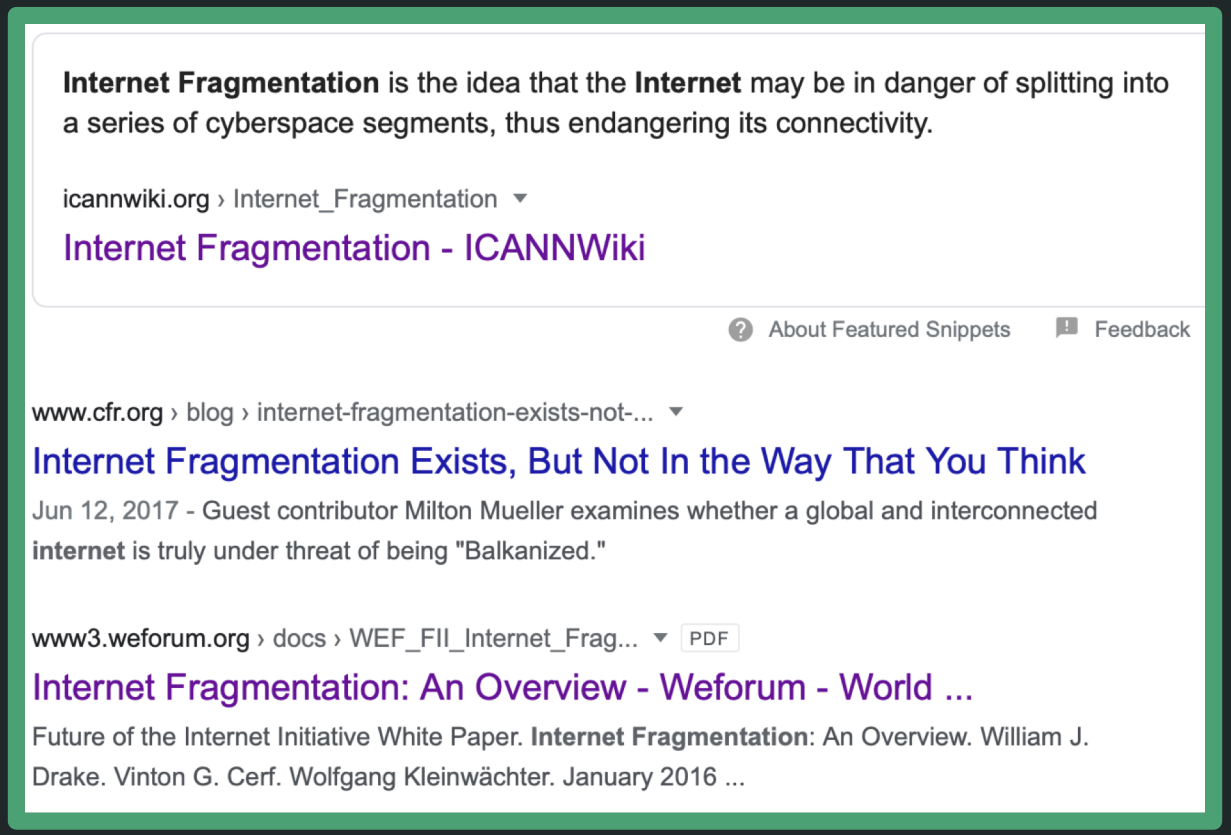
CS177 Lab #2

The fragmentation of the internet

Lukas Dresel, Apr. 13, 2020

What it’s not

Geopolitical Network Segmentation



What it is

IP Fragmentation

Fragmentation is simply taking big network packets and chopping them into smaller packets that can be sent individually.


12 Papers per Second

IP Fragmentation

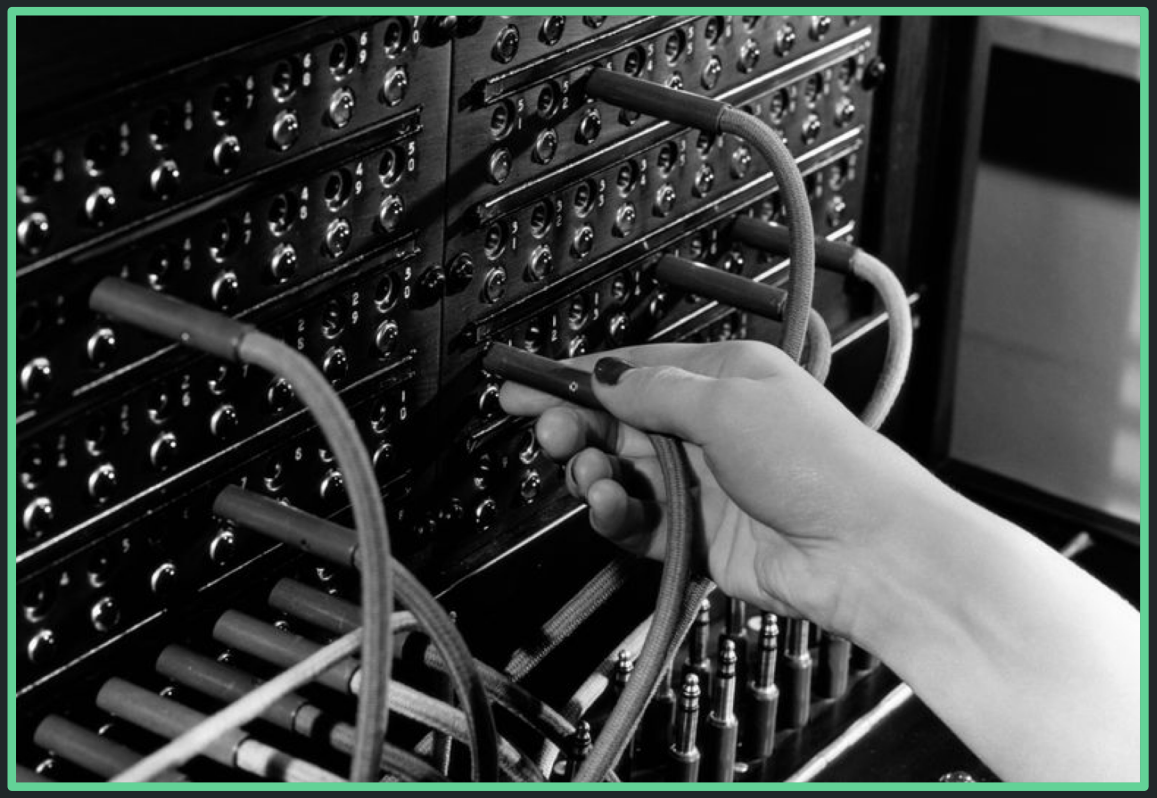
- ➔ Fragmentation is simply taking big network packets and chopping them into smaller packets that can be sent individually.
- ➔ By wrapping and re-packaging data sent across the network we can improve performance and be sure that every device can accept packets of any size regardless of underlying hardware.

History

Before the internet

Before the internet, what did we have?

Phone switching networks



Before the internet

- How did the phone networks work?
- Point-to-point communication (you can only call 1 person at a time)
- Directly over a wire, you are connected to the other side over one specific channel
- Singular, only one communication can go through that connection, while you're on the call no one else can use that line
- Switched directly by the operators, once the call is established the connection is fixed

The Internet Age

How does the internet work differently?

The Internet Age

How does the internet work differently?

- ➔ Packet based, communicating with someone is split into separate chunks
- ➔ Dynamic routing, different packets can get to the destination on different paths
- ➔ Much more efficient, the communication channel is not blocked while there is no data being sent
- ➔ Decentralized, if any specific part of the route drops out, the remainder of the packets will automatically be rerouted
- ➔ Load balancing, if any nodes have trouble keeping up with the amount of traffic, the traffic will be rerouted

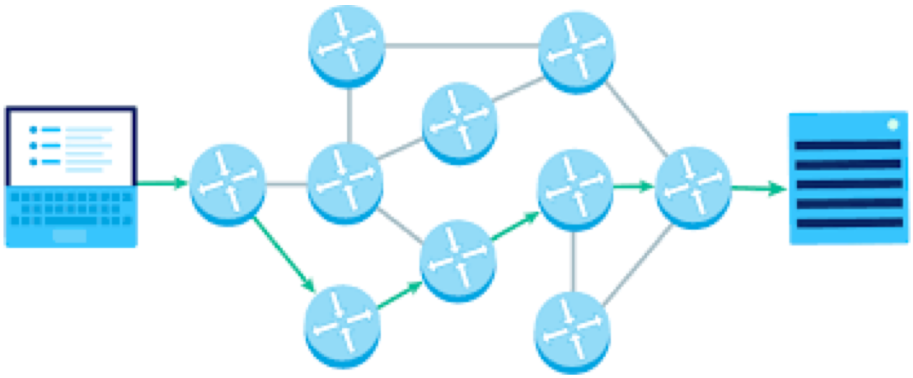

The Washington Post
Democracy Dies in Darkness
Your Internet is working. Thank these Cold War-era pioneers who designed it to handle almost anything

The Internet Age

Packets were a central piece that allowed the internet to be successful

> Even more elementally, the Netheads believed in an innovation called “packet-switching,” which broke from the telephone company’s traditional model, called “circuit switching,” that dedicated a line to a single conversation and left it open until the participants hung up.”
>
> - Washington Post

The Internet



How does it work?

Standards

How do we ensure this works?

Everyone can have their own machines with their own hardware, we need to ensure that we can communicate anyways.

Standards

How do we ensure this works?

Everyone can have their own machines with their own hardware, we need to ensure that we can communicate anyways.

Standards!

~~~~~~

RFC INDEX

0001 Host Software. S. Crocker. April 1969. (Format: TXT, HTML) (Status: UNKNOWN) (DOI: 10.17487/RFC0001)

0002 Host software. B. Duvall. April 1969. (Format: TXT, PDF, HTML) (Status: UNKNOWN) (DOI: 10.17487/RFC0002)

0003 Documentation conventions. S.D. Crocker. April 1969. (Format: TXT, HTML) (Obsoleted by RFC0010) (Status: UNKNOWN) (DOI: 10.17487/RFC0003)

0004 Network timetable. E.B. Shapiro. March 1969. (Format: TXT, HTML) (Status: UNKNOWN) (DOI: 10.17487/RFC0004)

0005 Decode Encode Language (DEL). J. Rulifson. June 1969. (Format: TXT, HTML) (Status: UNKNOWN) (DOI: 10.17487/RFC0005)

0006 Conversation with Bob Kahn. S.D. Crocker. April 1969. (Format: TXT, HTML) (Status: UNKNOWN) (DOI: 10.17487/RFC0006)

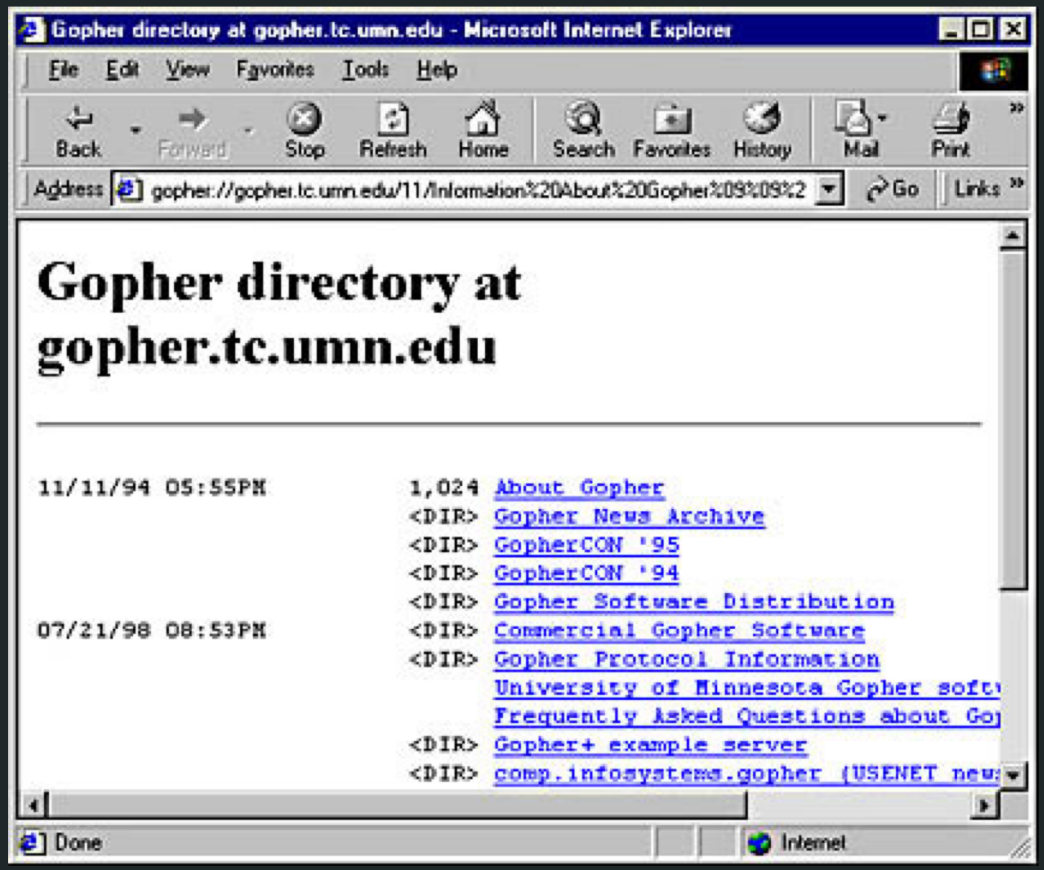
0007 Host-IMP interface. G. Deloche. May 1969. (Format: TXT, HTML) (Status: UNKNOWN) (DOI: 10.17487/RFC0007)

0008 ARPA Network Functional Specifications. G. Deloche. May 1969. (Format: PDF, HTML) (Status: UNKNOWN) (DOI: 10.17487/RFC0008)
~~~~~~

Standards

Some standards win, others lose.

RIP Gopher



Standards

Standards ensure all pieces of hardware on the internet can talk to each other without technical difficulties. They encode

- ➔ functionalities components must provide (receive packet, pass it on)
- ➔ limits on functionality (maximum packet sizes, timeouts)
- ➔ how to handle problems and errors
- ➔ how components interact with each other (protocols, etc.)

Standards

Standards ensure all pieces of hardware on the internet can talk to each other without technical difficulties. They encode

- ➔ functionalities components must provide (receive packet, pass it on)
- ➔ limits on functionality (maximum packet sizes, timeouts)
- ➔ how to handle problems and errors
- ➔ how components interact with each other (protocols, etc.)

This is where fragmentation comes in!

Standards

How do we ensure this works? Everyone can have their own machines with their own hardware, we need to ensure that we can communicate anyways.

Standards ensure all pieces of hardware on the internet can talk to each other without technical difficulties. They encode limits on functionality like **packet sizes**, how to handle problems and errors, how **components interact with each other**.

This is where fragmentation comes in!

Request for Comment (RFC)

RFCs describe all major public protocols used on the internet and specify what implementing a protocol means and what rules you have to follow:

- 1945 Hypertext Transfer Protocol -- HTTP/1.0. T. Berners-Lee, ... May 1996. (HTTP)
- 0793 Transmission Control Protocol. J. Postel. September 1981. (TCP)
- 0792 Internet Control Message Protocol. J. Postel. September 1981. (ICMP)
- 0791 Internet Protocol. J. Postel. September 1981. (IP)

Request for Comment (RFC)

RFCs describe all major public protocols used on the internet and specify what implementing a protocol means and what rules you have to follow:

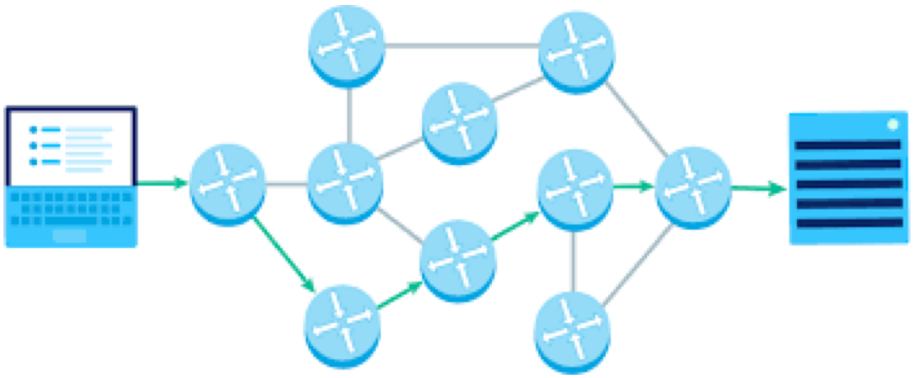
- 1945 Hypertext Transfer Protocol -- HTTP/1.0. T. Berners-Lee, ... May 1996. (HTTP)
- 0793 Transmission Control Protocol. J. Postel. September 1981. (TCP)
- 0792 Internet Control Message Protocol. J. Postel. September 1981. (ICMP)
- 0791 Internet Protocol. J. Postel. September 1981. (IP)

Request for Comment (RFC)

RFCs describe all major public protocols used on the internet and specify what implementing a protocol means and what rules you have to follow:

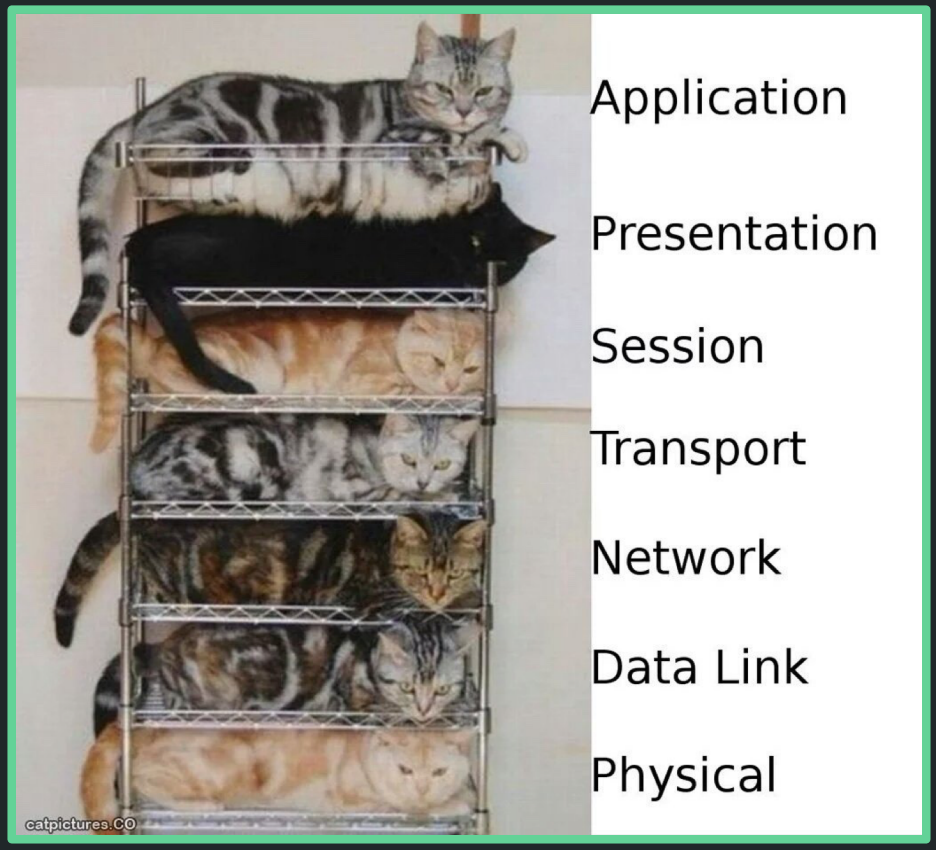
- 1945 Hypertext Transfer Protocol -- HTTP/1.0. T. Berners-Lee, ... May 1996. (HTTP)
- 0793 Transmission Control Protocol. J. Postel. September 1981. (TCP)
- 0792 Internet Control Message Protocol. J. Postel. September 1981. (ICMP)
- 0791 Internet Protocol. J. Postel. September 1981. (IP)

The Internet Protocol is the one we're interested in for Assignment #2!

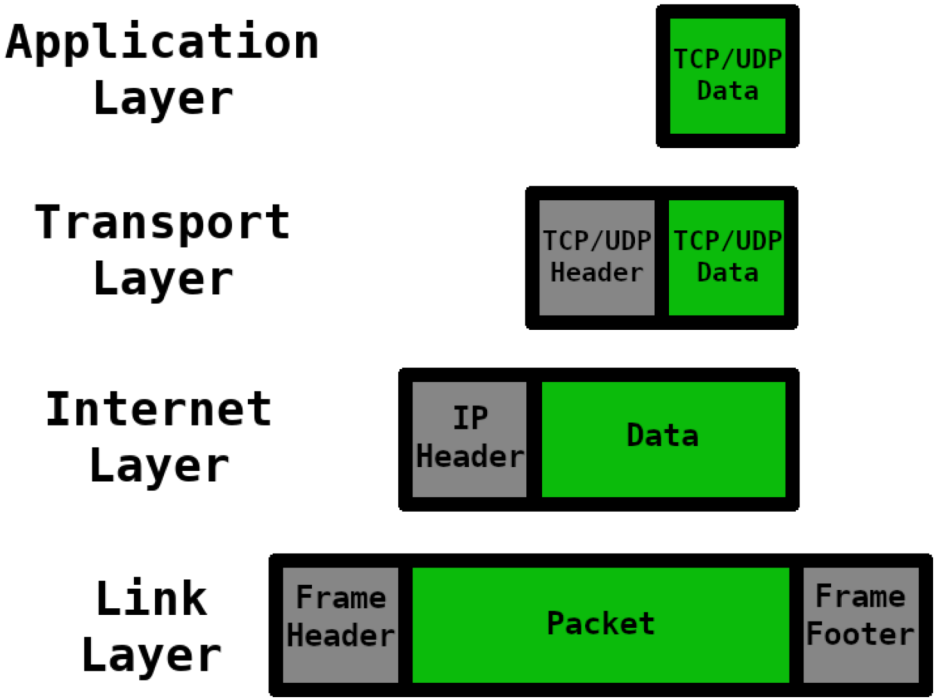


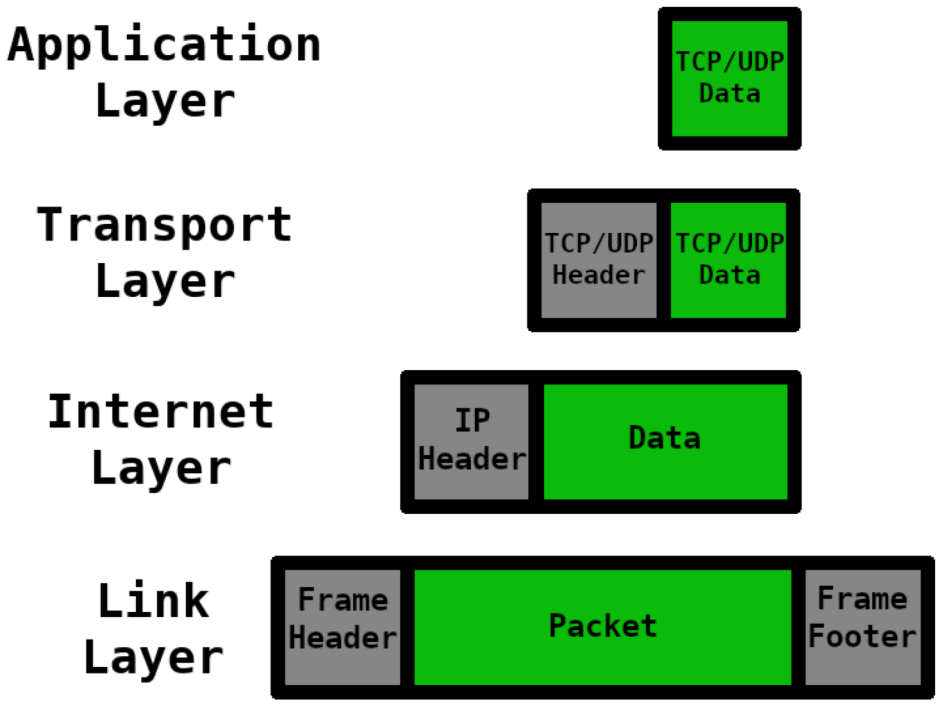
So with all this, how **does** it work?

Layers of Communication

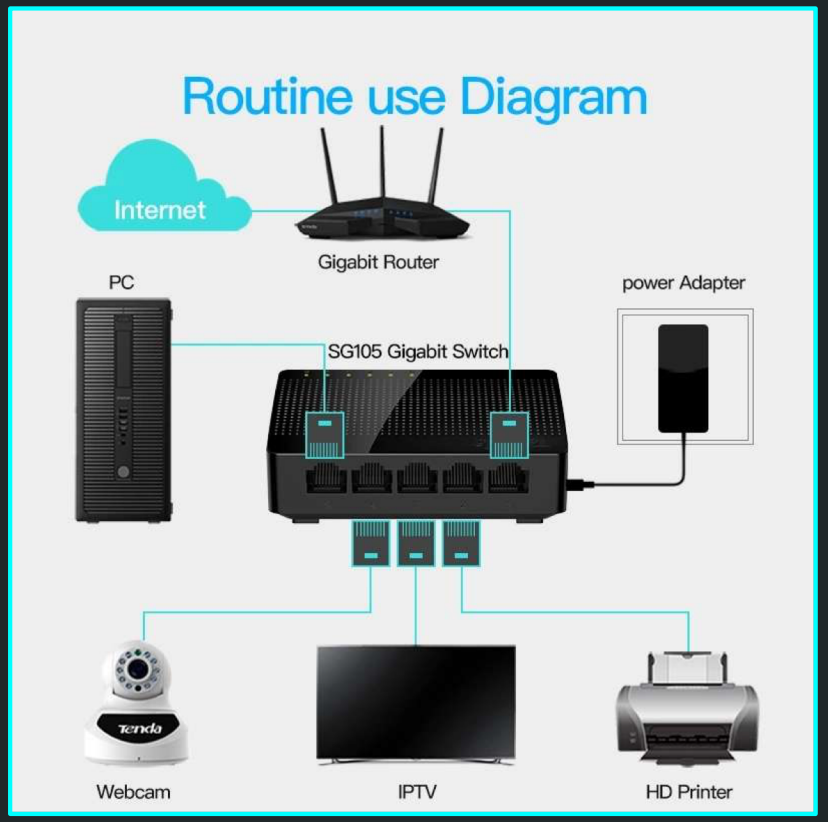


Layers of Communication

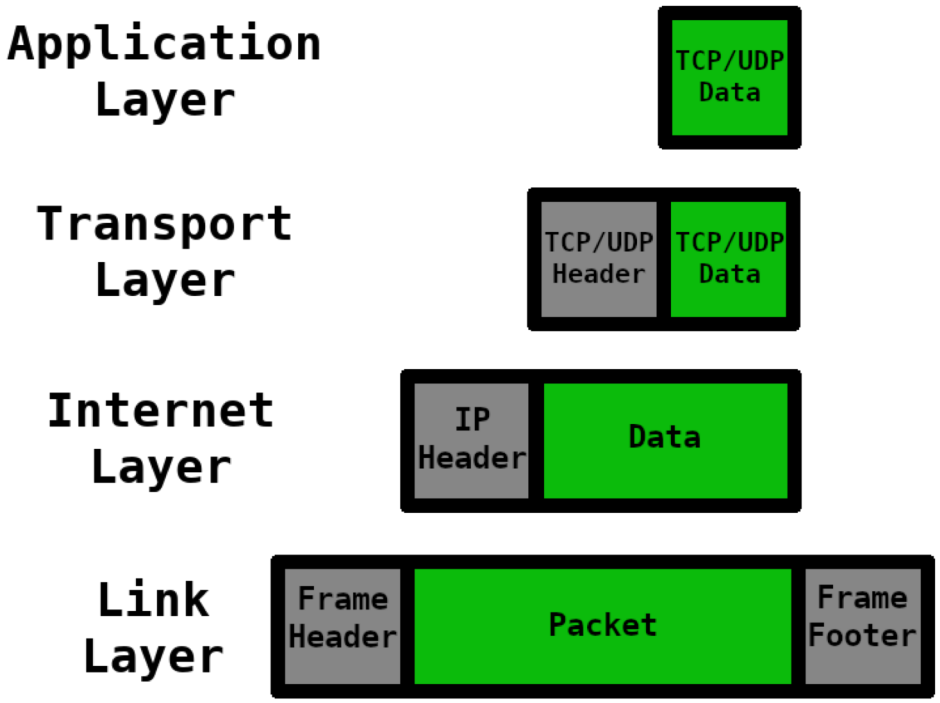




Which cable do I send my information to?

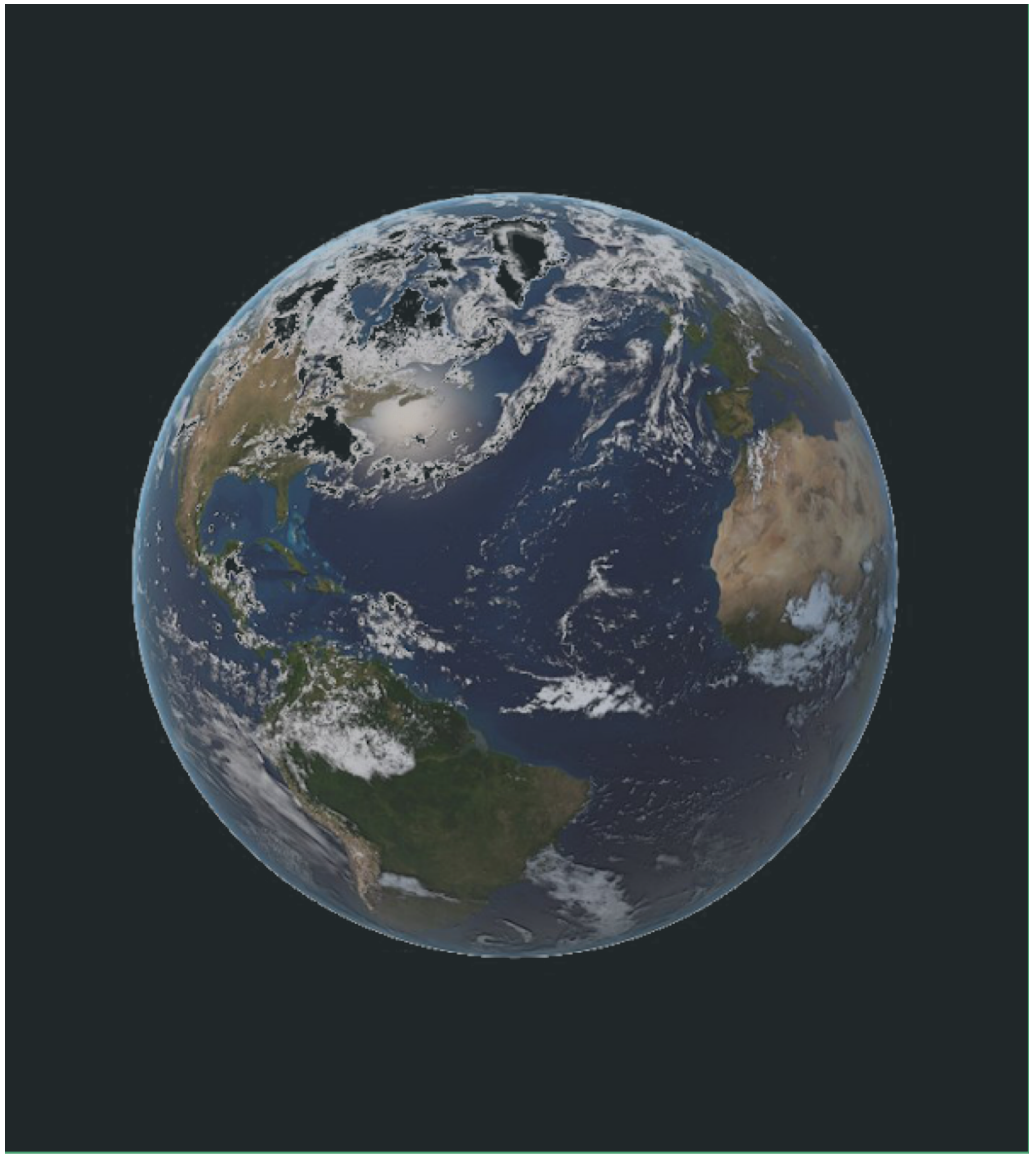

Routine use Diagram
Internet
Gigabit Router
PC
power Adapter
SG105 Gigabit Switch
Webcam
IPTV
HD Printer

Link Layer

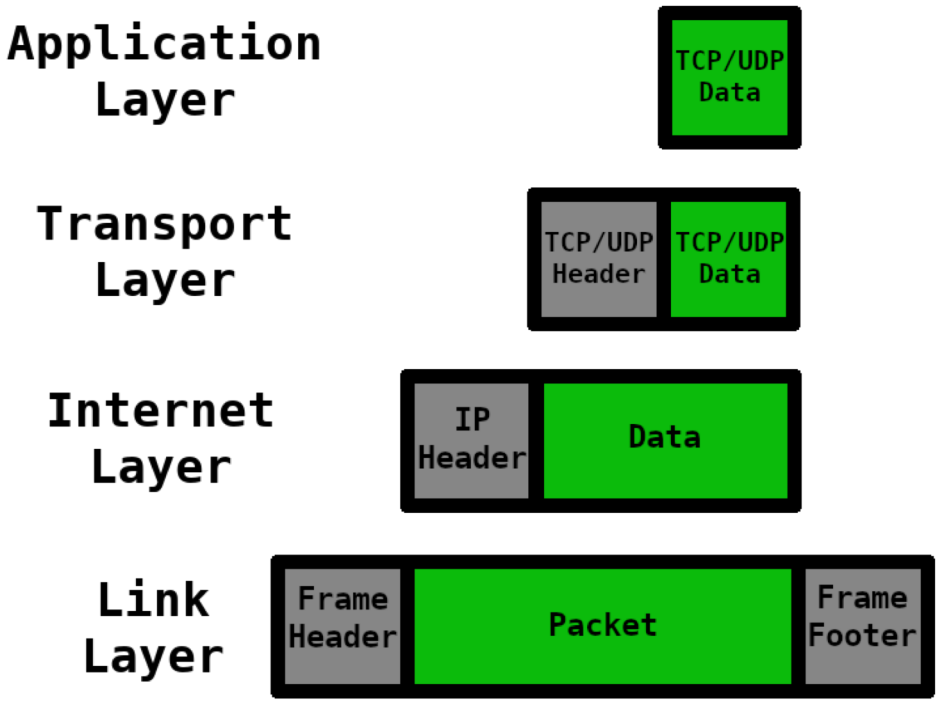


Which machine do I send my information to?

Which cable do I send my information to?



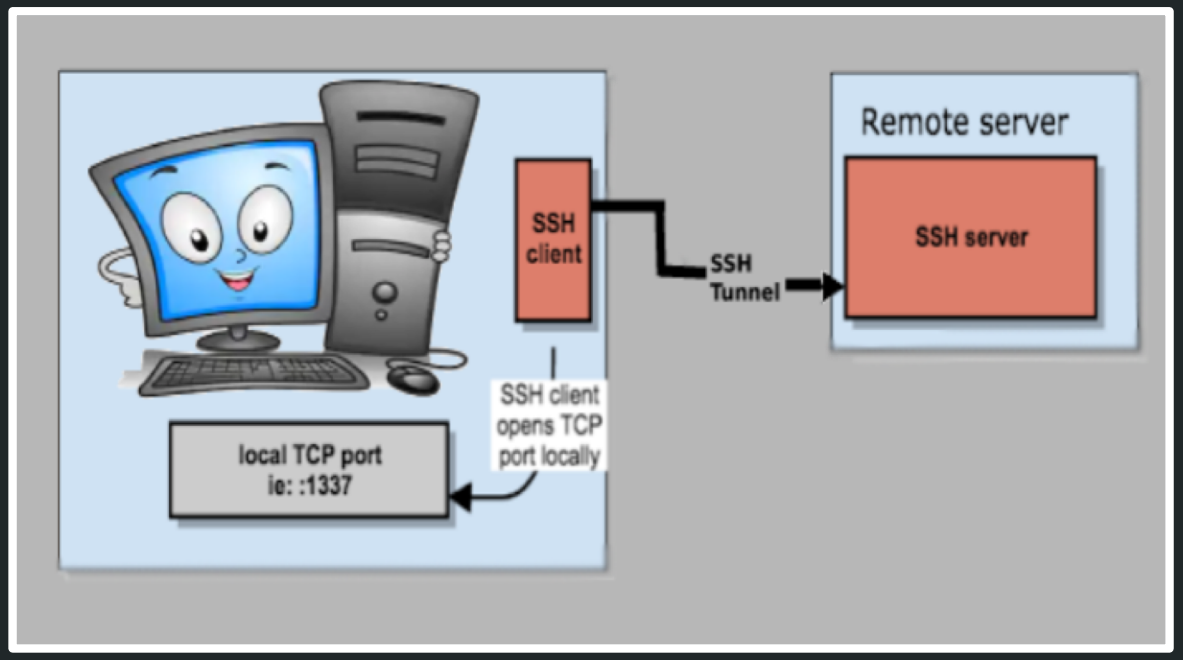
Internet Layer



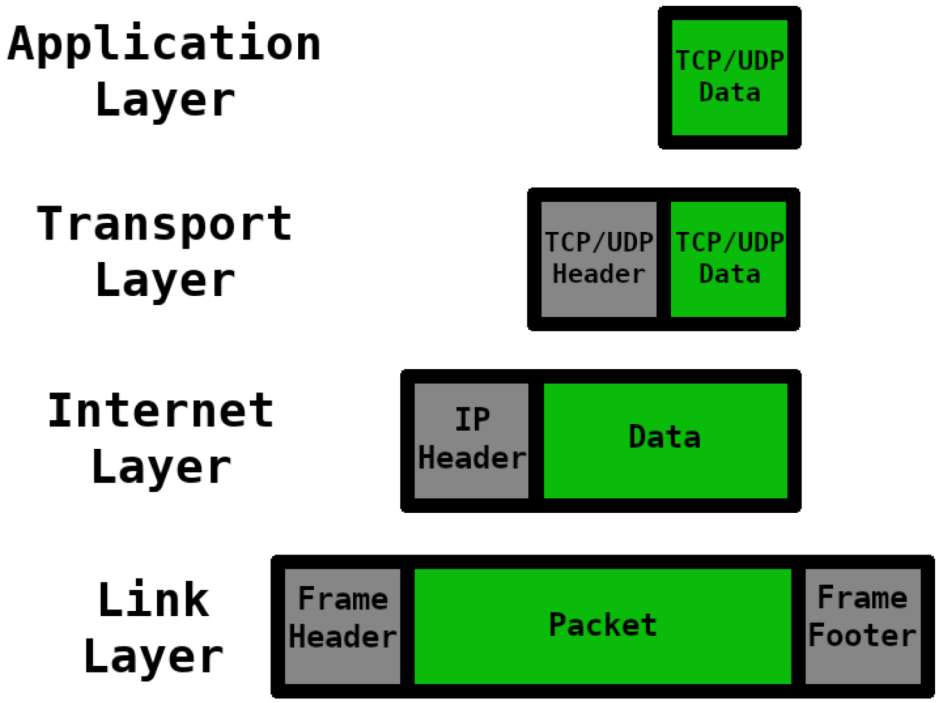
Which application do I send my information to?

Which machine do I send my information to?

Which cable do I send my information to?



Transport Layer

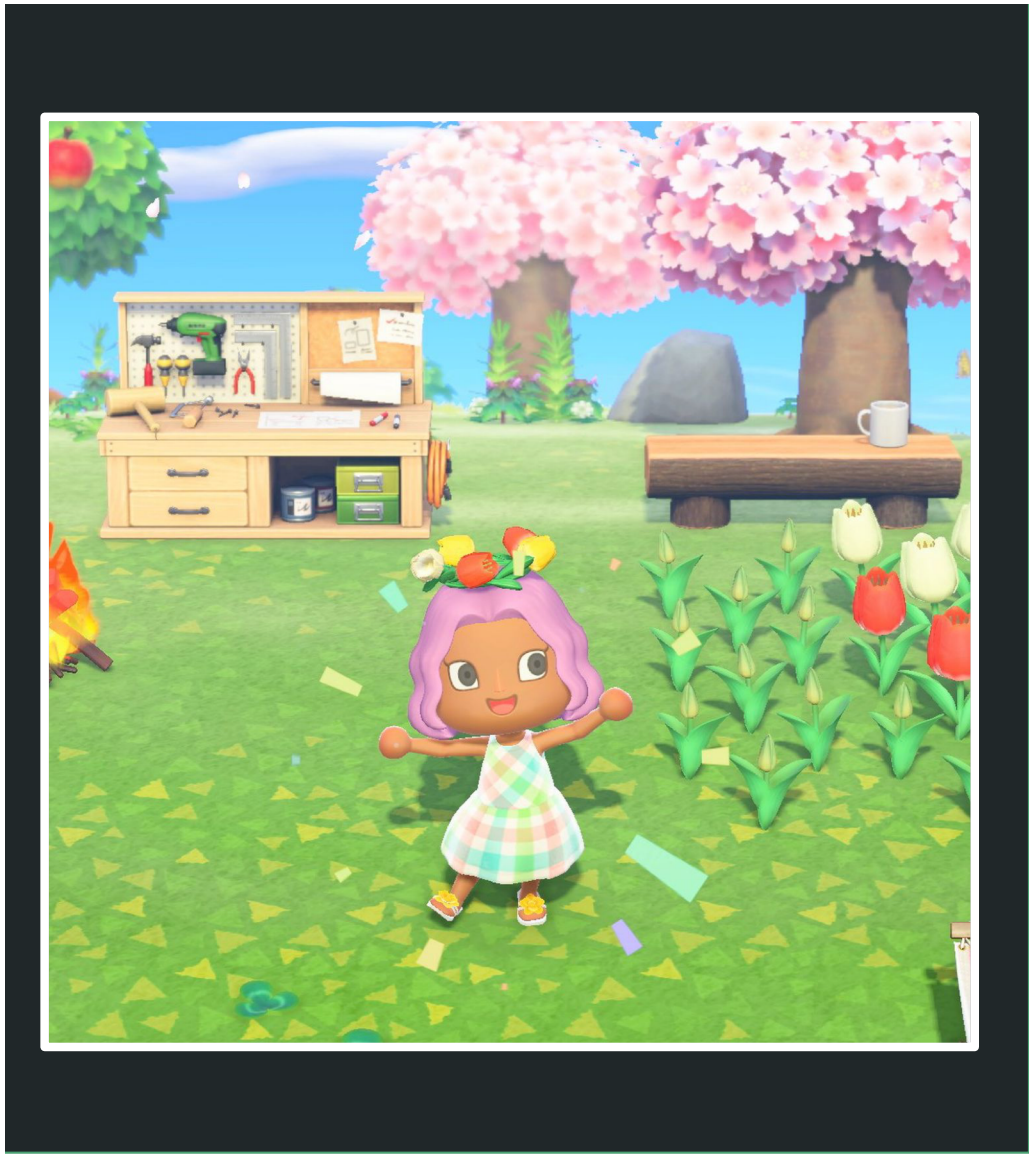


What am I sending?

Which application do I send my information to?

Which machine do I send my information to?

Which cable do I send my information to?



Application Layer

Internet Protocol

The Internet Layer (usually IPv4 or IPv6) is in charge of routing packets from one machine through the global internet to another. It has to work:

Internet Protocol

The Internet Layer (usually IPv4 or IPv6) is in charge of routing packets from one machine through the internet to another. It has to work:

- ➔ When components start failing, slowing down or are missing

Internet Protocol

The Internet Layer (usually IPv4 or IPv6) is in charge of routing packets from one machine through the internet to another. It has to work:

- ➔ When components start failing, slowing down or are missing
- ➔ When components have limited capacity

Internet Protocol

The Internet Layer (usually IPv4 or IPv6) is in charge of routing packets from one machine through the internet to another. It has to work:

- When components start failing, slowing down or are missing
- When components have limited capacity
- When components are malicious

Internet Protocol

The Internet Layer (usually IPv4 or IPv6) is in charge of routing packets from one machine through the internet to another. It has to work:

- ➔ When components start failing, slowing down or are missing
- ➔ When components have limited capacity
- ➔ When components are malicious
- ➔ When half the internet is gone ...

Internet Protocol

The Internet Layer (usually IPv4 or IPv6) is in charge of routing packets from one machine through the internet to another. It has to work:

- ➔ When components start failing, slowing down or are missing
- ➔ When components have limited capacity
- ➔ When components are malicious
- ➔ When half the internet is gone ...

It has to work.

IP Fragmentation

- ➔ Fragmentation lets us send packets between machines while allowing
- ➔ machines on the internet with different capacities
- ➔ smaller machines to route large packets, just in more than one step
- ➔ routing sub-packets in parallel along different paths to the same target

IP Fragmentation

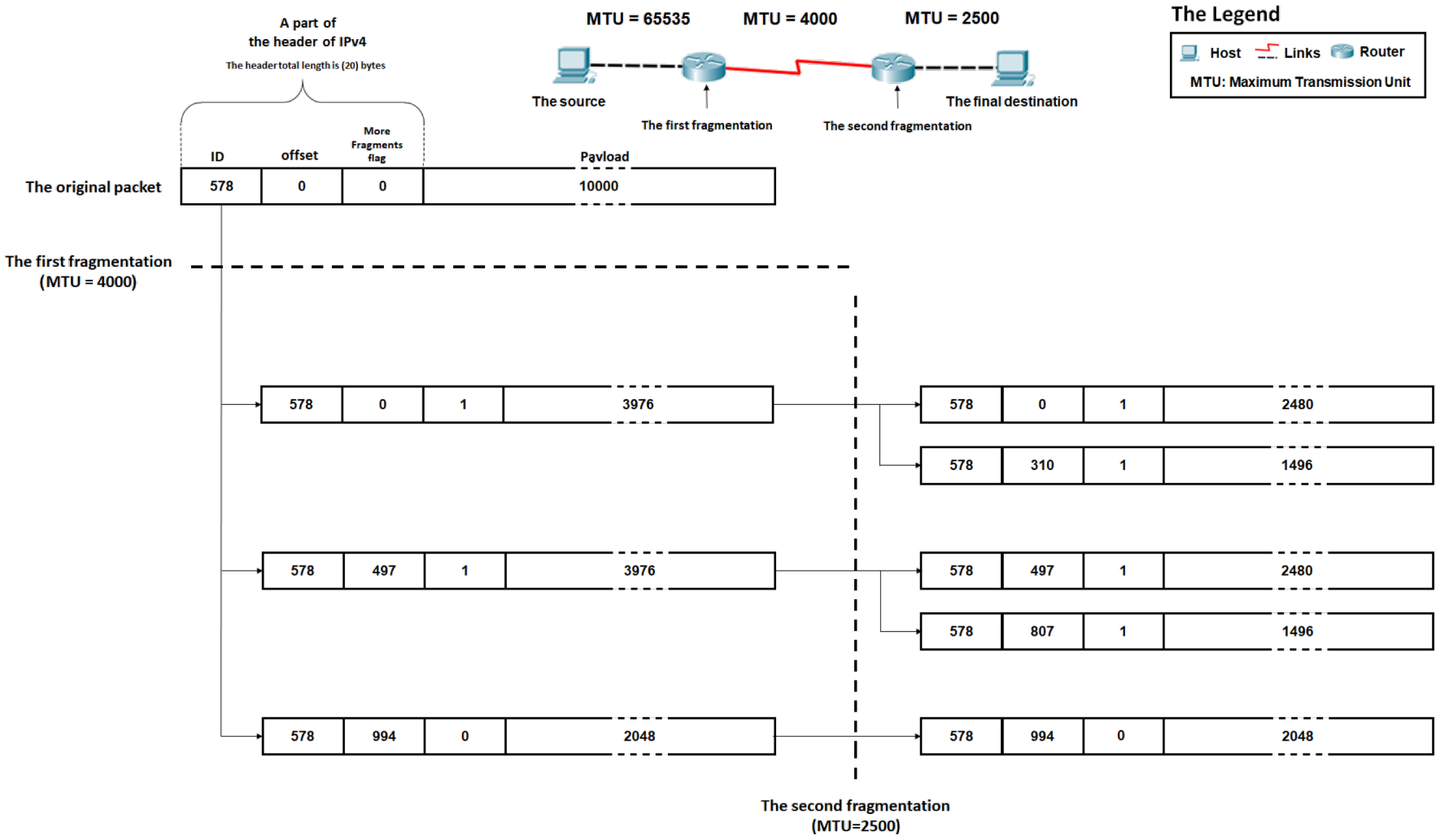
BUT, we need to define minimal requirements to ensure working communication!

From RFC 791 (Internet Protocol):

> Every internet module must be able to forward a datagram of 68 octets without further fragmentation. This is because an internet header may be up to 60 octets, and the minimum fragment is 8 octets.
>
> Every internet destination must be able to receive a datagram of 576 octets either in one piece or in fragments to be reassembled.

IP Fragmentation

How does it work? (I apologize to your eyes.)


A part of
the header of IPv4
The header total length is (20) bytes
MTU = 65535
MTU = 4000
MTU = 2500
The source
The first fragmentation
The second fragmentation
The final destination
The Legend
Host
Links
Router
MTU: Maximum Transmission Unit
ID
offset
More Fragments flag
Payload
The original packet
578
0
0
10000
The first fragmentation
(MTU = 4000)
578
0
1
3976
578
0
1
2480
578
310
1
1496
578
497
1
3976
578
497
1
2480
578
807
1
1496
578
994
0
2048
578
994
0
2048
The second fragmentation
(MTU=2500)

IP Fragmentation

So how can we create these packets?

Python → struct.pack

C/C++ → structures (must be packed)

Java → uhhh, please don't (but if you really need to, probably ByteBuffer)

IP Fragmentation

DEMO!