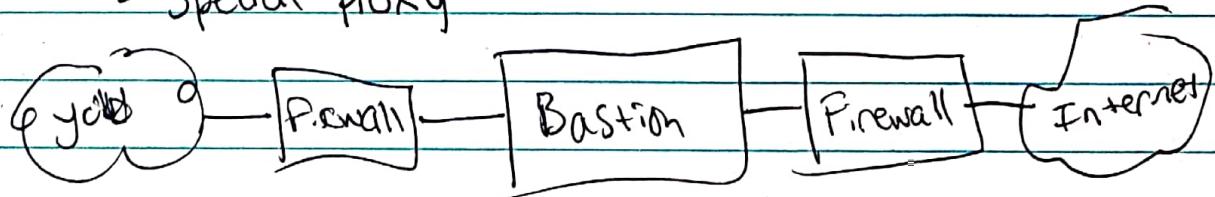
↳ look beyond packet header data

↳ CPU intensive on large scale deployments



DMZ - Bastion Hosts

↳ Special proxy



Adds extra layer an attacker has to get through

## Express Tunnel (VPN)

↳ IPsec - standard for forming these tunnels

## Intrusion Detection System

↳ Virus Scanner (hash checks, etc.)

↳ machine learning (pattern) ↳ no 0 day catches  
↳ not very successful violations

## NAT - Network Address Translation

↳ one to one inside to outside IP mapping

### Dynamic NAT

↳ one to many based on port numbers

↳ port forwarding

↳ breaks one-to-one design of IP addressing

Final Thursday at 8AM

↳ Start in Zoom

## Express Tunnel (VPN)

- ↳ IPsec - standard for forming these tunnels

## Intrusion Detection System

- ↳ Virus Scanner (hash checks, etc.)
- ↳ machine learning (pattern) ↳ no 0 day catches
  - ↳ not very successful violations

## NAT - Network Address Translation

- ↳ one to one inside to outside IP mapping

## Dynamic NAT

- ↳ one to many based on port numbers

- ↳ port forwarding

- ↳ breaks one-to-one design of IP addressing

Final Thursday at 8AM

- ↳ Start in Zoom

June 1<sup>st</sup> earliest

- Technical Guide Refresh

Prototyping Card

- ↳ finish if possible → document if not
- ↳ export project files

Backplane

- ↳ document
- ↳ export project files

Build

Test

Program → Test

Program → Test

Big Test

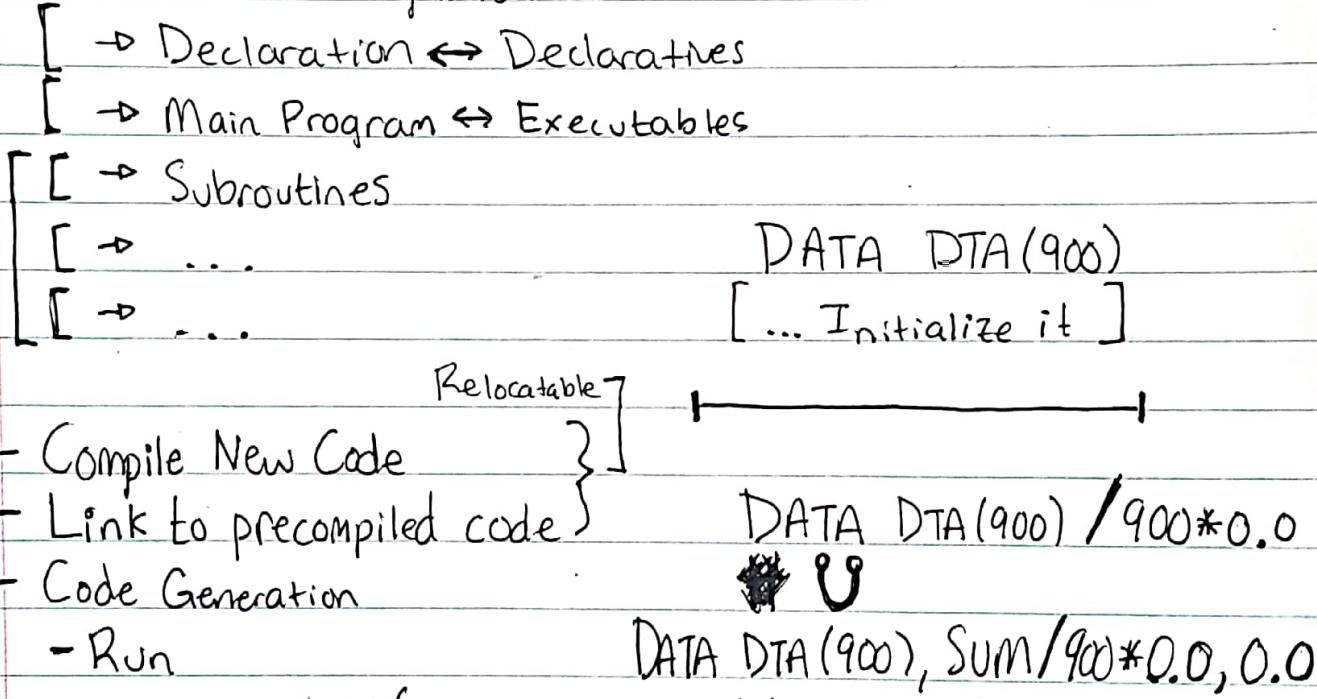
Zip folder

Uploads too

Structure - Pseudocode interpreters made problem solving and programming easier

- Floating point ops in pseudocode → hardware
- Libraries, a set of subroutines, became very popular → Compiling
  - Reused standardized pack of program cards
  - ↳ "Can it be linked in memory instead so we don't have to reinterpret it everytime we use it?" yes

### Structured Organization



→ Compiling and Linking Were Separate Steps

↳ Now Linking is often abstracted

- Sentences did not have a terminator

if(x.EQ.A(I)) k=1

U

if(x.EQ.A(I))

\$ k=1 p=2 Q=5

Statement Types:  
Declaration, Selection, Control, Executable

Lots of IF and GOTO statements tends to lead to unreadable Spaghetti code, which violates the Structure Principle.

GOTO(L<sub>1</sub>, L<sub>2</sub>, L<sub>3</sub>...L<sub>K</sub>), I

↳ GOTO (10, 20, 30, ..., 50), I \* J  
① ② ③ ④-8 ⑤

→ So if I \* J is equal to 2, goto line 20

~~Observe~~

Violates Syntactic Consistency Principle

GOTO N, (L<sub>1</sub>, L<sub>2</sub>, L<sub>3</sub>, ..., L<sub>K</sub>)

↳ GOTO N, (10, 20, 30, 40, ...)

↳ Like things should look the same. Unlike things should not look the same.

Weak Typing - N is used as an integer but it is also used as an address

↳ violation of Defense in Depth.

↳ If an error is not caught at one stage, it should be caught at another stage.

DO 20 I=1, N

:  
:  
:

Nest Loops [E] → Good, Clear  
Overlap Loops [E] → Bad, Confusing

CONTINUE

↳ Much more optimizable than a bunch of IF GOTO

↳ use register for ~~when~~ incrementing

→ Array indexing

↳ C, C+V, V+C, C\*V, etc. also more optimizable

↳ must be "fixed discrete changes"

Flags → pay attention to "fragment offset" or not  
→ ~~zero~~ "do not fragment" flag

IPv6 → no fragmenting allowed  
→ MTU discovery to find safe packet size  
↳ ~~max~~ Minimum Transfer Unit

- ↳ Aimed for fixed sized header
  - ↳ Checksum removed to ease router load
  - ↳ "Next Header" points to another header to access additional options and features

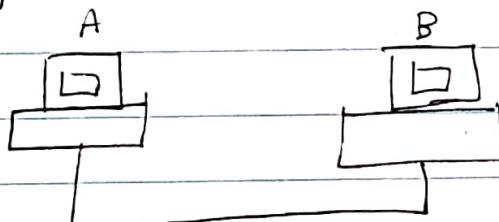
## Routing the Internet

MAC Addresses

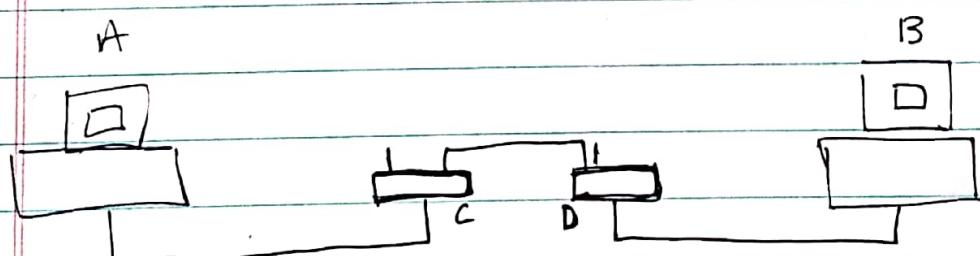
and IP Addresses

will only match in the

same frame in this scenario ~~A can't~~ A can't deliver to B



A can directly deliver the frame to B  
- A can ~~not~~ deliver to anything connected by layer 1 or 2 devices, but nothing higher like a router.



A cannot deliver to B, A can deliver to C, which can deliver to D, which can deliver to B.

Frame would have A's and C's MAC addresses, but the packet only has source and destination IPs  
MAC Addresses for frames      IP Addresses for Packets

If two nodes share the same netid (and subnetid) they must be on the same physical network

- Layer 1 and 2 devices do not split into subnets

Routers by definition are on multiple networks

Both Hosts and Routers must make routing decisions

- Hosts need to know whether they can send direct or which if any router to send it to
- hosts only handle packs that are ending or starting at that host
- if a host gets a packet for a different host it must discard it
- Routers need to know which physical networks its on so that it can deliver to nodes or forward through another router

Routing Decision based on lookup table

- local: netid, subnetid
- wide area: netid only
- routers only need to know the next step, not the entire path to the destination
- routers do not need entries for all netids in their tables
- they may know the routes to some netids and a default location to send other packets to

Lab 108

East

Across

From

Vending  
Machines

DO 20 I=1,N

~  
~

run this code to line 20  
with I increment towards N,  
then keep going past 20

20 Continue

SUBROUTINE DIST(D,X,Y) ....

D = X - Y

IF (D.LT.0) D = -D

RETURN

END

CALL DIST(R,Q,P)

~~#P#~~ All fortran subroutines

are pass by reference

↳ more memory efficiency

↳ more dangerous

Procedural Problem Solving → Abstraction

↳ Modularization

↳ Libraries

CALL CHG(3)

↳ 3 becomes a variable

name that stores the  
value 2.

SUBROUTINE (I)

I = 2

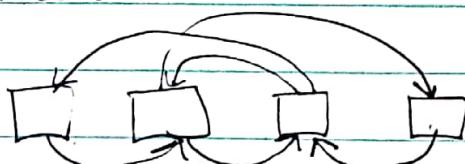
RETURN

END

↳ now everytime "3" is used, the variable is referenced, not  
the constant. Mars Probe was lost to a similar bug

Activation Record → when I'm done doing this loop or  
subroutine, where should I return to?

→ NO RECURSION



Linear Structure → Local Scope

↳ Function names only thing that's global

The Linear Structure ~~and~~ and Local Scope added a knew barrier though  $\rightarrow$  Subroutines cannot share information anyway other than parameters/arguments. They also had no type checking so multiroutine programs could easily result in garbage answers

### COMMON [Declaration]

$\hookrightarrow$  global declaration

### EQUIVALENCE —

$\hookrightarrow$  array combination?

Activation Record  $\rightarrow$  Fixed Size

$\hookrightarrow$  only one ~~type~~ Activation Record for a given subroutine

$\hookrightarrow$  if a subroutine was called twice,

its activation record was replace/written over, thus making recursion impossible

### Activation Record

#### Parameters

DL - Dynamic Link  $\rightarrow$  where to return to  
SL - static link  $\rightarrow$  where are the common variables

### Data Structure

Primitives - Integer, Real, Boolean, Complex, Fraction

word oriented 16 bit

Integer  $\rightarrow$  +, Explicit Conversion

$\hookrightarrow D = \text{IFLOAT}(I)$

Arrays only non primitive data type, maxed out at 3 dimensions

### Automatic Declaration

COUNT = COUNT + 1

$\hookrightarrow$  declares "COUNT" to be an entirely new variable

DO 20 I=10,20

↳ Automatic Declaration → [DO20I] = 10.20

Syntactic Structure - Fixed Format

- Ignore Blanks
- no nesting → except loops (3 levels)
- no keywords → 2 pass compilers

1<sup>st</sup> Generation Language - FORTRAN

- ↳ m/c oriented syntax, data, and control → high efficiency
- ↳ dependent on "GOTO"
- ↳ primitives and arrays, that's it
- ↳ weak typing
- ↳ no keywords
- ↳ disjoint scope

Assignment #1 is up → Read Only Folder

## Algorithmic Problem Solving

Universal Language - ACM-GAMM

↳ Designed Algol → use mathematical notation

→ usable for publications

→ translatable to machine code

↳ Grammar for Syntactic notation

↳ Short and Concise

↳  $E ::= C$

$E ::= E + E$

....

↳ Hierarchical Structure

1<sup>st</sup> Gen Languages

were using

static allocation

↳ maximum possible

memory space

↳ IF [ ] THEN

[...]

ELSE

[...]

  x x x   x x x x   x x x   x x x x x x x x

IF  
THEN  
[...]  
ELSE  
[...]

Keywords → begin, end → free formatting

↳ still no statement terminator ; )

↳ no I/O prescribed in the language

↳ must be done through libraries

↳ Similar to the idea of

drivers; it was too machine

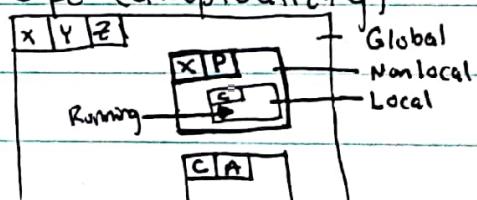
dependent to make native

Assignment →  $A := B$     ( $A \leftarrow B$ )

↳ keywords are part of grammar

↳ Concept of Scope (local, nonlocal, global)

↳ Contour Diagram (Scope (conceptualizing))



Impossible Error Principle → it is better to make an error entirely non-possible, than to detect ~~the~~ error after it has been made

- No implicit declaration → typos were less <sup>of</sup> a problem because a typo in a variable name was quickly caught during compilation
- Dynamic Allocation leads to records on the activation stack being different sizes. This makes the activation stack much harder to manage in memory
  - ↳ also an additional hinderance to allocation speed
  - ↳ static allocation happens once → fast program
  - ↳ dynamic allocation happens all the time → slow program, fast programming

Dynamic Scoping → hard to do in practice

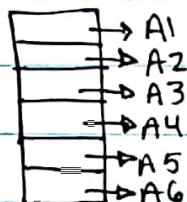
during execution you will be at some level of nesting. This means you are some nest distance from the static scope level.

### SAVE

sd, offset  
↳ static distance

### Array-Display

↳ pointers to activation records at that moment of execution



↳ keeps some nest but separate scope blocks actually separate

Static Scoping and Structured Programming was more reliable and easier on the developer

Efficiency reliability

↳ best use of resources

Economy

↳ best use of human / social resources

Elegance

↳ Aesthetics, Readability

Blocks were allocated and deallocated as necessary

↳ ex of Responsible Design Principle

↳ enable a user to describe specifically what they need

Language did not permit any type other than primitives

↳ eventually supported COMPLEX, FRAC, LONG, DOUBLE

↳ made compilers more costly even when some of these new features weren't even used

↳ Algol said no, we'll only use "REAL"  
and that's it



Mathematical Publication Implementation  
(Reference)

Algol → zero, one, infinity principle

↳ 0 → not allowed

↳ 1 → can happen only once → method names, variable names

↳ ∞ → can happen as much as required

→ any length of labels

→ any size of dimensions

→ any level of nesting

for loop it's baroque

Aesthetics

Localize Cost Principle  $\Rightarrow$  don't pay for more than you need. Algol For loops (baroque) violate this because it's a feature you may or may not need

Use of grammar to describe language lead to mathematical formulation of language

$\hookrightarrow$  Chomsky's Language Hierarchy

0  $\rightarrow$  Recursively enumerable

1  $\rightarrow$  Context sensitive

2  $\rightarrow$  Context free

3  $\rightarrow$  Regular

(YACC)

BNF  $\rightarrow$  Language Check

2<sup>nd</sup> (Algol)

2 generation languages

$\hookrightarrow$  machine independent

$\hookrightarrow$  block structure

$\hookrightarrow$  free format (white space  
label size)

$\hookrightarrow$  no I/O (native to language)

$\hookrightarrow$  retained some baroque

features

$\hookrightarrow$  no user defined types

Kleene Notation

$\hookrightarrow *$   $\rightarrow$  0 or More

$\hookrightarrow +$   $\rightarrow$  1 or More

$\hookrightarrow ?$   $\rightarrow$  0 or 1 ( $[ \dots ]$ )

199.17.161.140

199.17.161.150 direct send - same network

199.17.161.140

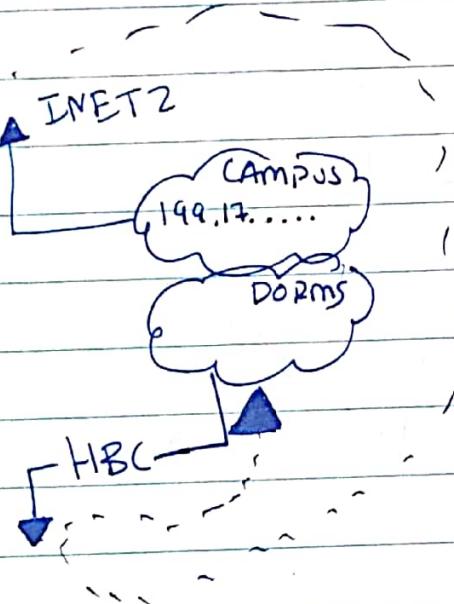
199.17.162.150 → not same network

↳ check for host specific route

- subnet specific route

↳ net specific route

↳ default route



No direct communication from

academic campus buildings and dorms

↳ net specific route could fix this  
but if such a route was

accidentally published to the

internet, bad things would happen

## IP Routing Algorithm

1. Extract Dest IP from packet

2. Ext netid

3. if it matches any directly connected network, send it

...

...

...

Network Layer never sees mac addresses

## Internet Control Message Protocol - ICMP (Network Layer)

IP is connectionless protocol

↳ network layer to layer communication

↳ packets can be lost

↳ —————— arrive out of order

↳ —————— have errors in them

↳ —————— get duplicated

→ This is all correct protocol behavior

Strictly speaking IP doesn't have to do anything about this but it would be nice to know when it happens

ICMP is a mechanism for reporting that exceptions have occurred, not correcting

↳ heuristic only

↳ if you don't receive a report, ~~don't assume~~

this does not mean there was no exception

→ most ICMP messages originate in router

→ no guarantee the messages will be delivered or sent

→ If ICMP message throws an exception, do not send further ICMP packets

↳ do not send an automated response to an automated response (infinite loop risk)

## ICMP - RFC 792

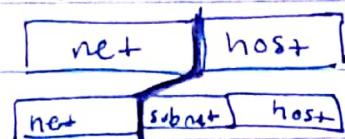
- ↳ describes the protocol
- ↳ formal
- ↳ ASCII ART and Fixed Size Fonts for early ease of printing

## ICMPv6 messages

- ↳ neighbor solicitation and neighbor advertisement
- ↳ ARP built into ~~ICMPv6~~ ICMPv6
- ↳ lack of fragmentation in IPv6 makes many ICMPv4 messages useless for ICMPv6 so they were removed

# Transparent Gateways (Subnet addressing)

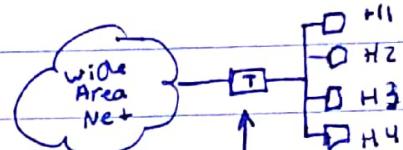
Class B



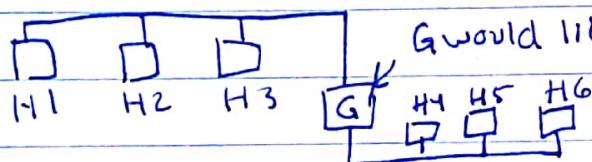
Routing based on net id's  
(gateway - gateway)

Normally Subnetting is  
a local issue, not visible  
to the Wide Area Network

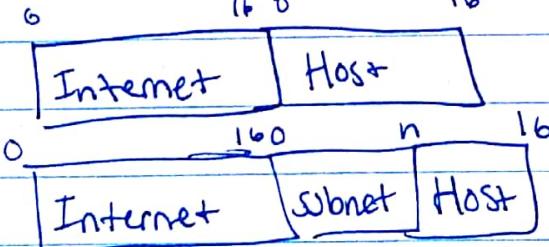
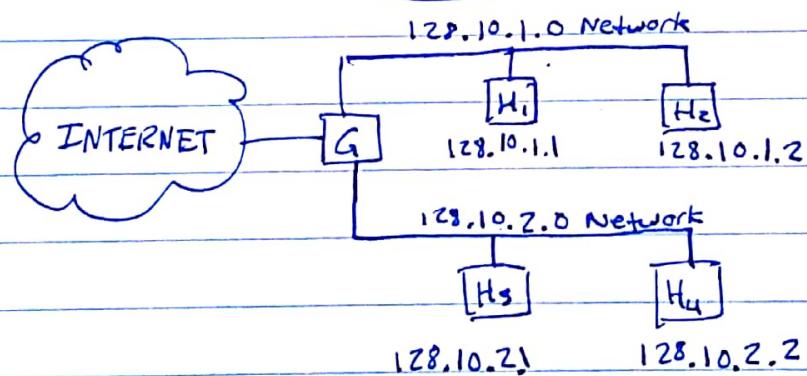
Internet "looks" like a tree  
L take bits from host id bits (local part)



put host here instead of  
router



G would lie about an ARP request from H1  
to H5 and pretend to be H5.



→ Subnet does not have to  
be the full octet, it can be  
1-15 bits

134

29

64

0

010  
subnet

00000

00000000

↳ 3 bit subnet id  
allows for 8 subnets

Subnet 2  
looks like 64  
because the bits  
don't line up with  
the address parsing

Subnet Masks shows which bits are  
relevant to the net id and subnet id

255 255 2<sup>4</sup> 0

11111111 11111111 11100000 00000000

↳ 8 subnets

/19

↳ 19 leading

1's

Masks are of local interest only

Rewrite of Routing Alg.

- Extract Dest. IP Address from datagram
- Compute IP Address of <sup>destination</sup> Network
- Send direct if possible (same network)  
else

-

Source Address  $\rightarrow$  Host A

Destination Address  $\rightarrow$  Host B

Host A  $\wedge$  Mask  $\rightarrow$  zeros out the host id  $\rightarrow$  src  
(my) only have access to your own mask

Host B  $\wedge$  Mask  $\rightarrow$  zeros out the host id  $\rightarrow$  dst  
(my)

SRC  $\oplus$  DST  $\rightarrow$  compares the two  
 $\hookrightarrow$  all zeros  $\rightarrow$  same Subnetwork  $\rightarrow$  send direct  
 $\hookrightarrow$  not all zeros  $\rightarrow$  different network or subnet.  
 $\hookrightarrow$  use routing table to send it

Supernetting (CIDR Block) RFC1338

C - Classless

How can dense routing tables

I - Inter

be made smaller?

D - Domain

$\hookrightarrow$  blocking

R - Routing

Move mask right to make blocks smaller  $\Rightarrow$  (subnet)

$\hookrightarrow$  left to make bigger (supernet)

Uncovered bits - first address low order block bit must be

100...  $\odot$   $\rightarrow$  odd address means 1 blocksize

10.0.0.0 - 10.255.255.255 Class A

172.16.0.0 - 172.31.255.255 Class B

192.168.00 - 192.168.255.255? Class C

$$3x + y + 3z = 4$$

$$x + 2y + z = 1$$

$$x + y + z = 2$$

$$\left[ \begin{array}{ccc|c} 3 & 1 & 3 & 4 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 2 \end{array} \right]$$

↓

$$\left[ \begin{array}{ccc|c} 3 & 1 & 3 & 4 \\ 0 & 1\frac{2}{3} & 0 & -\frac{1}{3} \\ 0 & 1 & 1 & 2 \end{array} \right] - \frac{x_0}{x_1}(x_0) - \frac{y_0}{y_1}(y_0), - \frac{z_0}{z_1}(z_0), - \frac{\alpha_0}{\alpha_1}(\alpha_0)$$

↓

$$0 \quad \frac{2}{3} \quad 0 \quad 2 - \frac{4}{3}$$

$$\left[ \begin{array}{ccc|c} 3 & 1 & 3 & 4 \\ 0 & 1\frac{2}{3} & 0 & -\frac{1}{3} \\ 0 & 0 & 1 & -\frac{5}{6} \end{array} \right] - \frac{y_2}{y_3}(y_2), - \frac{z_2}{y_3}(z_2), - \frac{\alpha_2}{y_3}(\alpha_2)$$

$$\frac{\frac{5}{3}}{\frac{2}{3}} = \frac{5}{3} \left( \frac{3}{2} \right) = \frac{15}{6}$$

$$\frac{5}{3} \left( \frac{3}{2} \right) \left( -\frac{1}{3} \right) = -\frac{15}{18} = -\frac{5}{6}$$

$$\left[ \begin{array}{ccc|c} -3 & 2 & -6 & 6 \\ 5 & 7 & -5 & 6 \\ 1 & 4 & -2 & 8 \end{array} \right] \quad -2, 3, 1$$

$$\left[ \begin{array}{ccc|c} 1 & 4 & -2 & 8 \\ 5 & 7 & -5 & 6 \\ -3 & 2 & -6 & 6 \end{array} \right]$$

$$\left[ \begin{array}{ccc|c} 1 & 4 & -2 & 8 \\ 0 & -13 & 5 & -34 \\ 0 & 0 & 0 & 30 \end{array} \right] \quad \downarrow \quad \begin{aligned} & -\frac{x_1}{x_0}(x_0), -\frac{x_2}{x_0}(y_0), -\frac{x_3}{x_0}(z_0), -\frac{x}{x_0}(\alpha_0) \\ & -\frac{x_2}{x_0}(x_0), -\frac{x_3}{x_0}(y_0), -\frac{x}{x_0}(z_0), -\frac{x_2}{x_0}(\alpha_0) \end{aligned}$$

$$\left[ \begin{array}{ccc|c} 1 & 4 & -2 & 8 \\ 0 & -13 & 5 & -34 \\ 0 & 0 & -6.6154 & -6.6154 \end{array} \right] \quad \downarrow \quad \begin{aligned} & -\cancel{13y}, \cancel{5z} - \frac{y_3}{y_2}(y_2), -\frac{y_3}{y_2}(z_2), -\frac{y_3}{y_2}(\alpha_2) \end{aligned}$$

$$-6.6154z = -6.6154$$

$$\textcircled{z=1}$$

$$-13y + 5z = -34$$

$$-13y + 5 = -34$$

$$-13y = -39$$

$$\textcircled{y=3}$$

$$x + 4y - 2z = 8$$

$$x + 12 - 2 = 8$$

$$\textcircled{x=-2}$$

## Princ. of Progra. Lang.

- Read Prefix Section
- 2.2, 2.3, 2.5
- Ch 1 all
- 3.1, 3.2, 3.3
- 4-9 Section 1

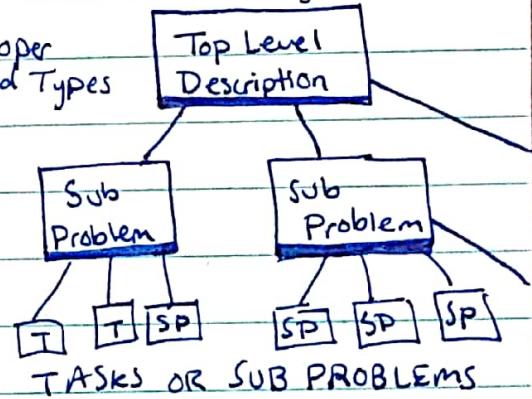
Open (unit=45, File=abc.txt)

Write(45, ...) ... → Write(45, I5) Data

Read ? ↳ if Data = 5 output: \_\_\_\_ 5

## Pascal

- ↳ made to teach non-scientists to develop programming solutions
- ↳ and create a process for methodical problem solving
  - ↳ "Top Down" → Waterfall Method (eventually)
- Teach, Methodical Problem Solving, Developer Defined Types



## Radio

HAM Bands: .75m, 2m, 3m, 6m, 13m ...

Longer Wave Lengths have a higher chance of Reflection Propogation

HF - High Frequency



VHF - Very High Frequency



UHF - Ultra High Frequency



HT - Hand Transmitter

MT - Mobile Transmitter

BS - Base Station

more power  
and higher  
cost

} Lower  
Frequencies

Non-Commercial Communications

Call Signs, Jargon (QSY)

Power Limits

# Adu. Netcomm. Lab 2

## Part 1

IP address: 199.17.162.21 (Our IPv4)

Ping 192.145.254.134 ✓

|         |                  |           |
|---------|------------------|-----------|
| Ping -r | fails to respond | (Windows) |
| Ping -R | —————            | (Linux)   |

Set time to live to one

↳ receive packet fail message from router

↳ increase TTL by one and repeat

|               |     |
|---------------|-----|
| Responding IP | TTL |
|---------------|-----|

|              |   |
|--------------|---|
| 199.17.162.1 | 1 |
|--------------|---|

|               |   |
|---------------|---|
| 199.17.175.10 | 2 |
|---------------|---|

|               |   |
|---------------|---|
| 199.17.175.43 | 3 |
|---------------|---|

|       |   |
|-------|---|
| ————— | 4 |
|-------|---|

|       |   |
|-------|---|
| ————— | 5 |
|-------|---|

|       |   |
|-------|---|
| ————— | 6 |
|-------|---|

|       |   |
|-------|---|
| ————— | 7 |
|-------|---|

|       |   |
|-------|---|
| ————— | 8 |
|-------|---|

|       |   |
|-------|---|
| ————— | 9 |
|-------|---|

|       |    |
|-------|----|
| ————— | 10 |
|-------|----|

|       |    |
|-------|----|
| ————— | 11 |
|-------|----|

|       |    |
|-------|----|
| ————— | 12 |
|-------|----|

|       |    |
|-------|----|
| ————— | 13 |
|-------|----|

|                 |    |          |
|-----------------|----|----------|
| 192.145.254.134 | 14 | (Target) |
|-----------------|----|----------|

## MTU discovery

- Set ~~Set~~ Do Not Fragment flag
- force packet size to small enough size

~~odd difference~~

- if <sup>Ping</sup> response increase size and try again
- if ~~if~~ Need to Fragment response decrease size and try again

→ shrink range until MTU found

MTU = 1472

## Part 2

Linux traceroute uses UDP which does not work after ~~sometimes~~ no response

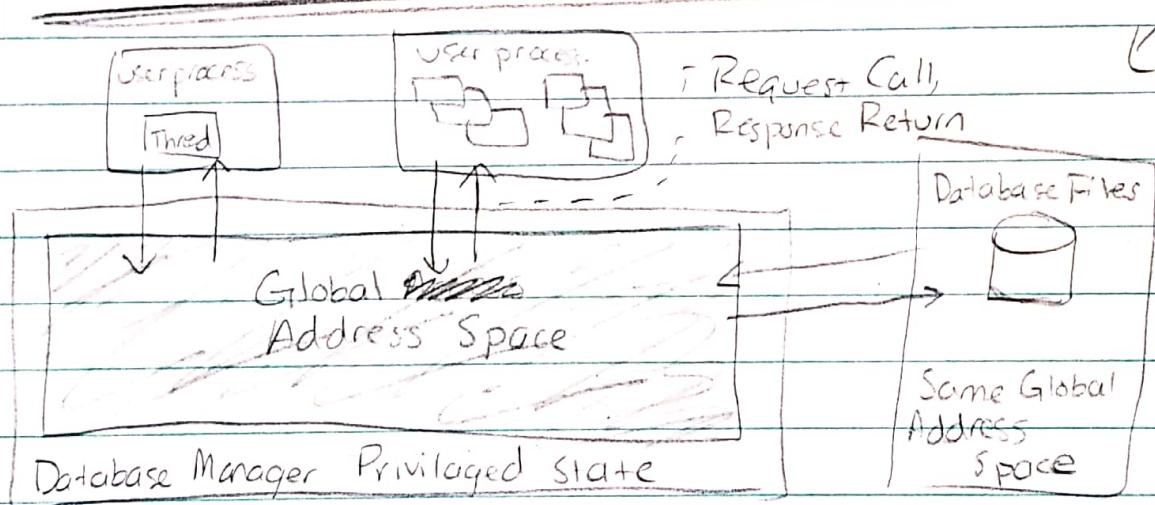
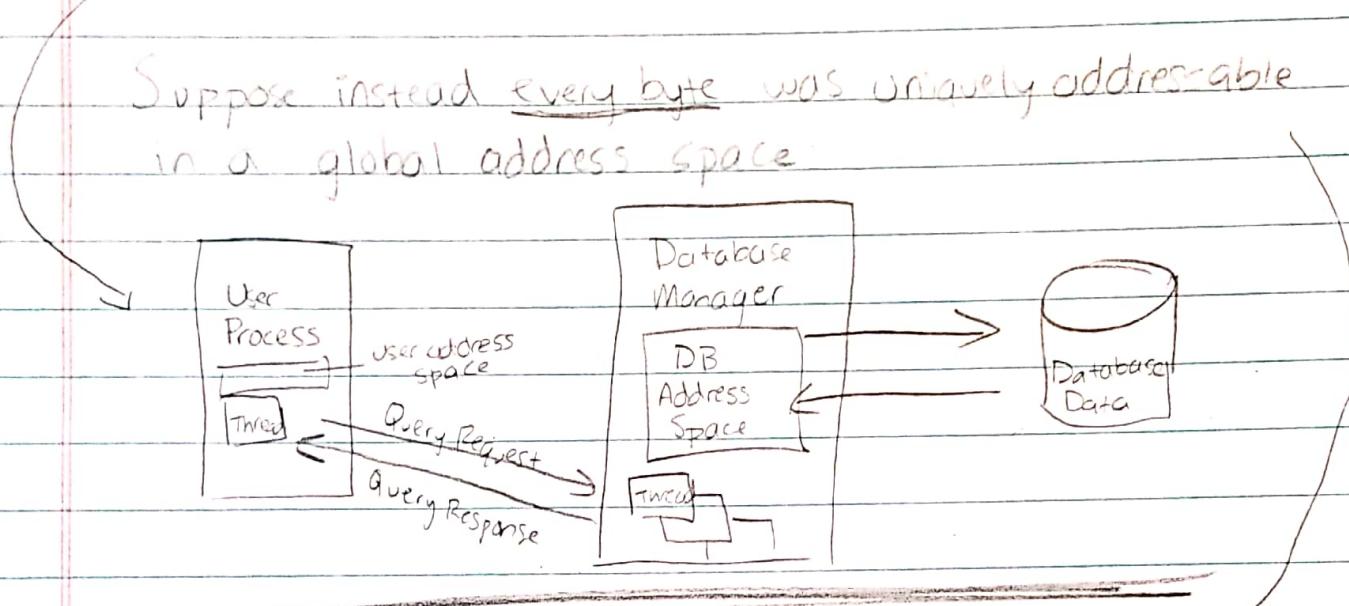
Effective Address is just a number

16 bits, 32, 48, 64, etc.

→ Every process has its own effective address space

↳ Security says these have to be separate

Suppose instead every byte was uniquely addressable  
in a global address space



We still need to stop a process from accessing other process' memory data. So each process is forced to work only within the box it has permissions

↳ ~~yes~~ an address represents the "Capability" to access something

|      |                |             |     |
|------|----------------|-------------|-----|
| Type | Type-dependent | SLS Address | TAG |
|------|----------------|-------------|-----|

Valid operations leave the tag set

Invalid ops set it to off

↳ without the tag it is considered a null pointer

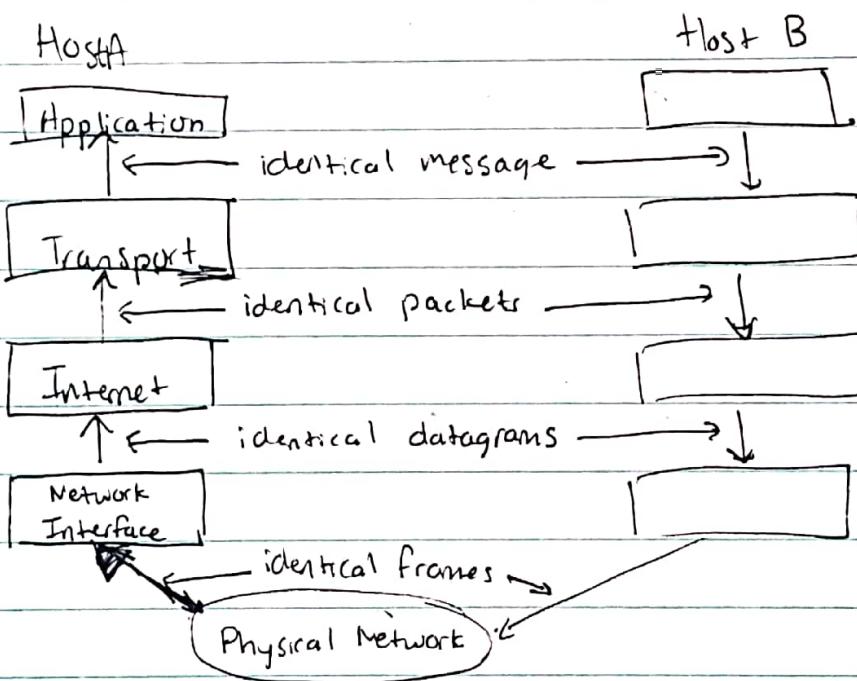
↳ null pointer use may cause a hardware interrupt

→ acts kind of like a key. If you shouldn't have access to that memory location, you shouldn't have the key in the first place

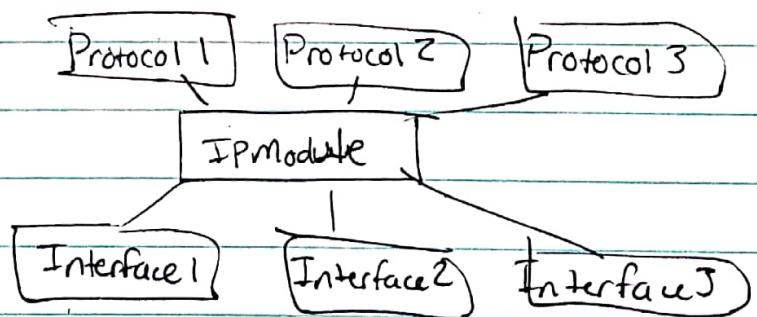
traceroute → reports back ip → usually ~~is~~ the approached ip

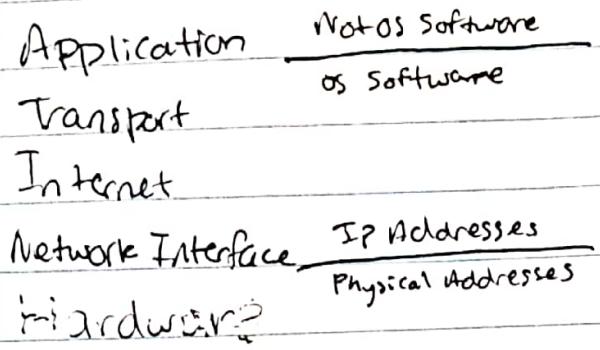
1994 - routing tables were getting too big, non linear growth  
wasn't sustainable

## How Data Passes Through Protocol Layers

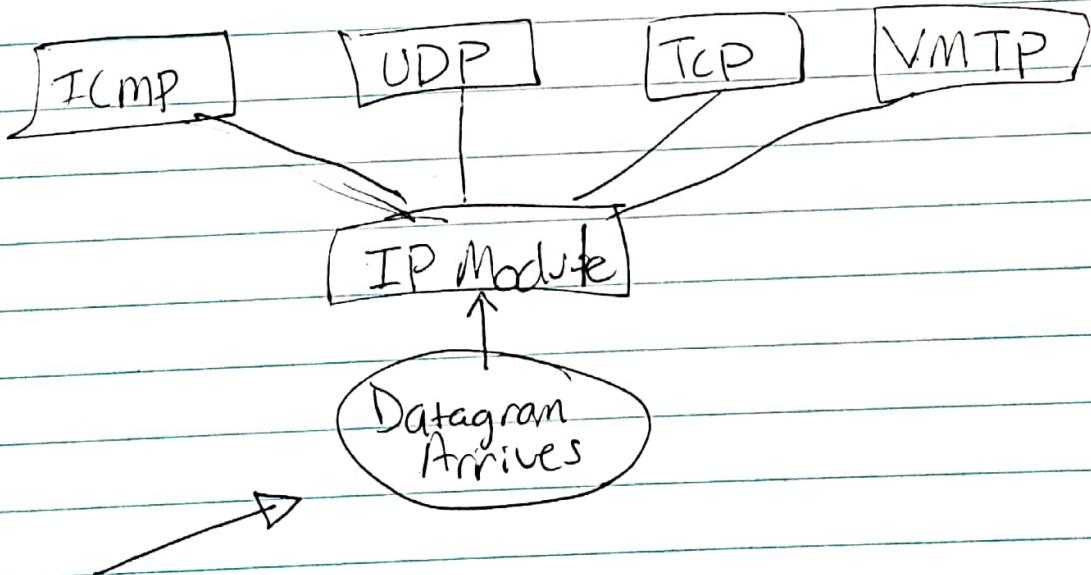
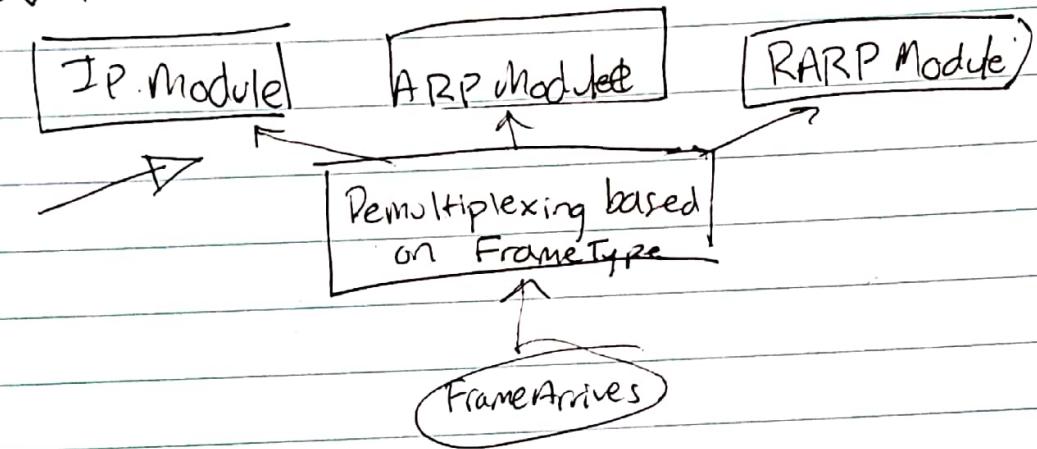


IP Module is  
essentially a ~~is~~ two  
way multiplexer between  
the interfaces and your  
protocol stacks





IPv4



The Innovators

The Pentium Chronicles

Side-effect: Static Scoping  
overrides pass by value

Gen 1

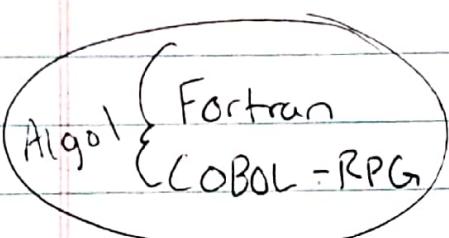
Weak Type

Machine Oriented

Gen 2

Grammar

Structure Programming



PLI (IBM)

extend languages

Pascal Structure

Input, Output  
Files

→ Not case-sensitive

→ Program [Program Name] (I/O Streams)

→ Free format

→ Statement terminator

→ procedure [Name] ( ) [ ]

declaratives

Const: → declare here constants

Type: → { user defined types }

Var: → { define variables }

begin

executables

:=, =, <, >, <=, >=, <>

end;

Pascal -  
- easy to use and teach  
- methodical problem solving  
- feedback - secure  
- broad set of applications

→ Syntax and semantics clearly  
defined and used grammar

→ Compiler gives feedback

↳ nondeterministic

↳ line by line <sup>syntactic</sup> error checking

→ Static Scoping, Top down design

→ New types definable by developer

↳ application specific

↳ verified and enforced by compiler

→ all variables had to be initialized

Signature allowed for a violation of  
Top Down argument calling

Forward Procedure [Name](param.);

x:integer, var y: real  
(pass by value) (pass by reference)

$$x: 8 \otimes 5 \otimes 2 + 1 = 3$$

$$\text{Temp}[5] = 5$$

$$y: 1$$

$$5: 1 + 1 = 2$$

data 1: 1

2: \* 24

3: 1

4: 1

5: 1

6: 1

7: 1

8: 1

9: 1

10: 1

ret 1 1  
2 1

3 \* = 12

4 1

5 1

6 1

7 1

8 1

9 1

10 1

Out Temp: 1, 1, 1, 1, 5, 1, 1, 1, 1, 1

Out Data: all ones

Out Data: All Ones

Out Ret: 1, 1, 12, 1, 1, 1, 1, 1, 1

Out Data: 1, 24, 1, 1, 1, 1, 1, 1, 1, 1

Out Data: 1, 24, 1, 1, 1, 1, 1, 1, 1, 1

# Application Oriented Features

Niklaus Wirth 1968-1970 first pascal compiler

## Enumeration Type

↳ Ex. Days of the Week (1...7)  
↳ Type

days of week = 1...7; or months = (Jan, Feb, Mar, ...);

Var days of week today; months this month = Jan;

Array bounds check at runtime succ, PRED (+, --)

↳ Security Princ. Ex.

Fortran if ([condition]) then case ([expression]) of  
begin [statement(s)] ⇒ 1: begin  
end [ ] end  
else if ... nodefault until later ⇒ 2: begin  
end [ ] end  
end

Pascal all declarations at the beginning

↳ all dec. must be known before runtime

Procedure [name] ([param], [param], ...) // var → pass by reference  
{ declarations (local to proc.) }

begin  
...  
end

Function funcName(...): typename

{ [ declarations ] }

begin

funcName := [return value]

end

Type

range = 1..10;

difrange = 0..9;

Var

Data : Array[range] of real;

Data2 : Array[difrange] of real;

...

Data2 := Data does not work

↳ same structure but type naming was different

Pointer types

Type → a range of values and set operations

↳ address ← of a particular type

X : pointer of real;

new(x); allocating ... dispose(x);

\* X = 45; stores 45 at address X

Type

player = Record of

Age : Integer;

FName : packed array [1..10] of char

...

end

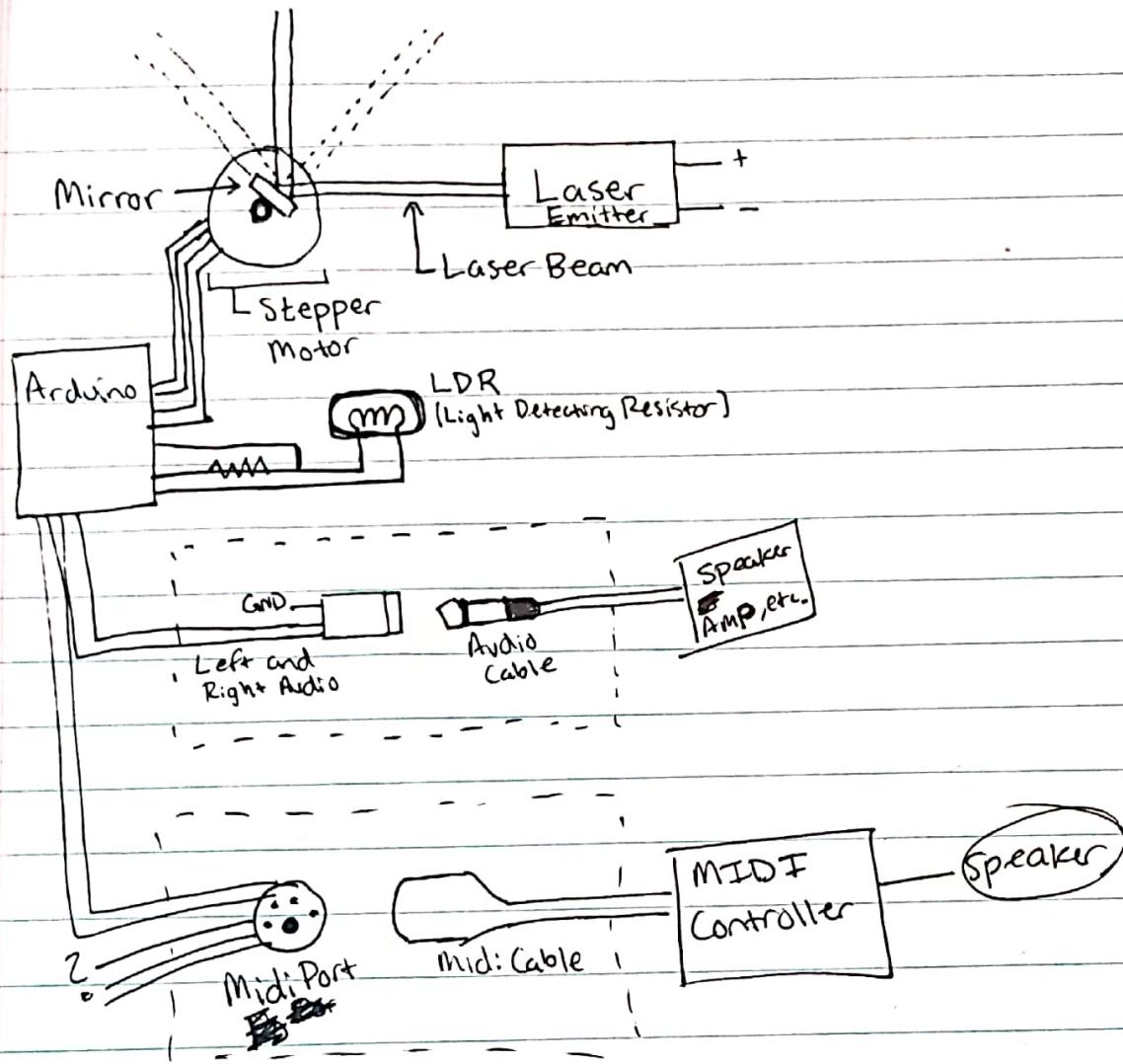
Var

Player : player;

Player. Age = 45;

Strong Typing

Type of an array included its dimension



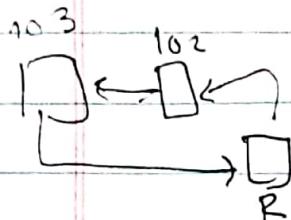
## Lab Post Discussion

### Error Types, Response Types

102.X      /24      no talk  
 103.Y      /24

/23  
 /24

) talk but no response  
 (general failure back)



if there had been a router

102 direct 103 through router  
 if there had been a router between (not symmetric)

Subnetting Netid

192.168.0.0

/24

C

4 subnets

/26  
 netid gets bigger

host id 6 bits

Host id cannot be all zeros (should not be allowed)

blocking

192.168.0.-

192.168.15.-

/24

block of 16 (4 bits) /20

( $\log_2(16) = 4$ )

00000000

00000000 00000000

0000 0000

0000 0000

0000 0000

0000 0000

1111 1111

1111 1111

1111 1111

0000 0000

1111 1111

1111 1111

1111 1111

0000 0000

252, 253, 254, 255

/22

1111 1111

1111 1111

1111 1111

1100 0000

1111 1111

1111 1111

1111 1111

1111 0000

True

header content, no format

IPv4 vs IPv6

32 bits

vs

128 bits

dotted decimal

hex

flat

hierarchical (by service provider)

127.0.0.1

::1

Class Based

~~no classes~~

255.255.255.255

No broadcast

224.0.0.1

ff02::1

(multicast)

nope

link-local

192.168.2.1, 10.0.0.1

fec0: ...

(site local)

— nope

anycast

(broadcast by service)

### Headers

Variable size, parsing

fixed size, format

TTL

HOP Count

Header checksum

too intensive so no checksum

Minimum Safe MTU (Hierarchical discovery)

Flow Label (Where are similar packets routing?)

### Type of Service

Application Layer  
stuff

Next Header → extention headers

↳ auth info

Neighbor Discovery (ARP)

ICMPv6 (Neighbour Solicitation)

MAC Addresses

↳ broadcast

↳ multicast

Hard to configure

"Self Configuring" (from Router Advertisement)

↳ Trackable, so Random is an option

route length

Austere       $\text{fn } x : = \text{expression} \text{ do } \text{exp}$  by  $\text{exp}$

begin  
...  
end

( like a for loop with an iterator )

repeat ~~X~~  
\* ---  
until()

dangling  
else  
problem

if ( ) then  
if ( ) then  
else  
... which if does  
the else follow?

### 3rd Gen Langs

Simple, efficient,

Clear Name & control structure

Shift from machine emphasis to application development

↳ Linkers came out of C (makefile)

1111 1111 1111 1111 1111 0000 0000 0000

1111 1111 1111 1111 1000 0000 0000 0000

2 192.168.

1111 0000 0000 0000

1111 1110 0000 0000

254 or 255 . □ 2 block

199.17.128.0 48 class C addresses

199.17.161.0 and 199.17.162.0

1  
128  
32  
160

1111 1111 161 → 1111 1111 162 →

10100001

80

10100000

166 10100110

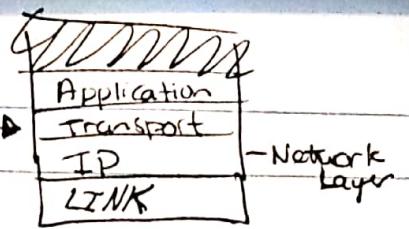
11111

XXXXXXXX XXXXXXXX XXXXX XXXX

11111111 11111111 11111110 00000000

~~IP Pack~~

Transport Layer



TCP - Transport Control Protocol

UDP - User Datagram Protocol

RDP - Reliable Datagram Protocol

- Others being developed, especially for realtime traffic

Transport Protocol is end-to-end

IP is connectionless (datagram)

Transport layer can provide many services regardless  
of that provided by the ~~IP~~ Network Layer (IP)

Process to Process talking

↳ don't care how it moves between hosts

Simplest transport service is the one most  
like the network layer

↳ in IP this is UDP (connectionless)

UDP - User Datagram Protocol

- connectionless protocol
- best effort delivery only
- can arrive out of order
- have errors in them
- no connection between multiple packets
- packets may take different routes
- no flow control (fast sender overwhelms slow receiver)

- Voice, Video, Gaming, Messages Small Enough To Fit  
In One Datagram

Need to be careful though because

- testing these Applications is often done on local network (LAN) where errors are infrequent
- Once deployed on on WAN many unforeseen behaviours may arise

WAN Emulators, use separate ISPs, cellular network

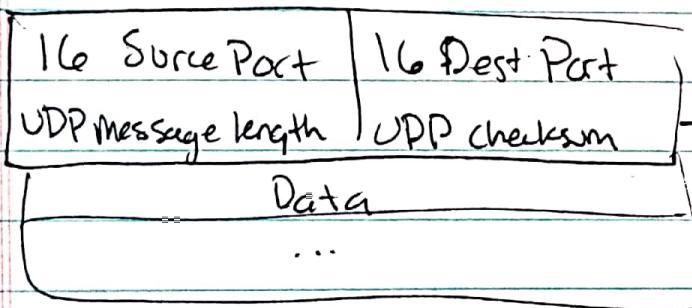
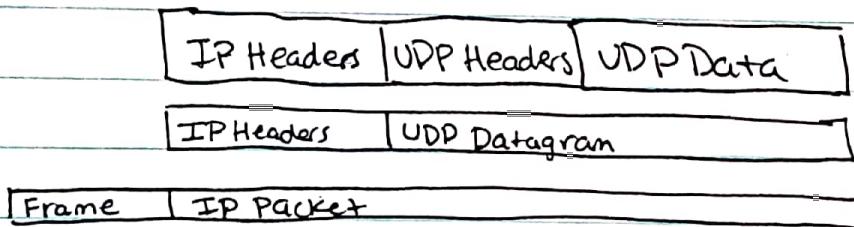
How do we identify which process needs the UDP packet?

↳ ports → 16 bits → (0-65535)

↳ source and destination ports

↳ origin

↳ receive



To compute this checksum it violates some network layering by looking at the IP header and comparing the destination IP address with your own when you receive a packet all 0s or all 1s

Port assignments are a policy issue not a protocol issue.

Usually there is a queue assigned to a port and received ~~pack~~ datagrams are put ~~s~~ in that queue. The process then pulls from that queue as necessary.

↳ OS determines how to handle the queue

Only one process per port/queue unless either process could handle it because only one process will get it

Compile → Link → execute

With so many versions of Pascal, you never knew if your code would run on the systems of another lab/company

So DoD developed Ada. It was sold out to various companies as a very standardized compilation language. Since the DoD does everything, the language had every feature anyone could need and started many concepts of parallelization. This made the language too big.

↳ The DoD also made each customer agree to only use Ada

Compiler → Linker → Execute

↳ these need have the same output for proper compatibility

One must provide the intended user with enough information to use the module and nothing more

## Modula-2 definition file, implemen. file

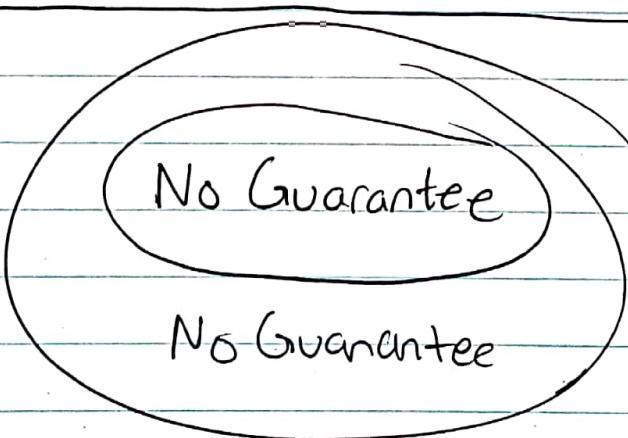
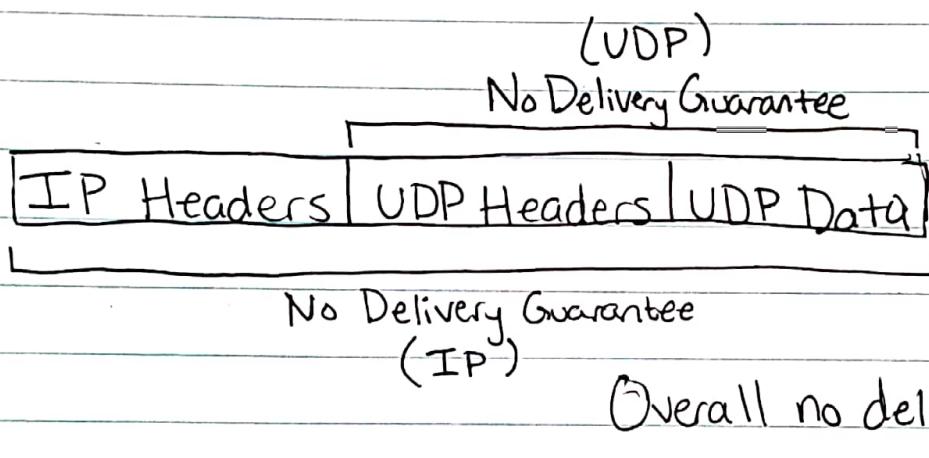
Compiling and Linking via command line  
can quickly fail due to typos or other mistakes.  
↳ IDEs automated this and it's a big help

→ exceptions

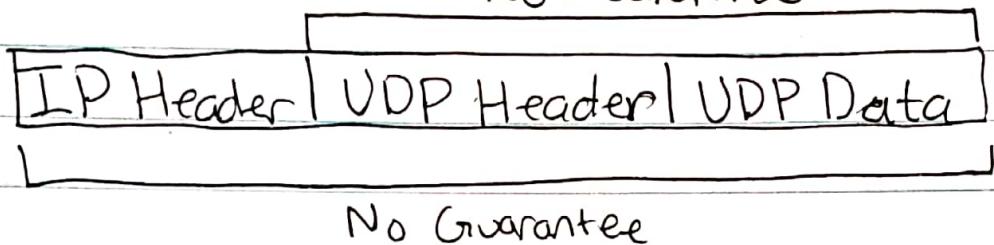
Video Games → combine modules to  
create "Objects" → birth of object oriented  
programming

Oliver Heaviside

IP Headers -

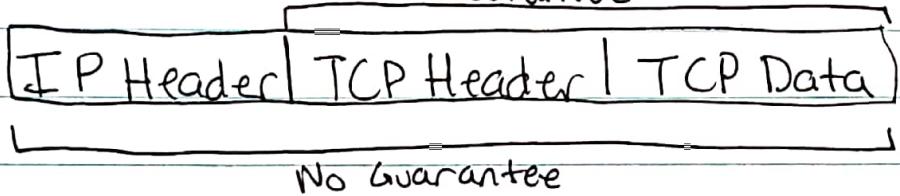


No Guarantee



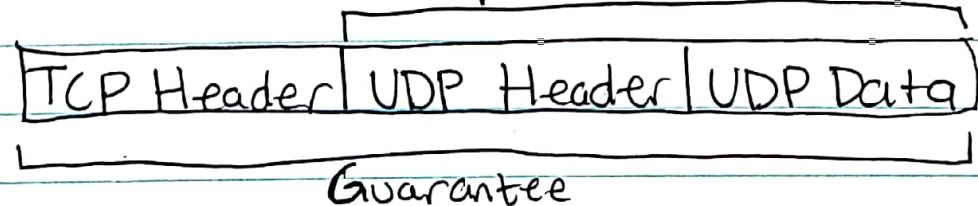
NG wrapped in NG → no guarantee on data

Guarantee



Guarantee wrapped in NG → guarantee on data

No Guarantee



NG wrapped in G → Guarantee on Data?

~~Cidr~~ (CIDR brought the routing table entries on the global viewpoint down.

- ↳ one entry for all Minnesota Schools
- Problem was this ~~was~~ added masks (starting address of block and mask)
- could push subnet to global net which was not intended
  - ↳ subnet in Rochester and one in Winona, make the internet route between the two
  - ↳ grows routing table a lot

|              |       |
|--------------|-------|
| CPU          | \$80  |
| GPU          | \$100 |
| Motherboard  | \$80  |
| RAM          | \$30  |
| Hard Drive   | \$30  |
| SSD          | \$50  |
| Power Supply | \$70  |
| Case         | \$60  |

\$500

+ labor  
+\$100

\$600

+ Windows

\$90 → \$700