Based on “Computer Networks: A Systems Approach”, 5e

By Larry L. Peterson and Bruce S. Davie, Morgan-Kaufmann / Elsevier

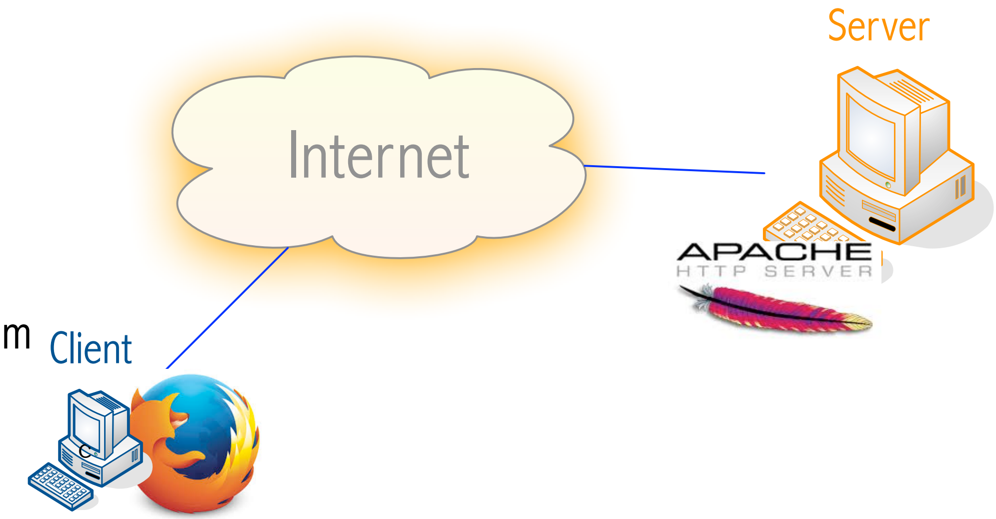
Extended and adapted by José María Foces Morán (2015)

CHAPTER 1: FOUNDATION

An introduction to Computer Networks

Essential example: www

- www = World Wide Web:
  - A distributed and Client/Server application
    - Server program (e.g., Apache)
    - Client program (e.g., Firefox)
- URL
  - Uniform Resource Locator
  - http://paloalto.unileon.es/cn/index.html
- Web pages are downloaded by the client from the server
  - Client and server speak the http protocol
  - http = Hyper Text Transfer Protocol
- HTTP, in turn uses the TCP protocol
  - TCP = Transmission Control Protocol
  - TCP provides reliability



Main metrics used in Networking

- Bandwidth
  - Directly related to the acceptable speed of bit transmission over some medium
  - Number of bits transmitted in one second:
    - Bps (Bits Per Second = Bits/Sec)
  - Since bandwidth is a rate, the multipliers take on the following values:
    - K (Kilo = 10^3)
    - M (Mega = 10^6)
    - G (Giga = 10^9)
    - T (Tera = 10^{12})
- Delay
  - Seconds
  - How much time it takes to transport one bit from a source to a destination directly connected
  - Propagation delay
- Jitter
  - The variance of the delay

Connectivity

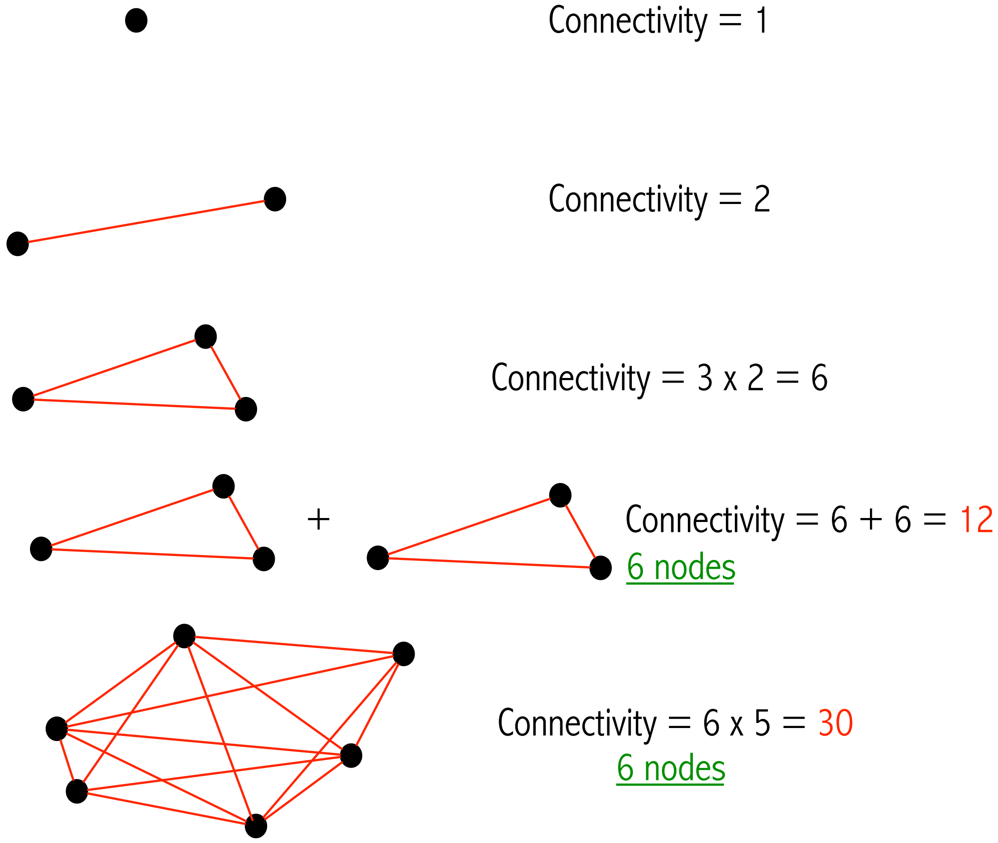
Computer Networks connect computers

Theoretical connectivity

□ Connectivity is the capacity of connection of a network

□ If a network has N hosts, its connectivity is: $N \cdot (N-1) \cong N^2$

□ Metcalf’s law: The connectivity of a network grows fast as we add more nodes (N^2)



Connectivity = 1

Connectivity = 2

Connectivity = 3 x 2 = 6

+

Connectivity = 6 + 6 = 12

6 nodes

Connectivity = 6 x 5 = 30

6 nodes

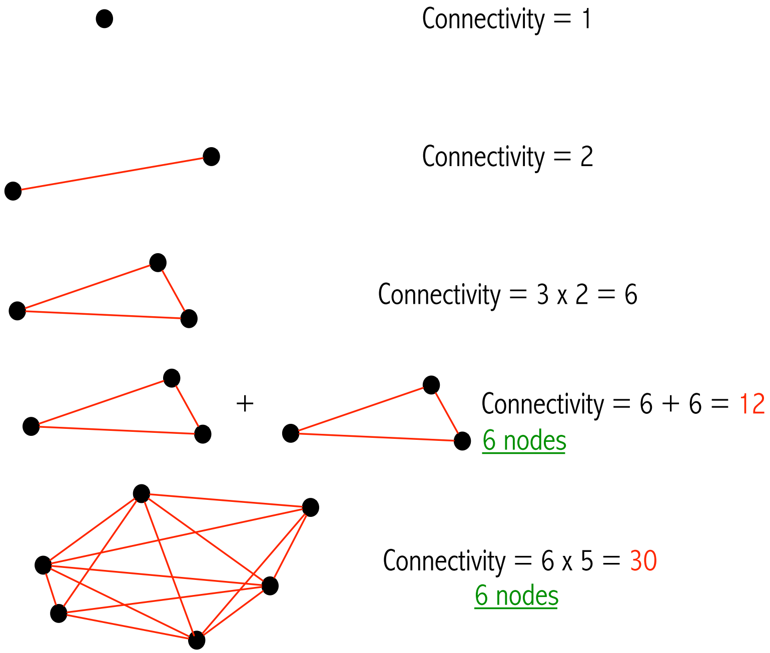
Therоretical connectivity is bounded by network *technology*

□ Metcalf's law: Represents the potential connectivity of a network

□ Nodes communicate by sending/receiving messages through the links which bandwidth is limited

□ Each node must 'have some knowledge' about the topology of the network

□ These factors limit the use of the potential connectivity



Connectivity = 1

Connectivity = 2

Connectivity = 3 x 2 = 6

Connectivity = 6 + 6 = 12
6 nodes

Connectivity = 6 x 5 = 30
6 nodes

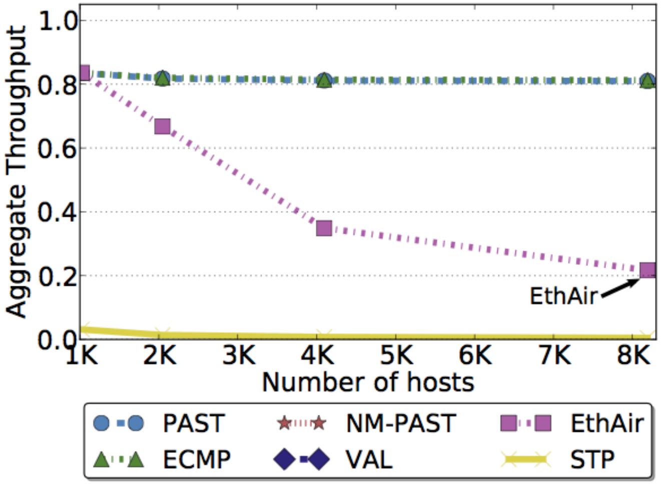
Scalable connectivity

- Not all network technologies use the available connectivity with the same efficiency

- Ethernet can function efficiently up to certain network size: we say that Ethernet scales well up to that limit.

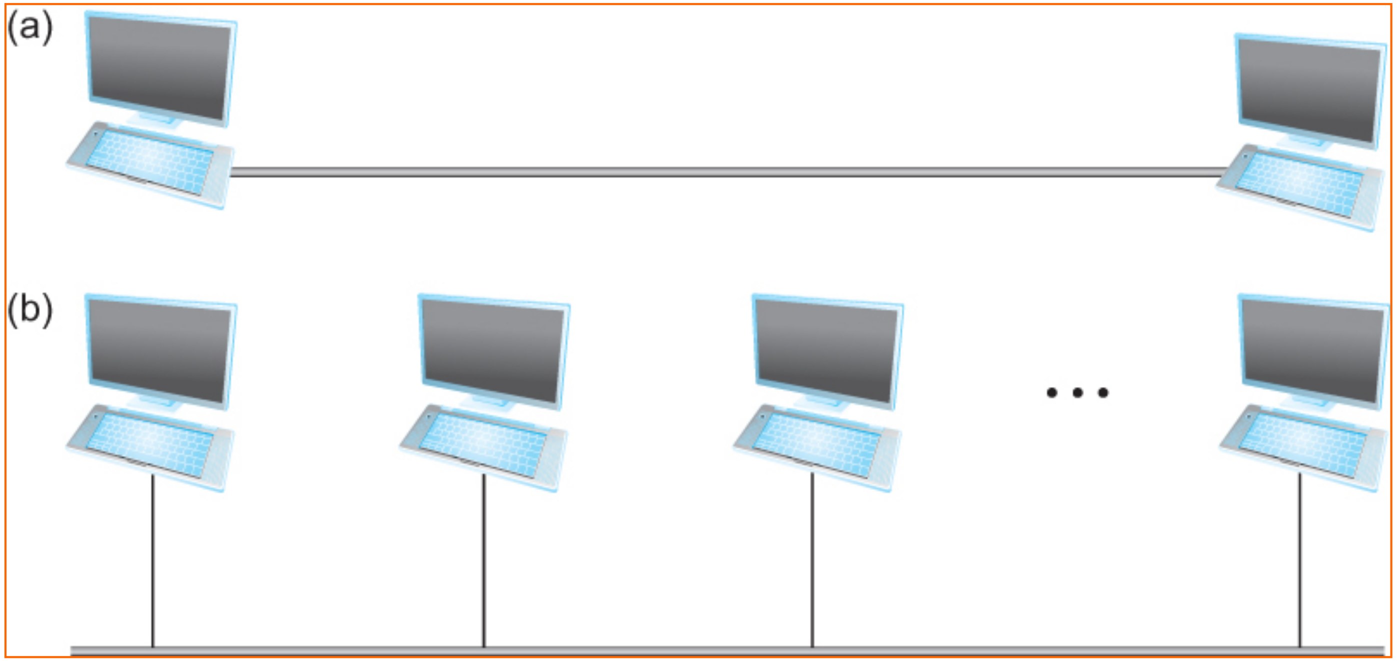
- Then, how come the Internet has 2000M hosts? How can the Intenet scale to such a huge size so well?
  - IP protocol

EthAir scales poorly



Attaining scalable connectivity

- Something is __scalable__ if it can grow to huge sizes preserving its functionality
- We want that the connectivity offered by a network be scalable, i.e., that it can grow as needed

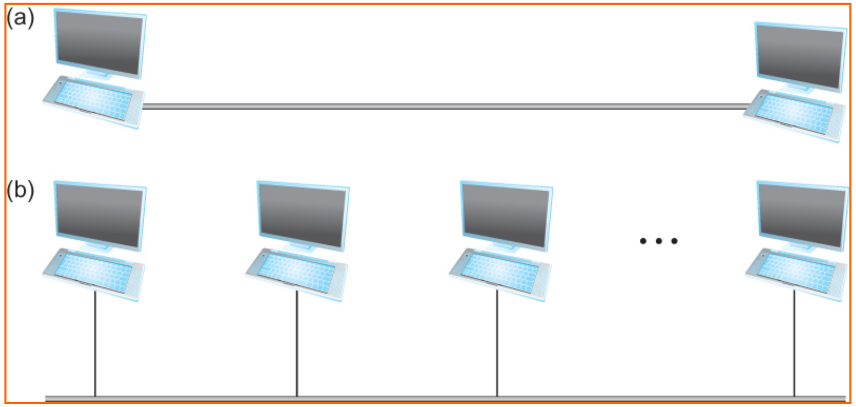


Basic Network terms (i)

- Scale (Size)
- Host

- Link

- Communication links:
  - (a) Point-to-point
  - (b) Multiple access



Which one scales better?

Basic terms (ii)

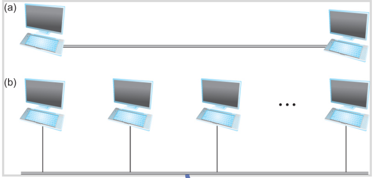
□ Switch

□ Network

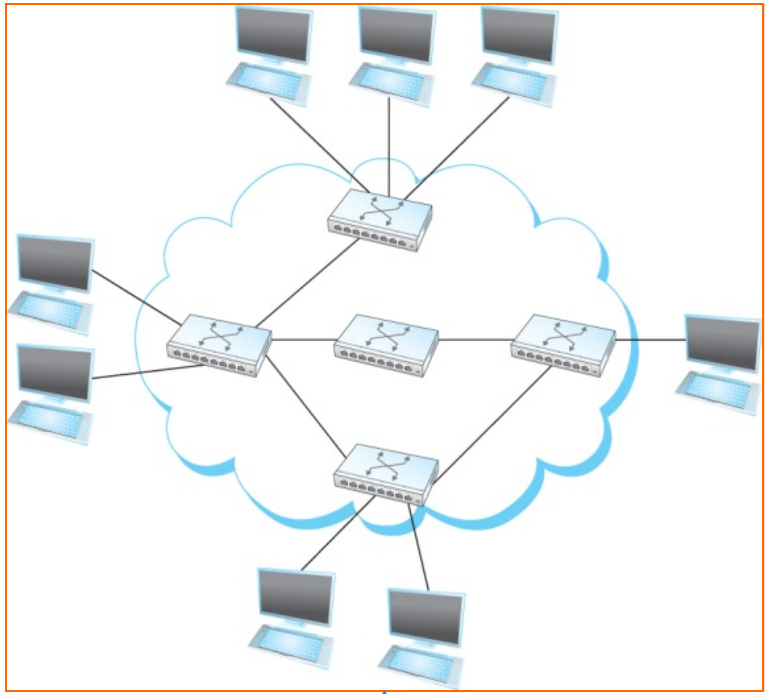
□ Switched Network
- ◘ Circuit Switched (Old telephone network)
- ◘ Packet Switched

□ Packet and message

□ Store-and-forward



Switched network (A single network !!!)



- Improved scalability

Basic terms (iii)

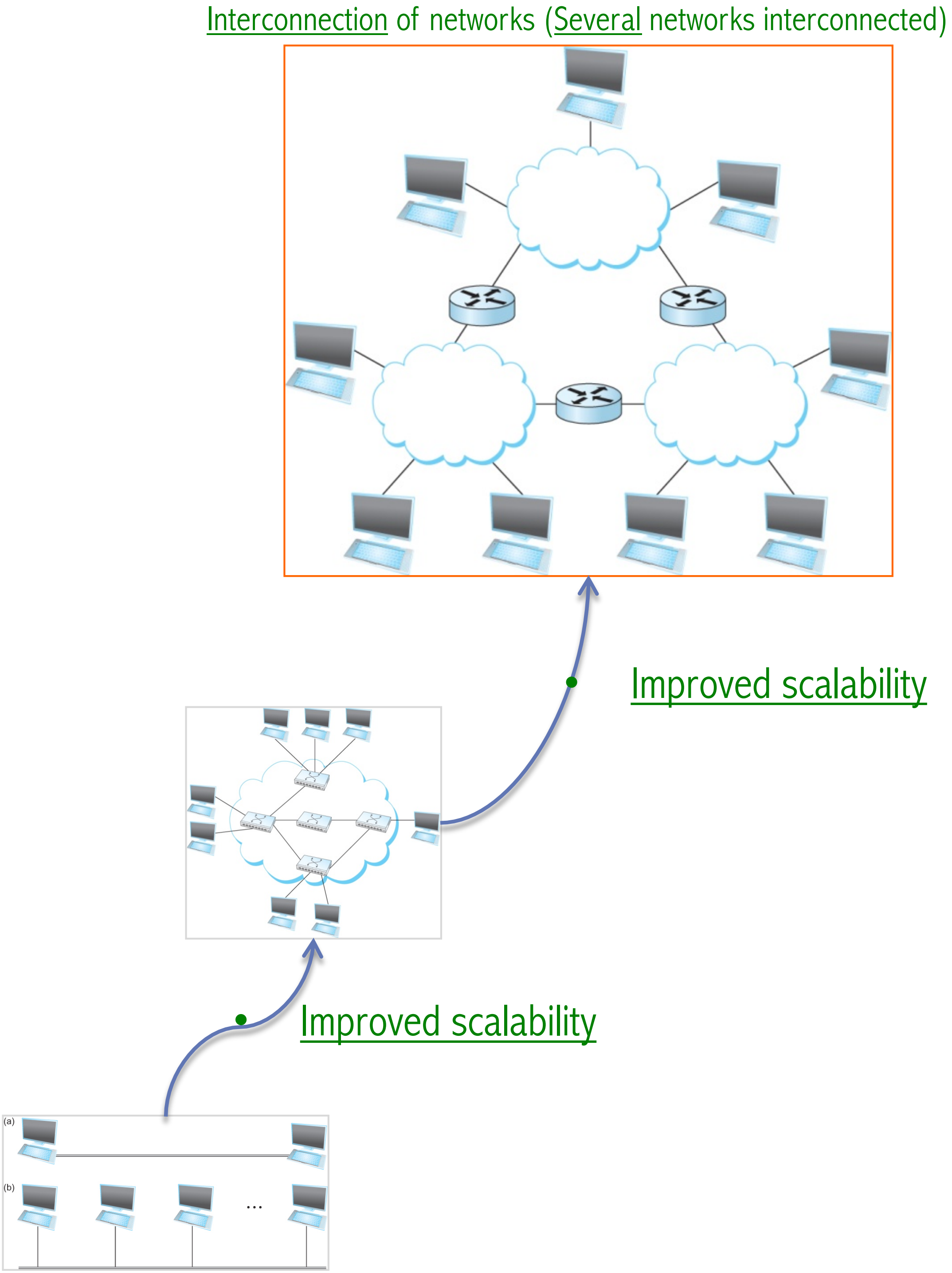
□ Router/gateway

□ Host-to-host connectivity

□ Address

□ Routing

□ Unicast/broadcast/multicast



Interconnection of networks (Several networks interconnected)

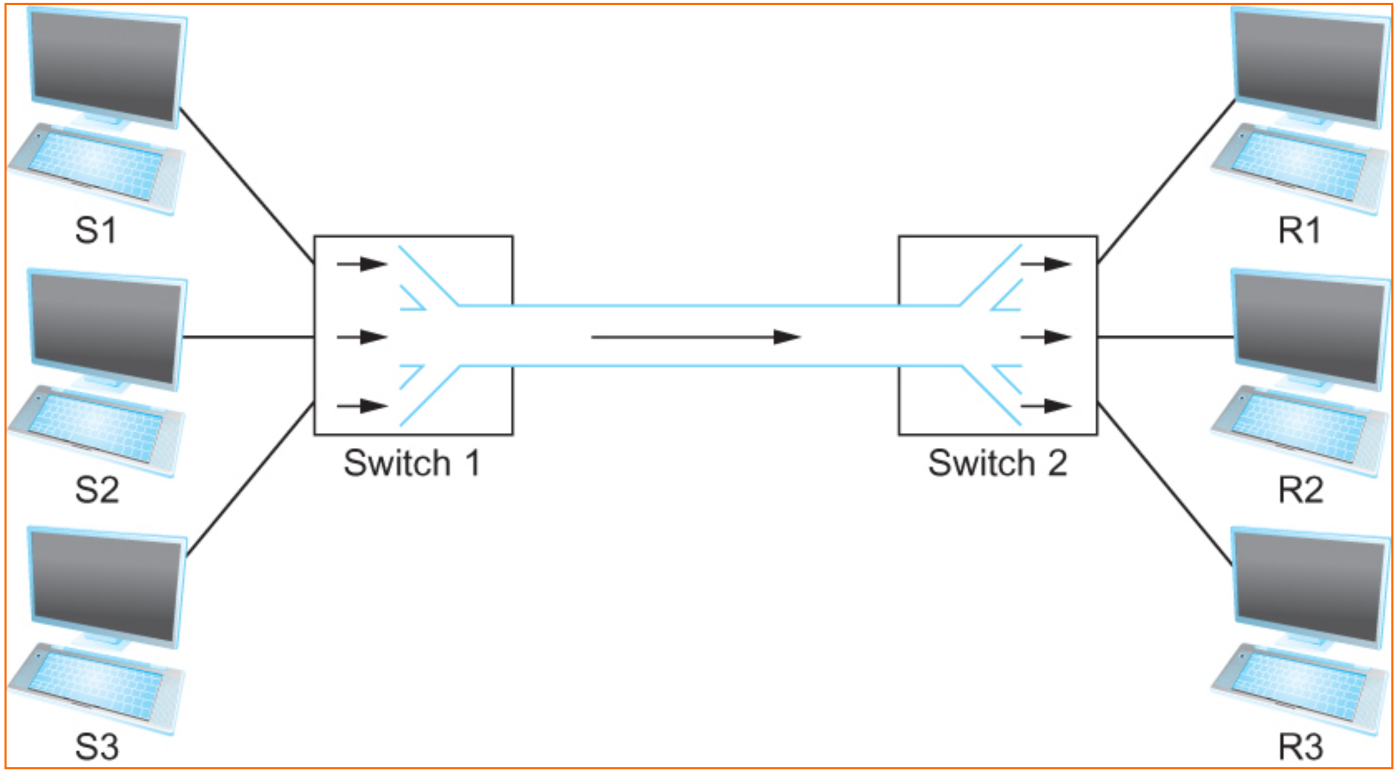
Improved scalability

Improved scalability

Sharing connectivity

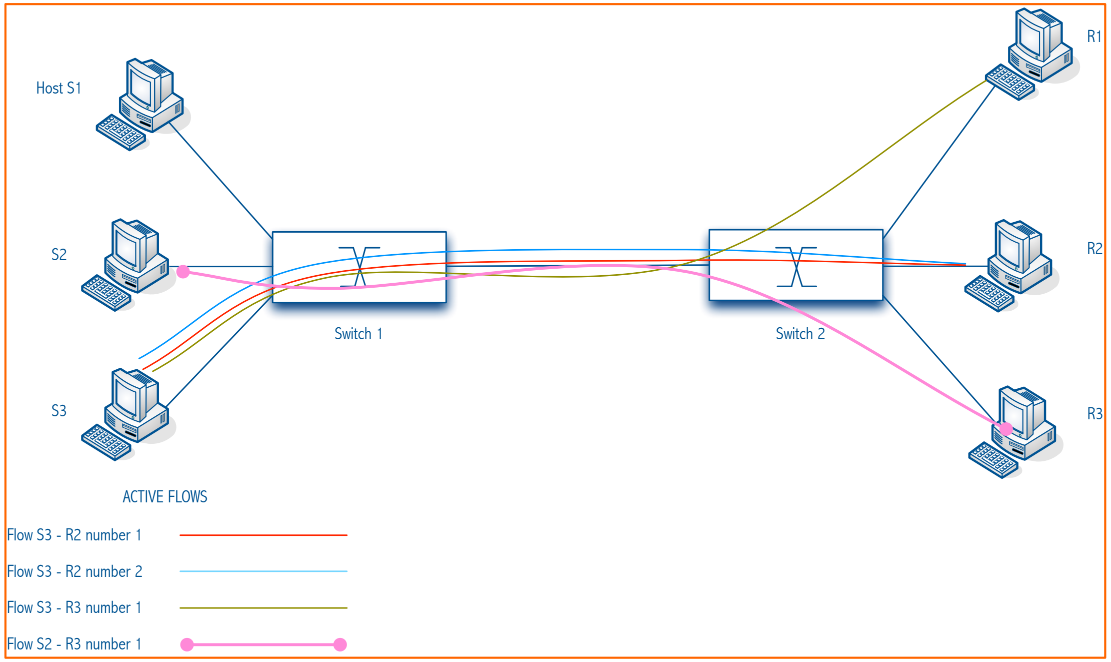
How to share a link, a network among multiple users

Cost-Effective Resource Sharing



Multiplexing multiple **logical flows** over a single physical link

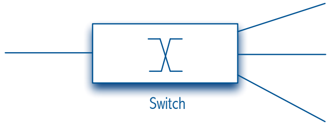
Review of terminology



Multiplexing multiple logical flows over a single physical link

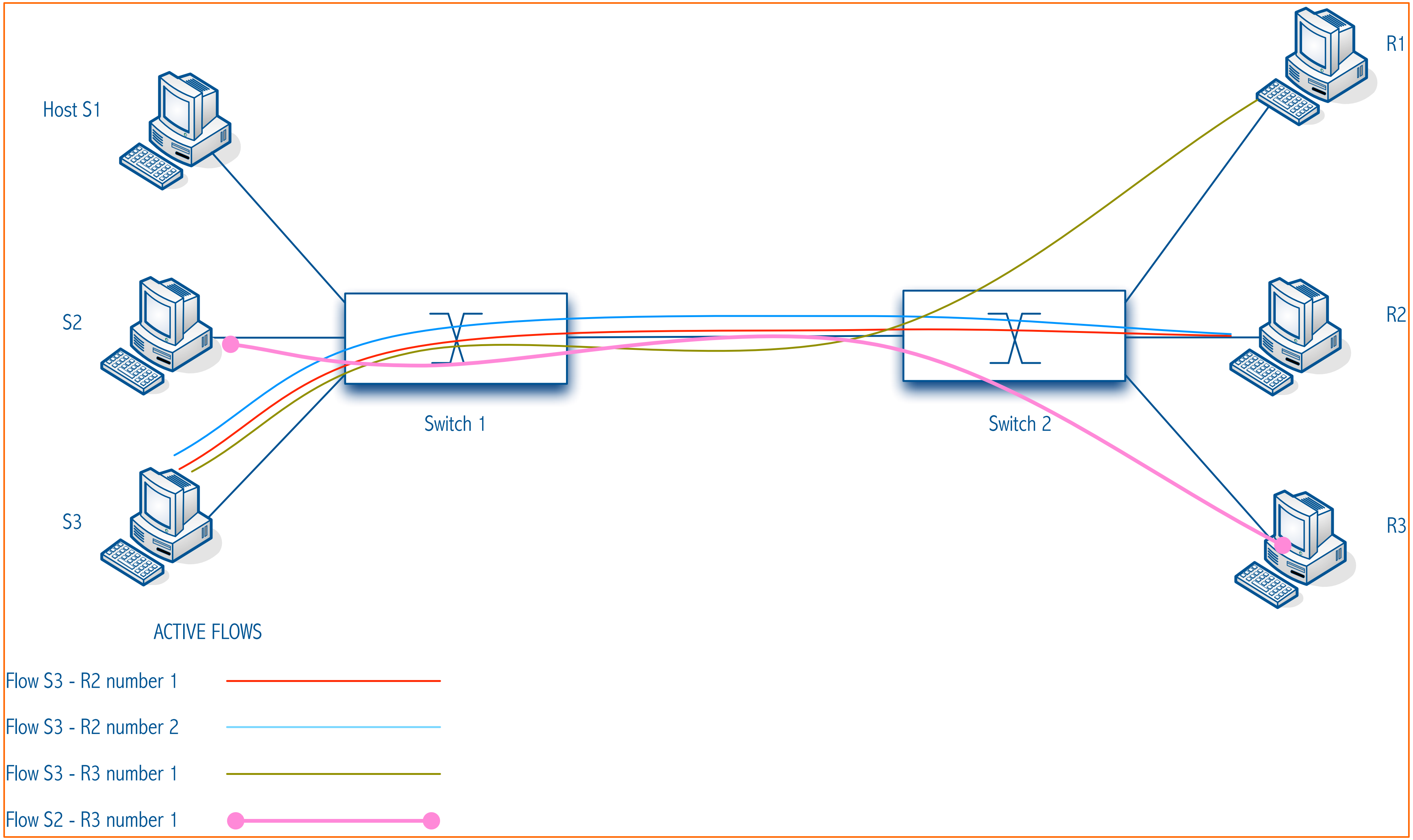
- **Hosts**: the computers that run the application programs
  - Clients
  - Servers

- **Network nodes**
  - DataComm equipment (DCE) for building a network

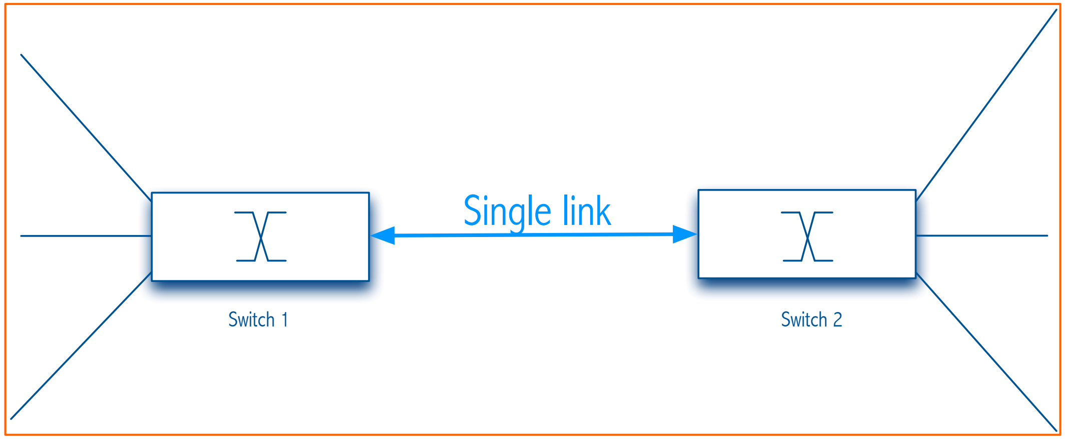


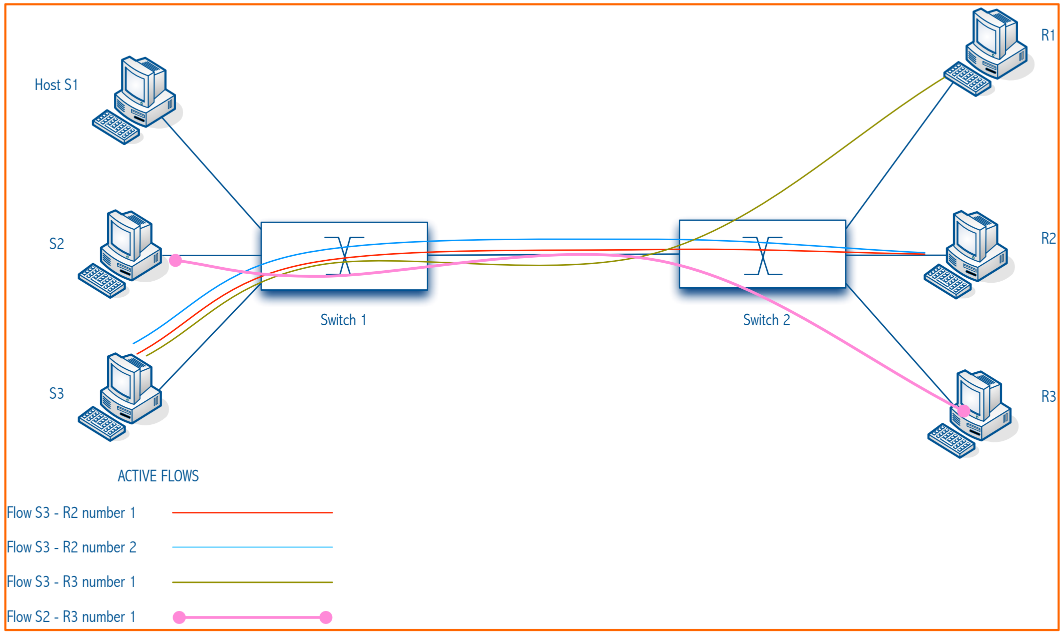
- **Links:**
  - The physical transmission media and other logical characteristics
    - Simplex
    - Half-duplex
    - Full-duplex
  - *WiFi, Twisted pair cables, optical fibers, etc*
- **FLOW** comes next

What is a *logical* flow?



Multiplexing = sharing



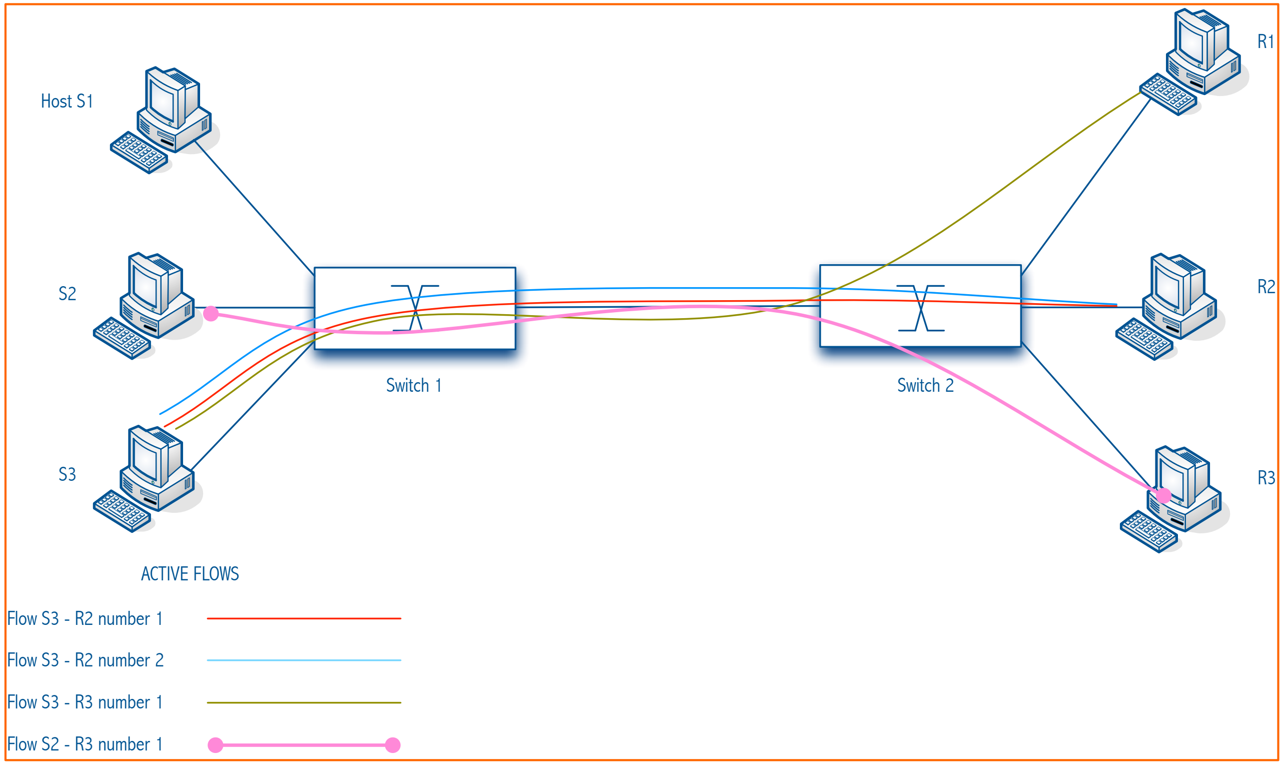


- Switch1 and Switch2 are linked by a single link
- How can that link be shared among the hosts?
- How can that sharing be done?
  - By multiplexing the flows of packets over the link that connects the switches

Multiplexing techniques

1. Synchronous Time-division Multiplexing (STDM)

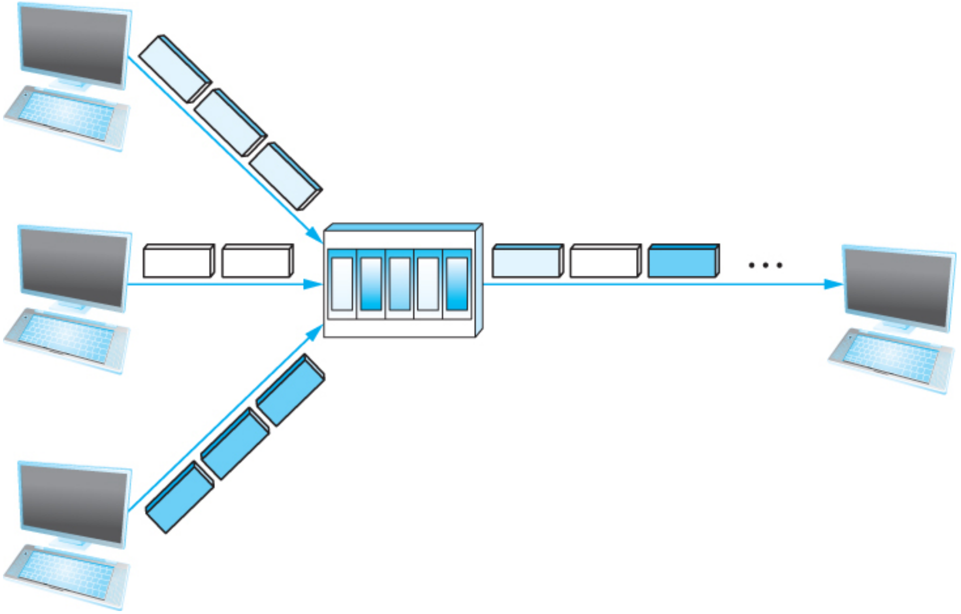
- Allocate a predetermined time slot to each flow
- Drawbacks
  - If a flow is not transmitting (idle), its slot remains allocated and is wasted



Multiplexing techniques

2. FDM: Frequency Division Multiplexing

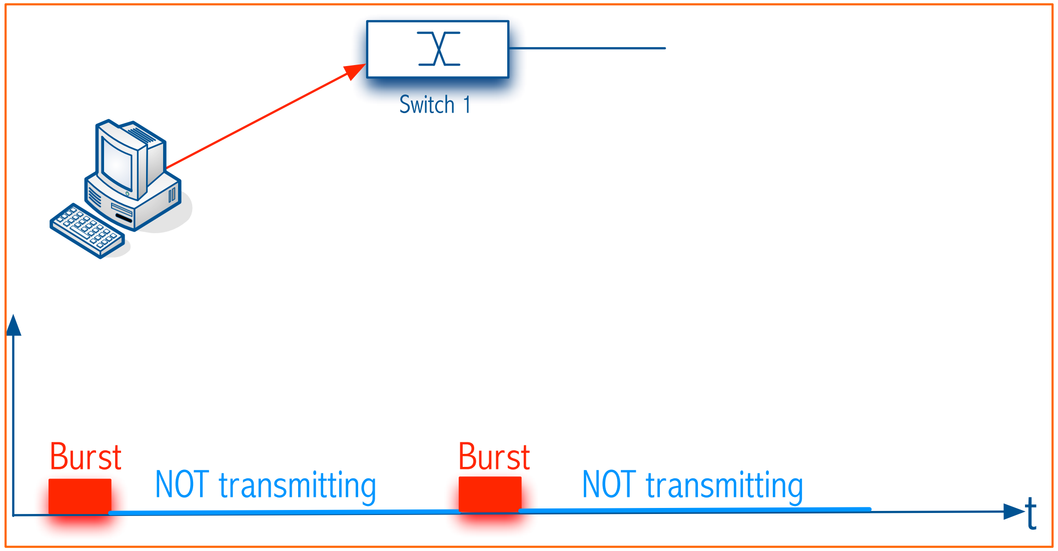
- *Similar* to FM radio
- Each flow is assigned a portion (a band) of the link spectrum, each portion is the same size (bandwidth)

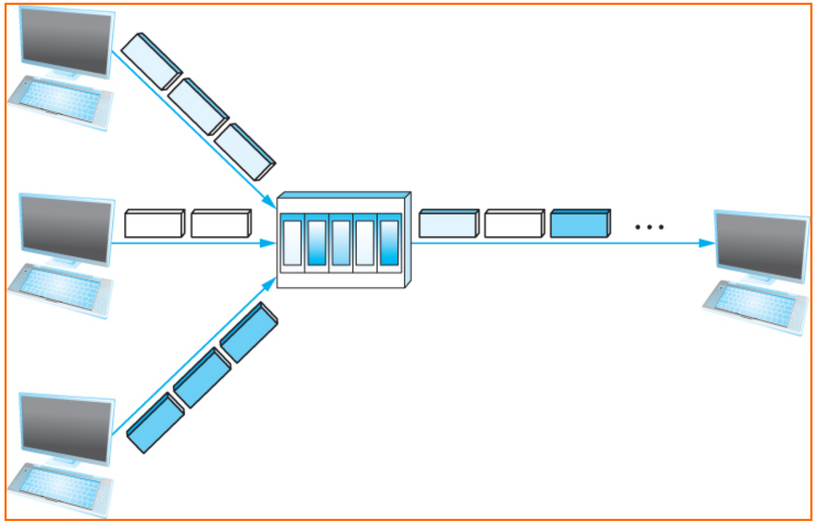


Multiplexing techniques

3. Statistical Multiplexing

- Data is transmitted based on the level of demand (offered load) of each flow
  - Sources are supposed to be bursty
  - The bandwidth of the shared link is a fraction of the peak aggregated bandwidth of the other links



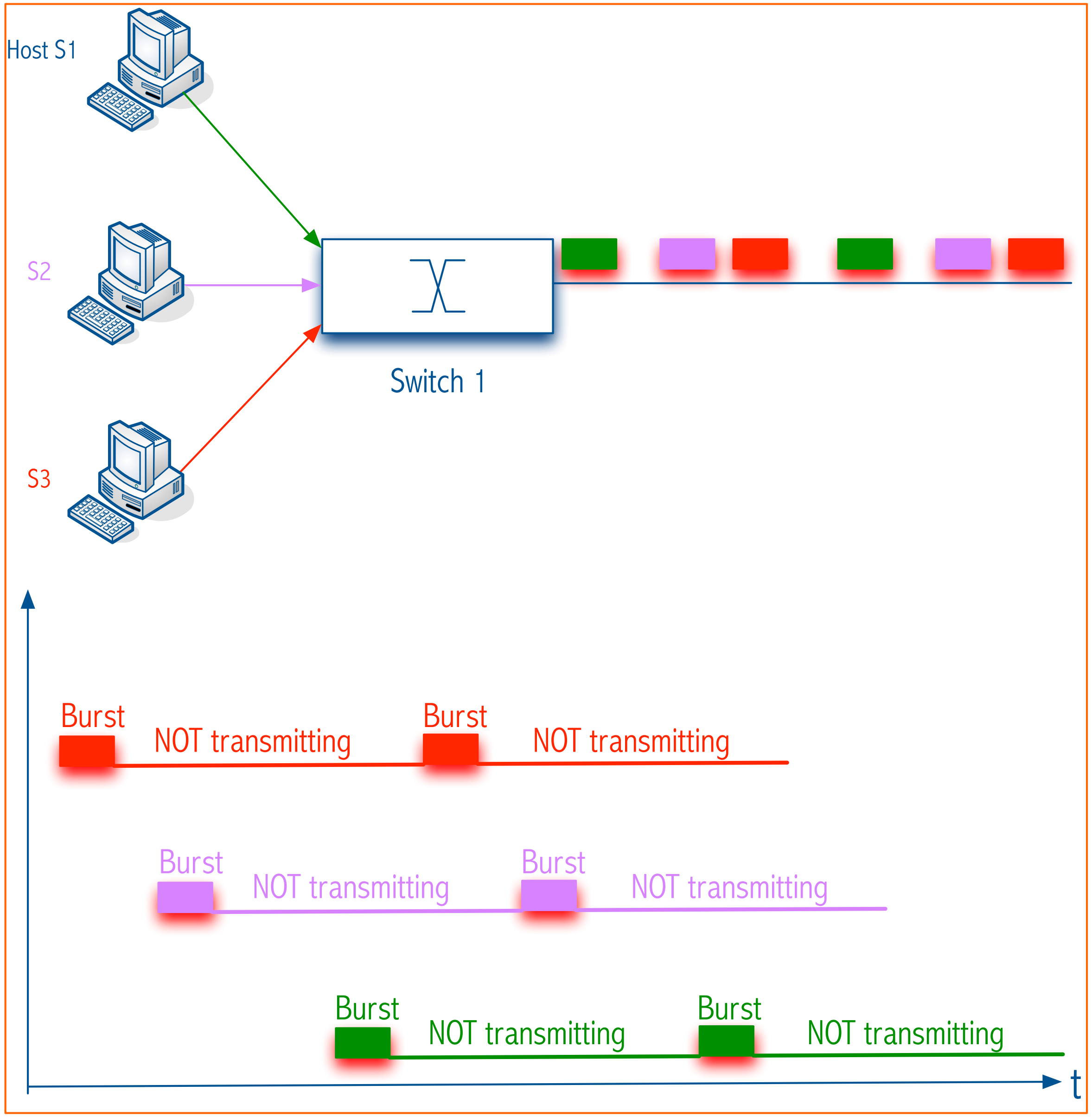


A bursty source

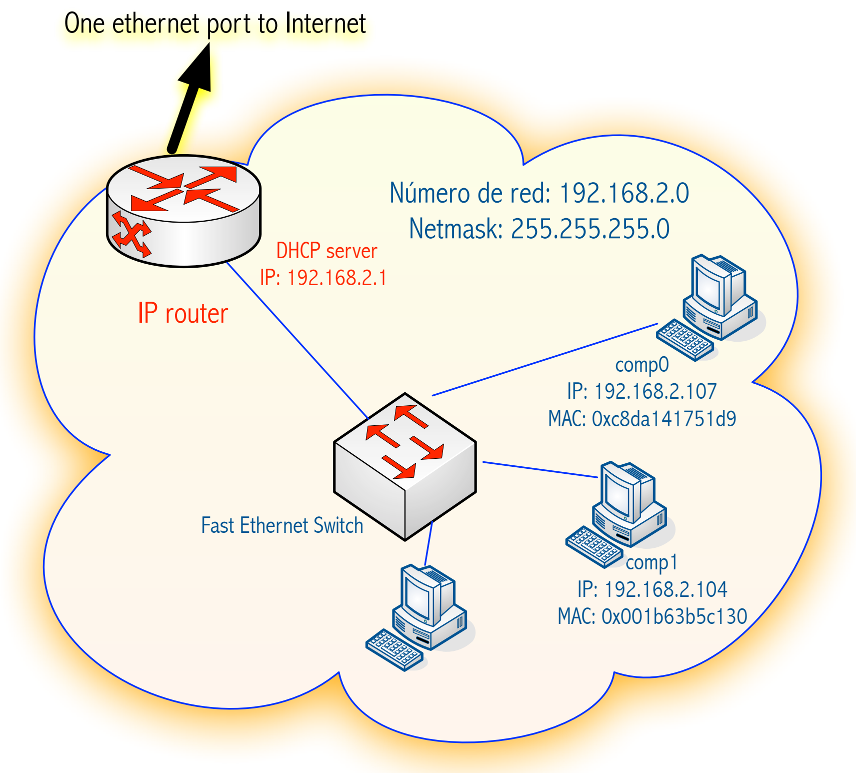
Multiplexing techniques

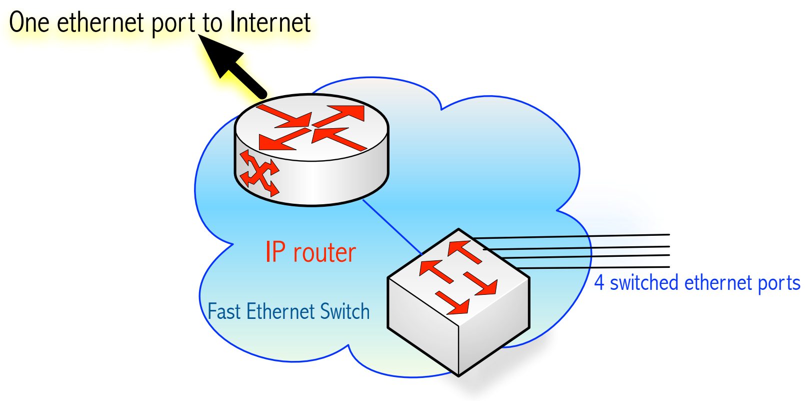
3. Statistical Multiplexing

- Data is transmitted based on the level of demand (offered load) of each flow
  - Sources are supposed to be bursty
  - The bandwidth of the shared link is a fraction of the peak aggregated bandwidth of the other links
- Packets vs. Messages
- Some switches may apply QoS (Quality of Service):
  - FIFO, Round-Robin, Priorities
- Congestion



Traditional types of networks





Depending on their geographical area

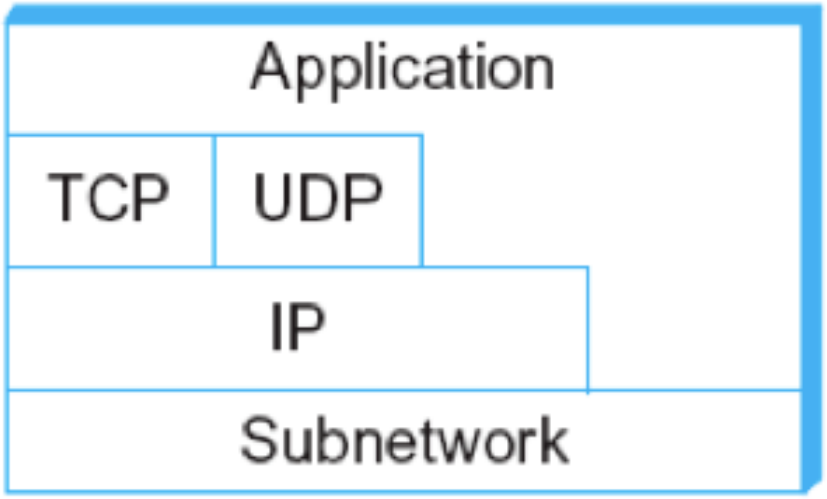
- LAN : Local Area Network
  - Ethernet
- MAN: Metropolitan
  - WiMax
- WAN: Wide
  - X.25
- SAN (System Area Networks)
  - Interconnect hard disks, network storage
  - High speed

Network Architecture

How to hide the complexity of networks in order to promote the development of new network uses

Architecture

TCP/IP architecture!



Application

TCP

UDP

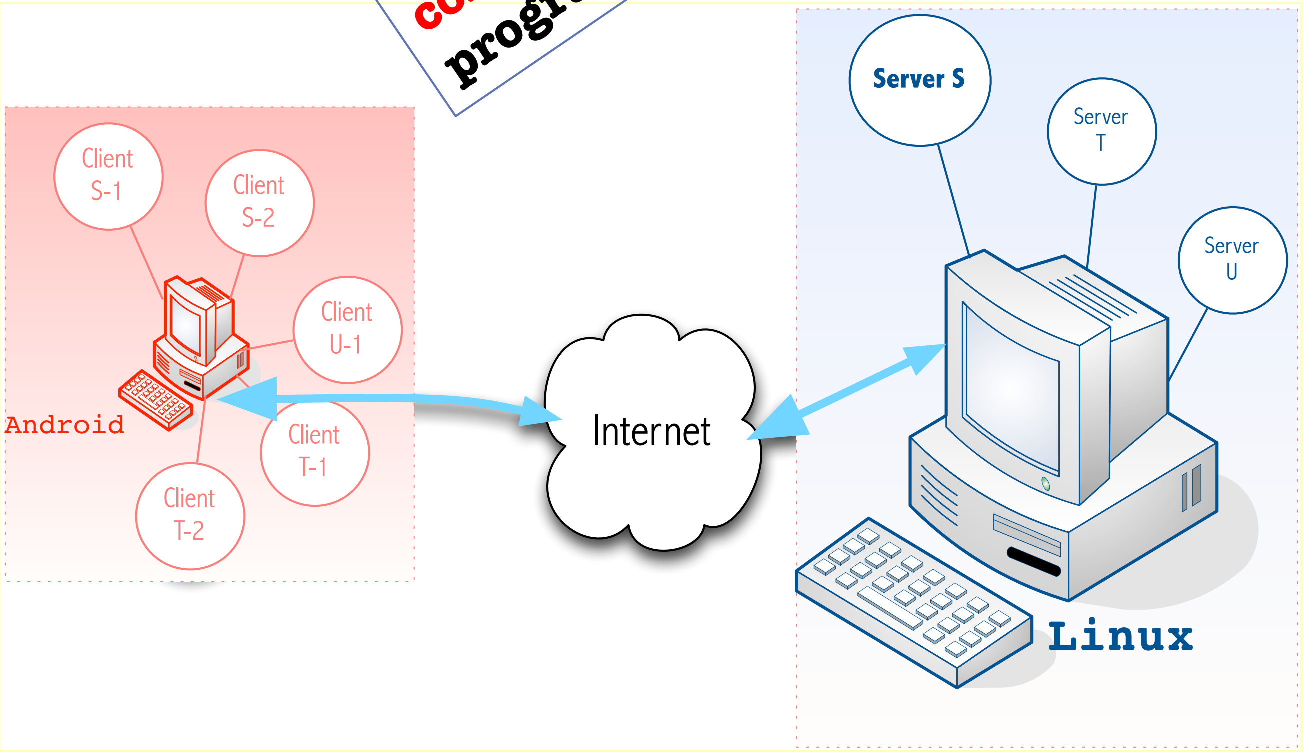
IP

Subnetwork

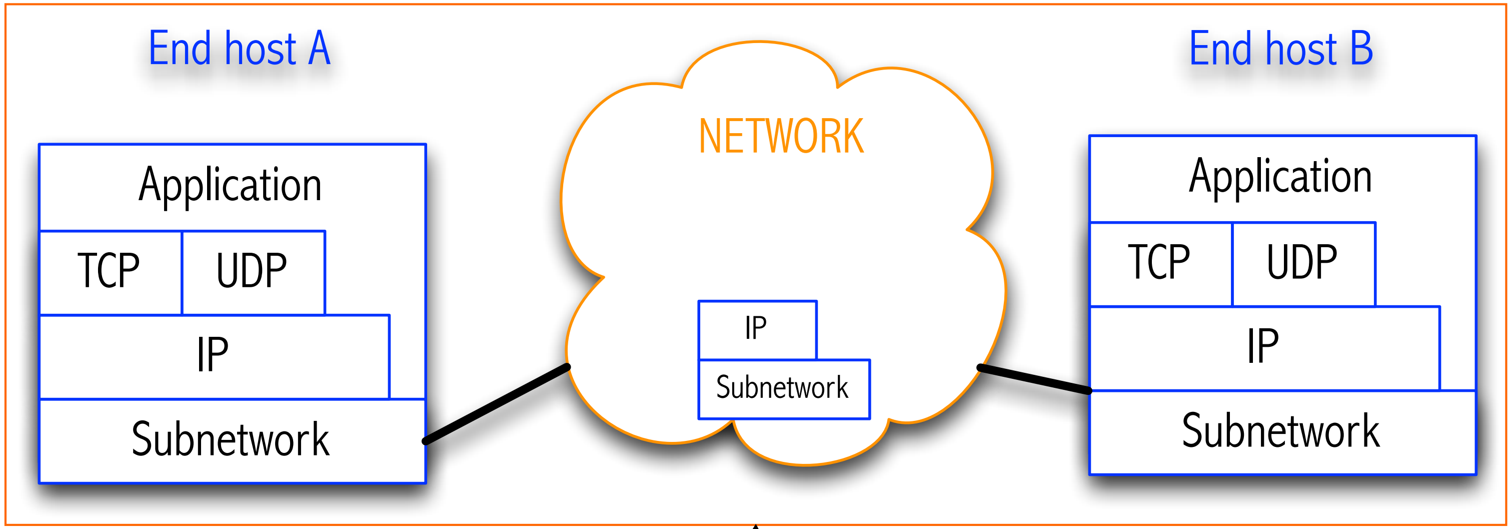
Network Architecture

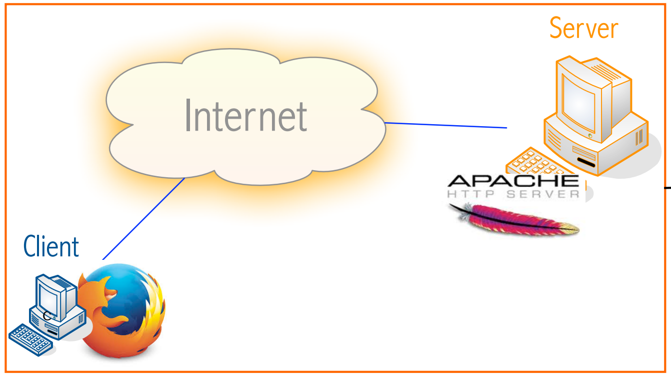
- CHANNEL
  - *Connects* applications
  - *Logical* channels
  - Hosts must be identified:
    - IP address
  - Applications/processes must be identified:
    - Port numbers

How to hide network **complexity** from application **programmers?**

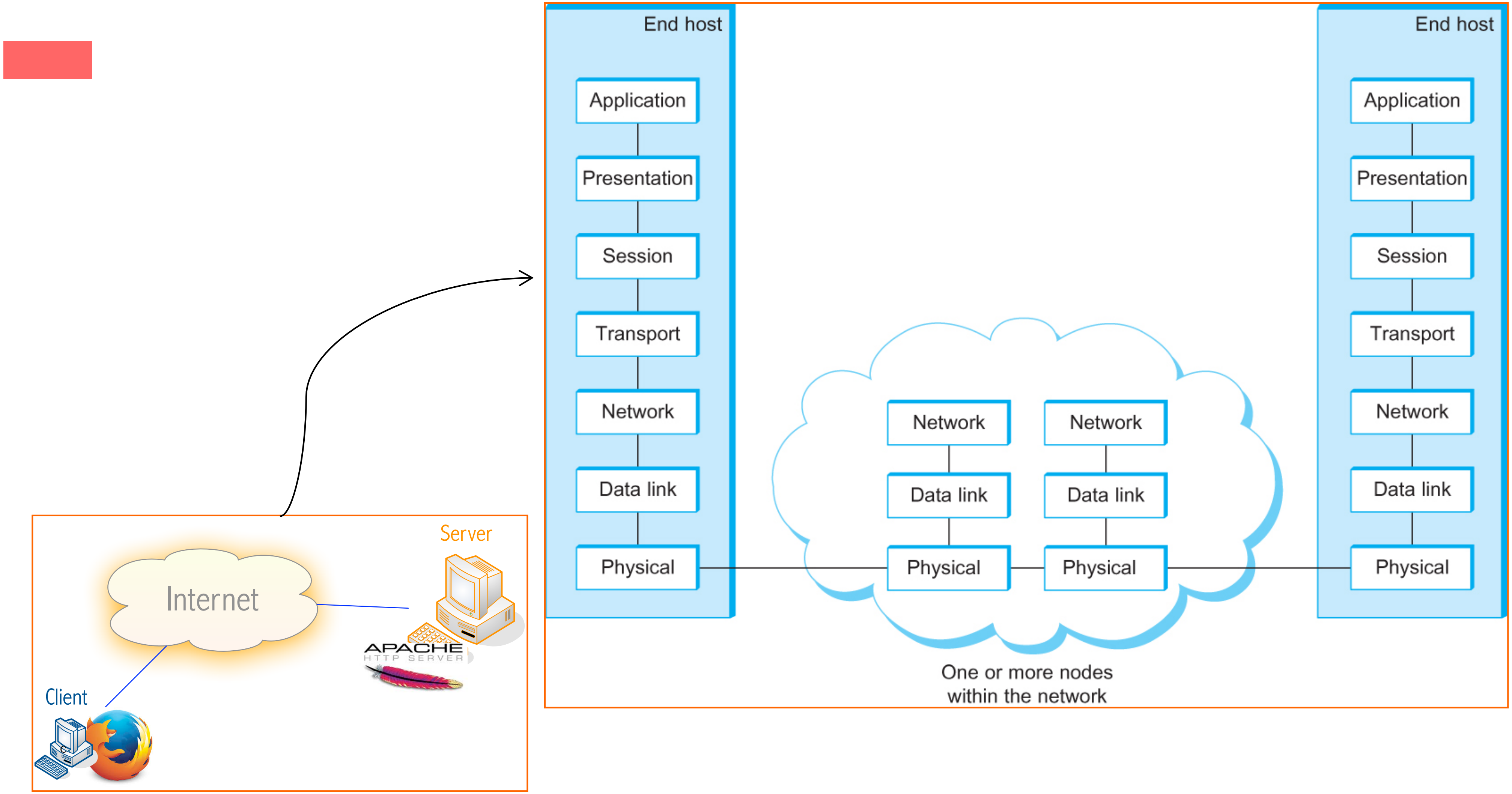


TCP/IP architecture



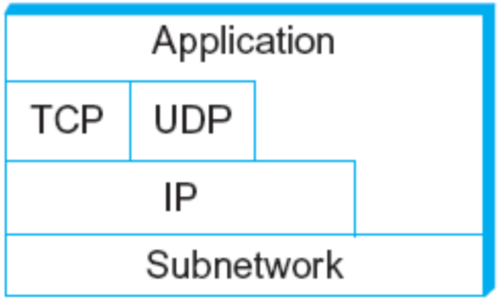


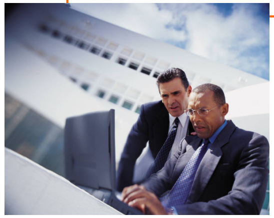
OSI Architecture



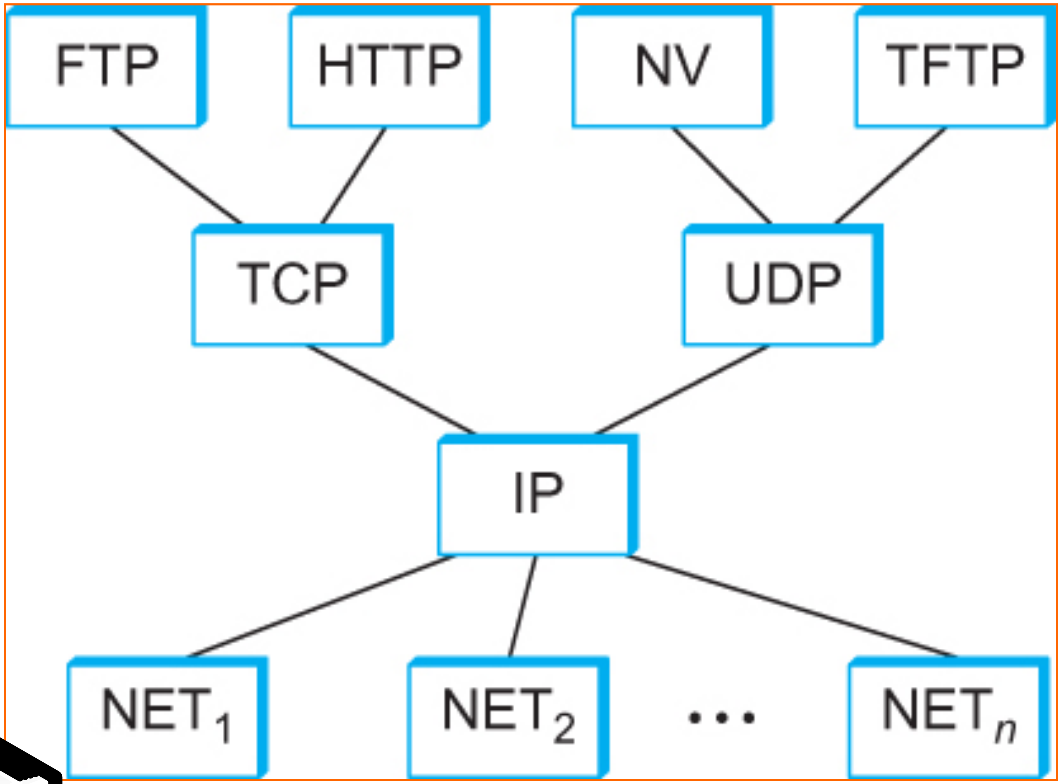
Example

Internet architecture





A protocol graph of a host that implements the Internet architecture

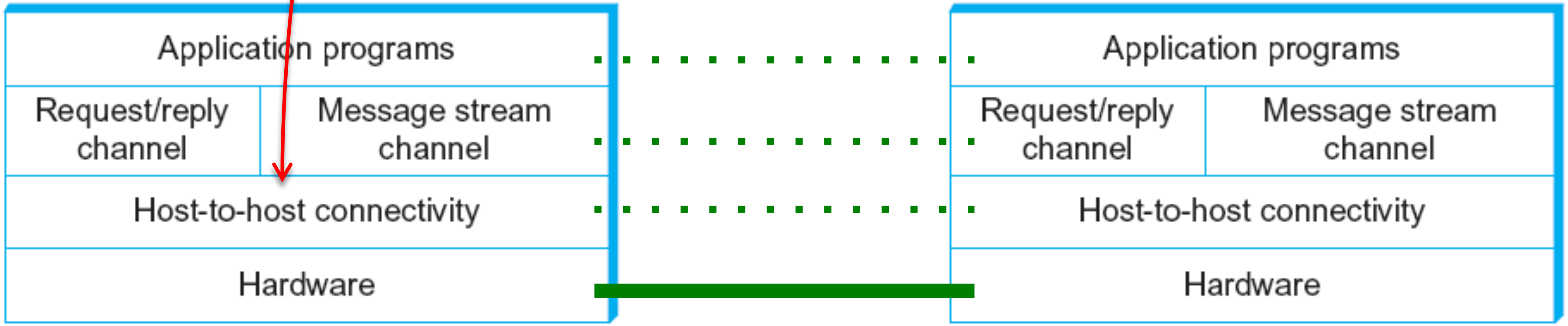


Protocols and Services

The abstractions exposed by the architecture

Protocols: building blocks of a network architecture

- A protocol defines two interfaces:
  - 1. Between the **layers in the** same system
    - Operations available at a specific layer
  - 2. Between **layers of** peer systems (A – B)
    - Messages exchanged between A and B, their syntax and semantics



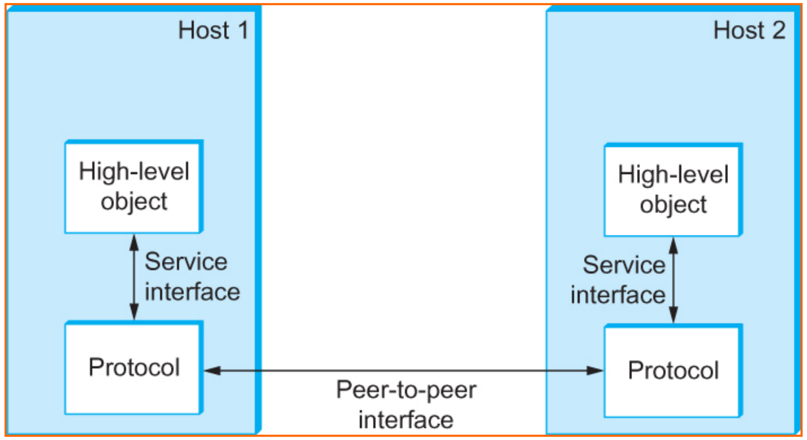
Host A

Host B

Protocols

- Each **protocol object** has two different *interfaces*
  - **service** interface: operations on this protocol within this host
  - **peer-to-peer** interface: messages exchanged with peer (The peer host)

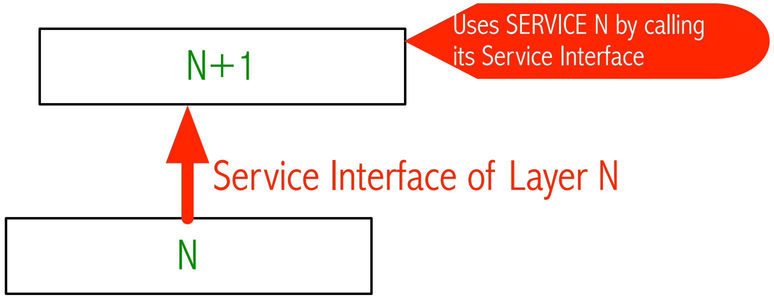
- Term "protocol" has two uses in CN:
  - Specification of **peer-to-peer** interface
  - Module (of software) that implements this interface



Layering in CN

A layer uses the services provided by the layer below

- Implementation details remain hidden
  - Layer N+1 knows nothing about the implementation of layer N
- Service use via Service Interface, only
- OSI 7-layer model

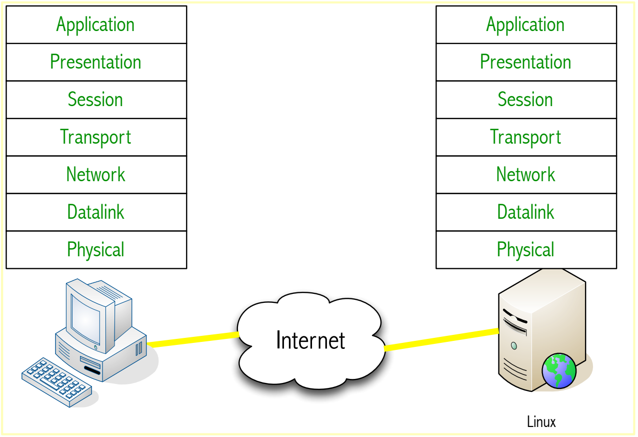


N+1

Uses SERVICE N by calling its Service Interface

Service Interface of Layer N

N



Application
Presentation
Session
Transport
Network
Datalink
Physical

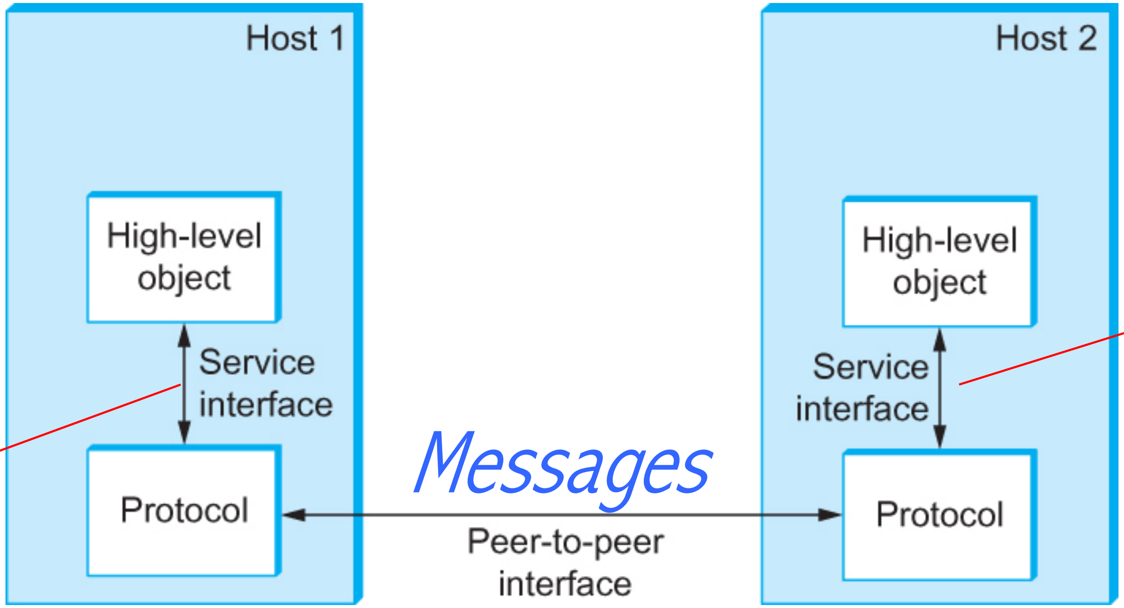
Application
Presentation
Session
Transport
Network
Datalink
Physical

Internet

Linux

Interfaces and Protocols

- This happens at any two-layer interface
  - 3/2
  - 2/1



Operations (methods) available at each layer

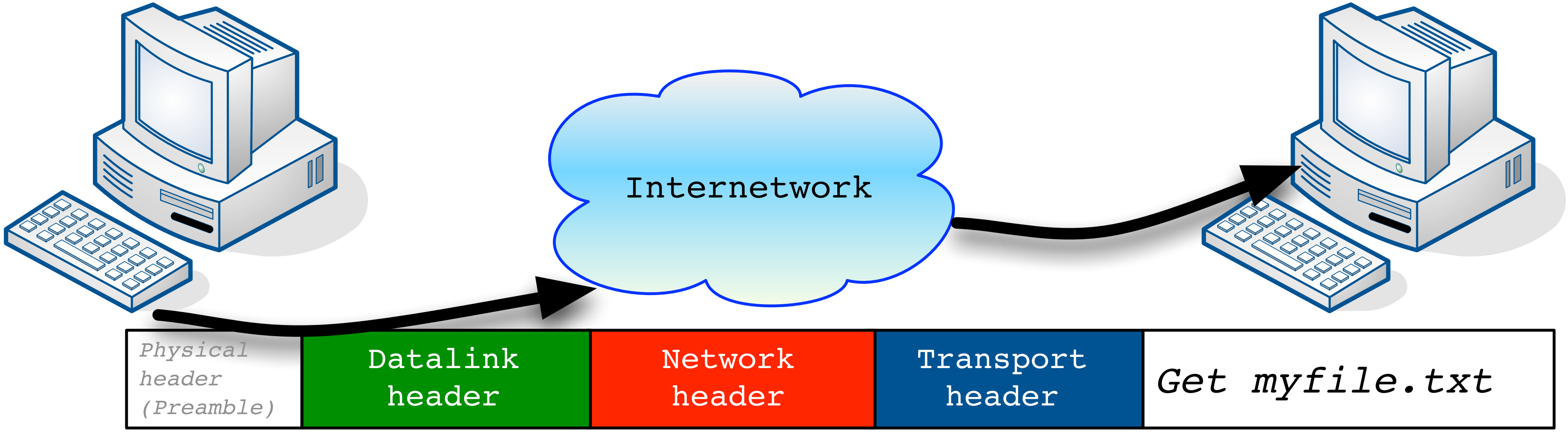
Messages

Operations (methods) available at each layer

Protocols

- □ Protocol Specification: prose, pseudo-code, state transition diagram
- □ Interoperable: when two or more protocols that implement the specification accurately
- □ IETF: Internet Engineering Task Force

Illustration of encapsulation in TCP/IP



Internetwork

Physical header (Preamble)

Datalink header

Network header

Transport header

Get myfile.txt

Get myfile.txt

Application

Presentation

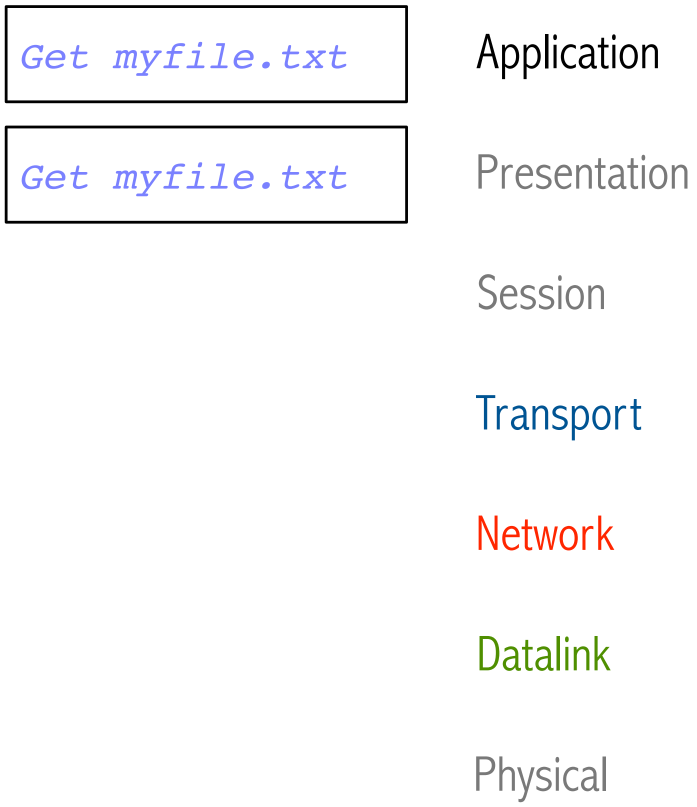
Session

Transport

Network

Datalink

Physical



Get myfile.txt

Get myfile.txt

Application

Presentation

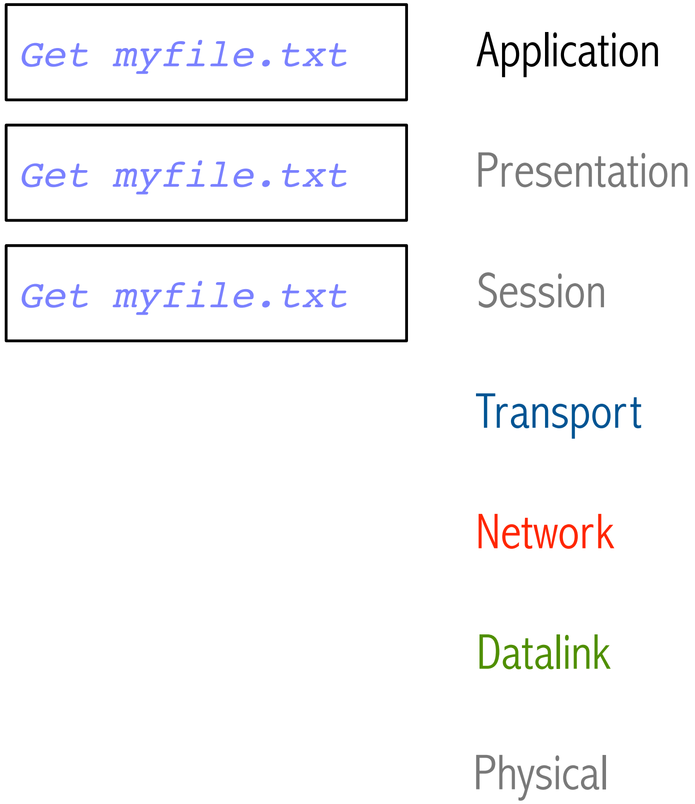
Session

Transport

Network

Datalink

Physical



Get myfile.txt

Get myfile.txt

Get myfile.txt

Application

Presentation

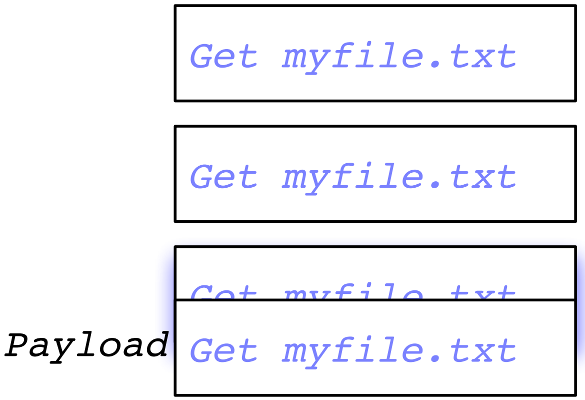
Session

Transport

Network

Datalink

Physical



Get myfile.txt

Get myfile.txt

Get myfile.txt

Payload

Get myfile.txt

Application

Presentation

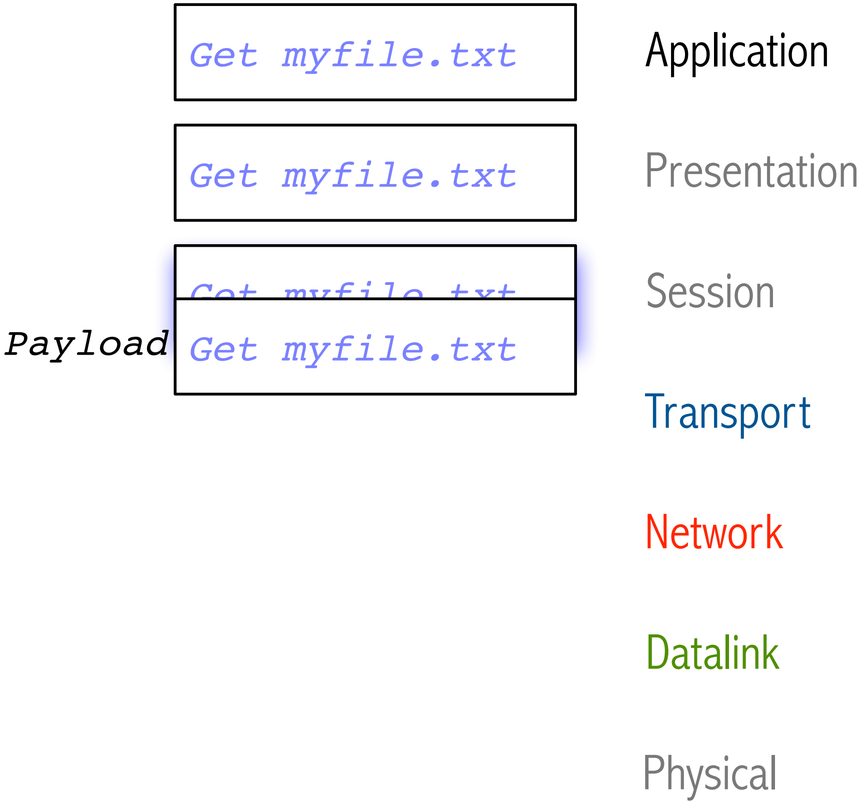
Session

Transport

Network

Datalink

Physical



Get myfile.txt

Application

Get myfile.txt

Presentation

Get myfile.txt

Session

Payload

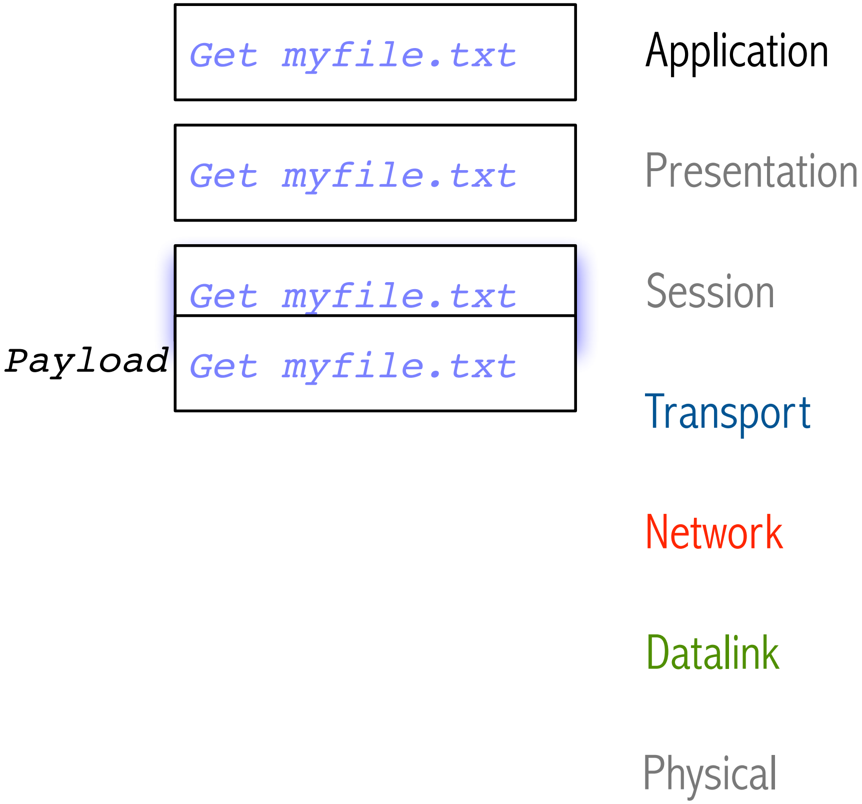
Get myfile.txt

Transport

Network

Datalink

Physical



Get myfile.txt

Application

Get myfile.txt

Presentation

Get myfile.txt

Session

Payload

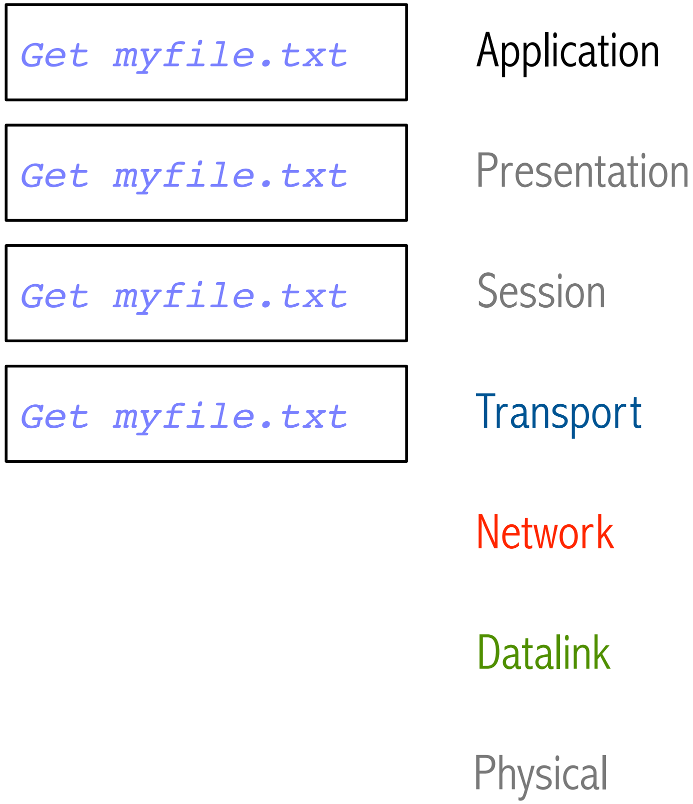
Get myfile.txt

Transport

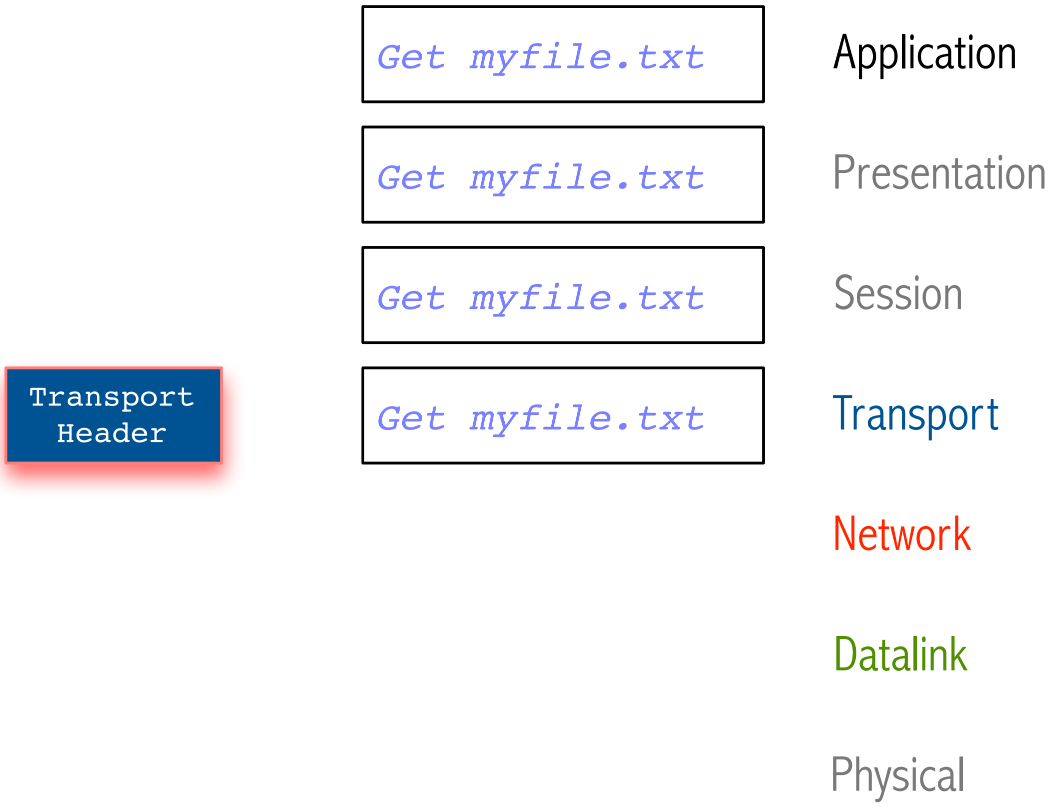
Network

Datalink

Physical



Get myfile.txt	Application
Get myfile.txt	Presentation
Get myfile.txt	Session
Get myfile.txt	Transport
	Network
	Datalink
	Physical



Get myfile.txt

Application

Get myfile.txt

Presentation

Get myfile.txt

Session

Transport Header

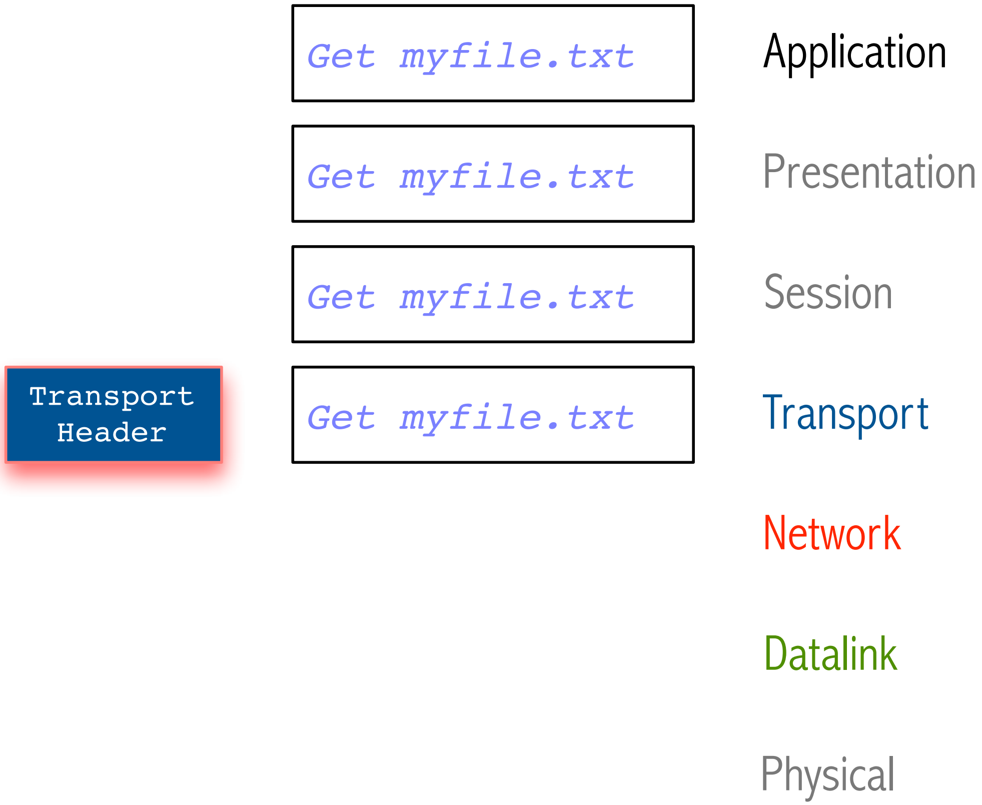
Get myfile.txt

Transport

Network

Datalink

Physical



Get myfile.txt

Get myfile.txt

Get myfile.txt

Transport Header

Get myfile.txt

Application

Presentation

Session

Transport

Network

Datalink

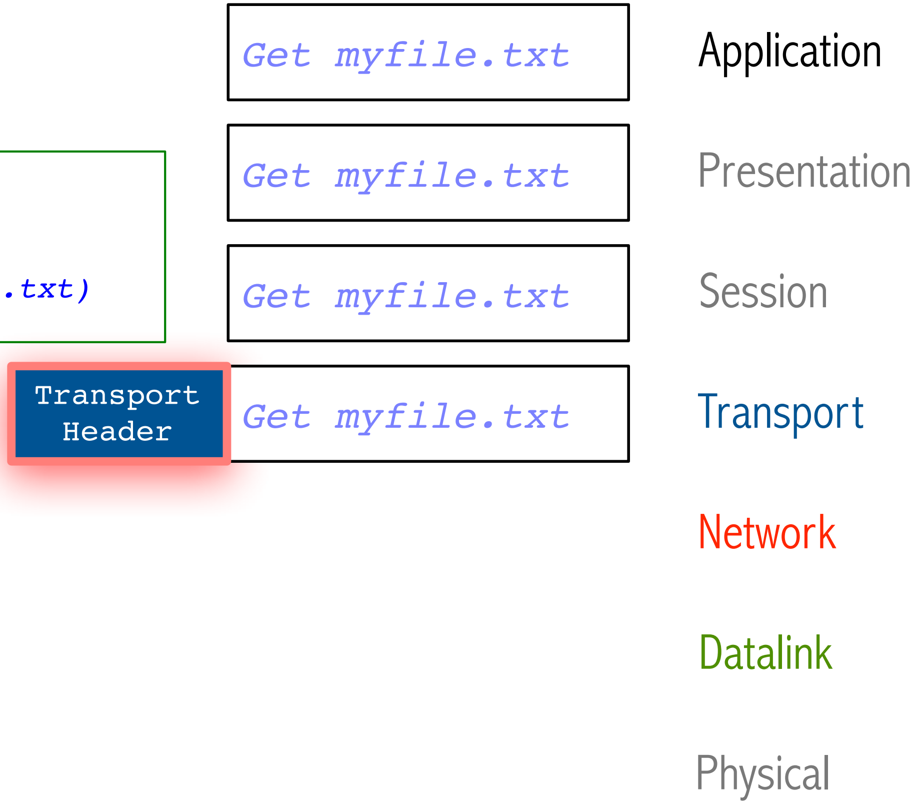
Physical

Encapsulation at layer 4

Get myfile.txt is encapsulated into a Transport Protocol Data Unit (TPDU)

=

Transport Header + Payload (Get myfile.txt)

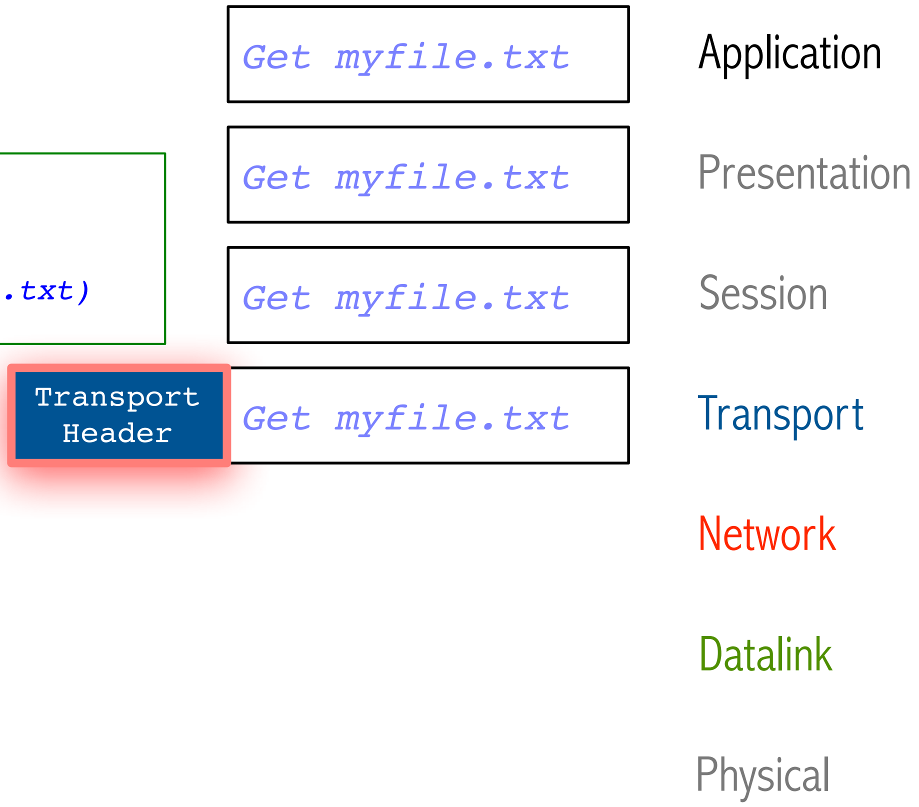


Encapsulation at layer 4

Get myfile.txt is encapsulated into a Transport Protocol Data Unit (TPDU)

=

Transport Header + Payload(Get myfile.txt)



Get myfile.txt

Application

Get myfile.txt

Presentation

Get myfile.txt

Session

Transport Header

Get myfile.txt

Transport

Network

Datalink

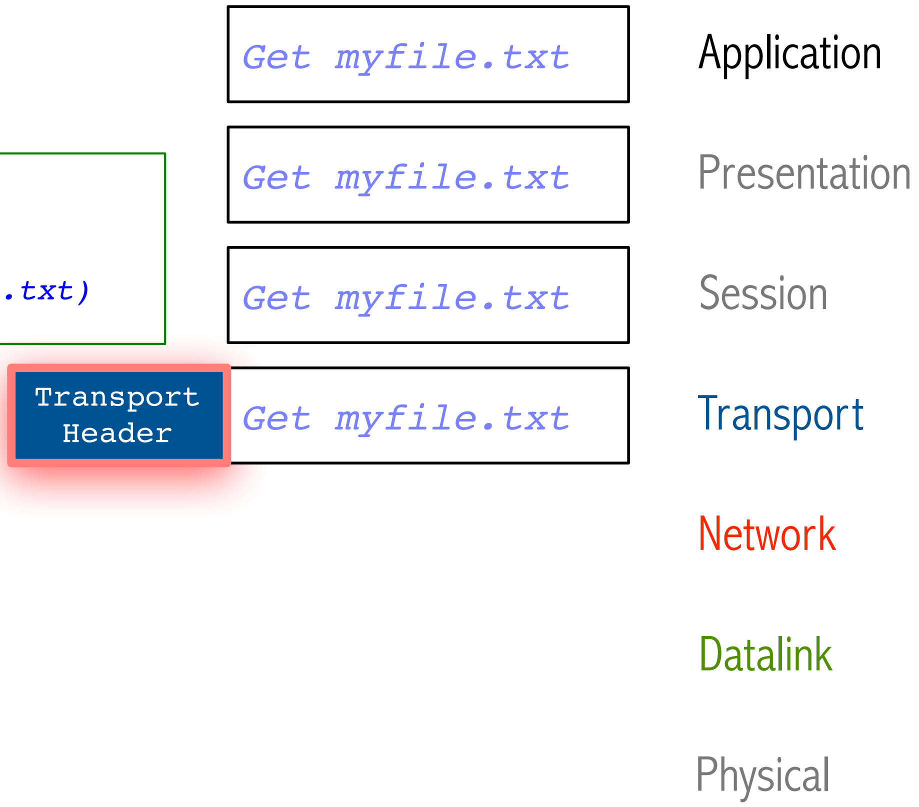
Physical

Encapsulation at layer 4

Get myfile.txt is encapsulated into a Transport Protocol Data Unit (TPDU)

=

Transport Header + Payload (Get myfile.txt)



Get myfile.txt Application

Get myfile.txt Presentation

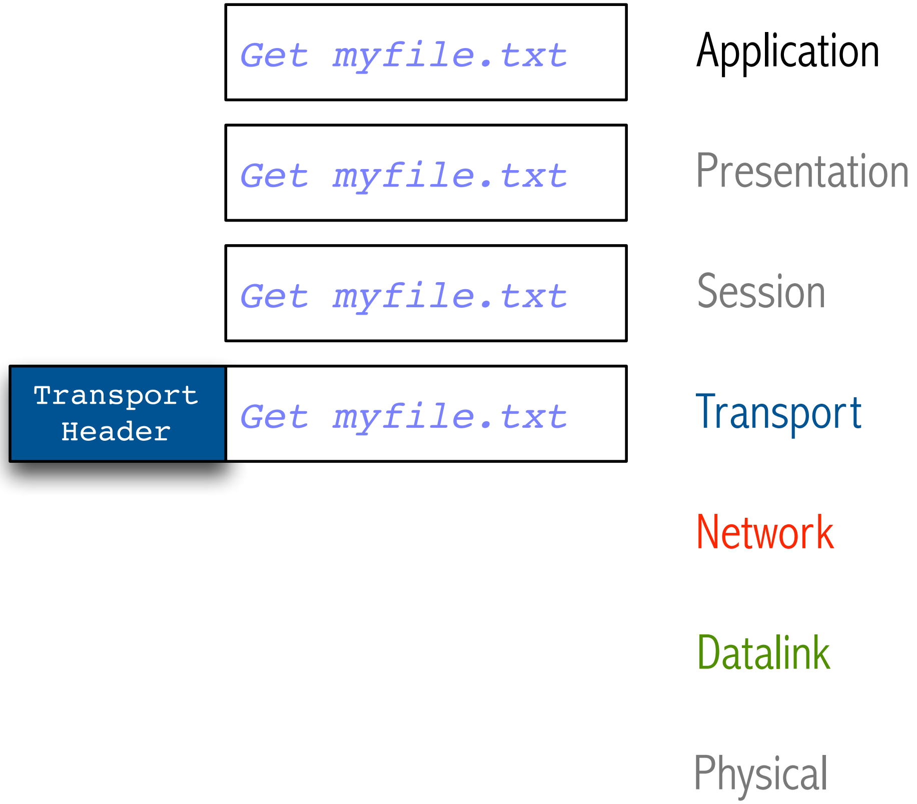
Get myfile.txt Session

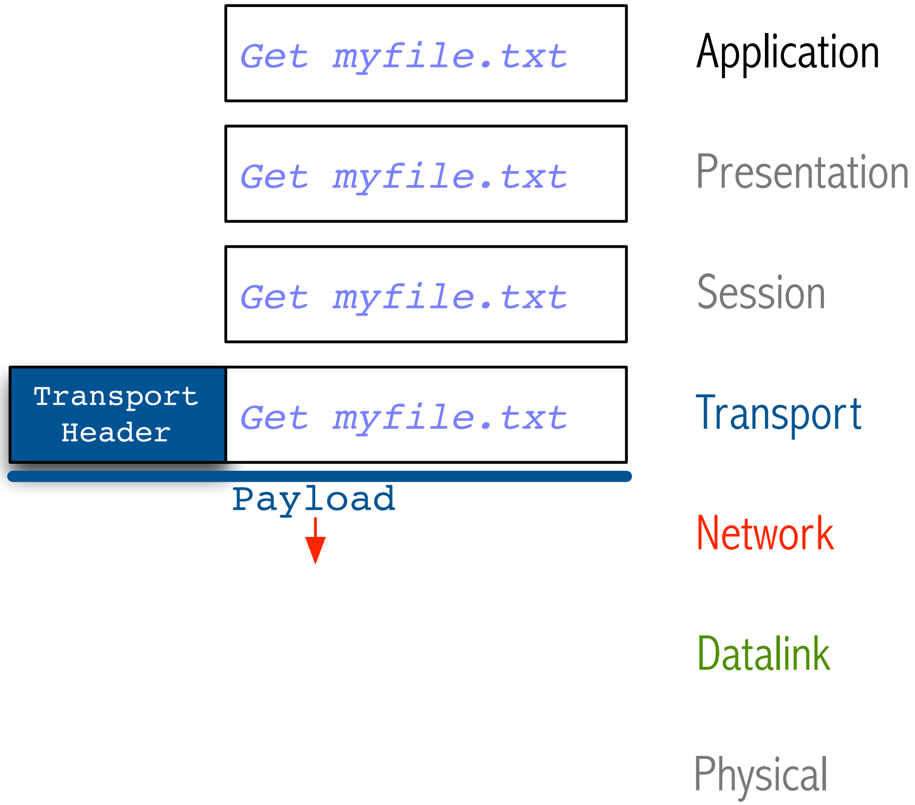
Transport Header Get myfile.txt Transport

Network

Datalink

Physical


Get myfile.txt
Application
Get myfile.txt
Presentation
Get myfile.txt
Session
Transport Header
Get myfile.txt
Transport
Network
Datalink
Physical



Get myfile.txt

Application

Get myfile.txt

Presentation

Get myfile.txt

Session

Transport Header

Get myfile.txt

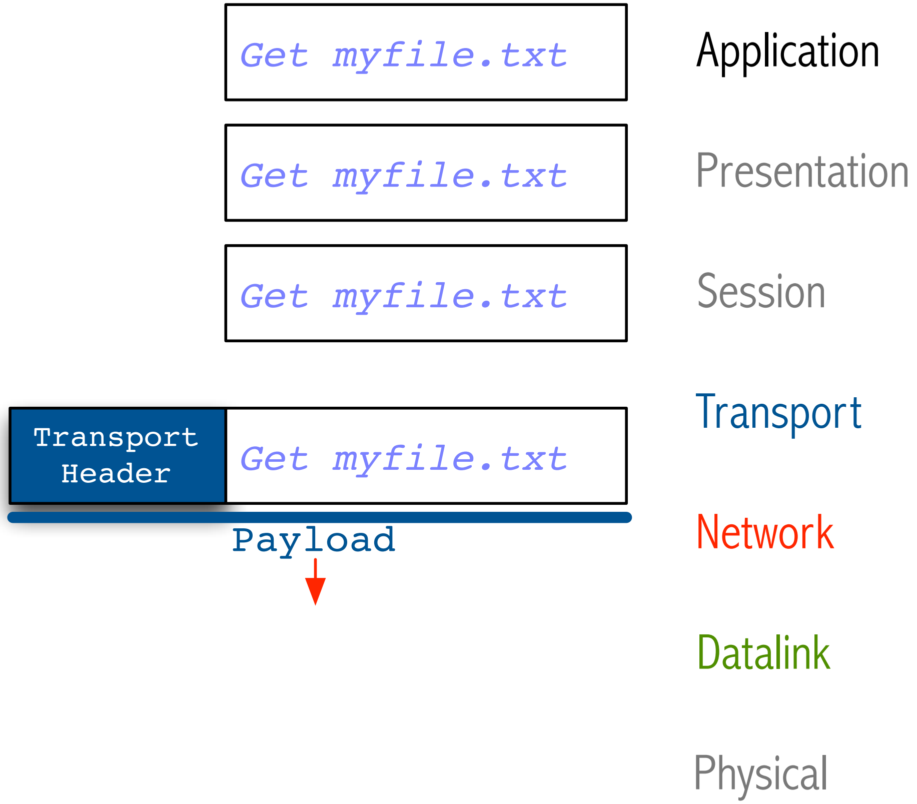
Transport

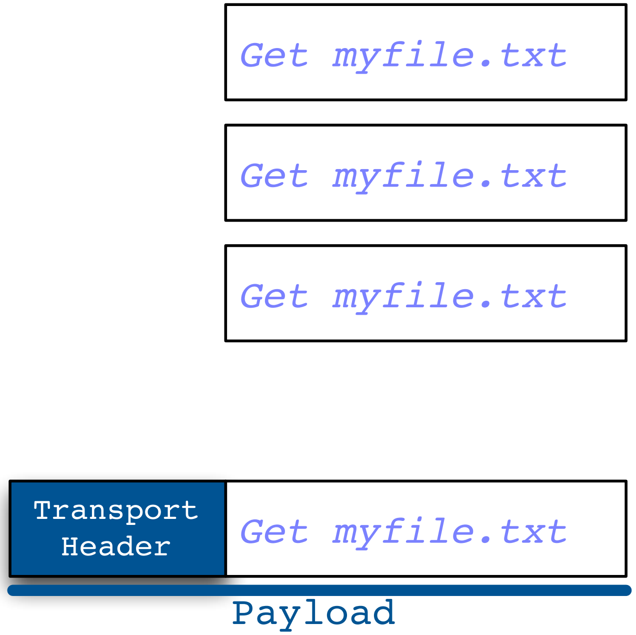
Payload

Network

Datalink

Physical


Get myfile.txt
Application
Get myfile.txt
Presentation
Get myfile.txt
Session
Transport Header
Get myfile.txt
Transport
Payload
Network
Datalink
Physical



Get myfile.txt

Get myfile.txt

Get myfile.txt

Transport Header

Get myfile.txt

Payload

Application

Presentation

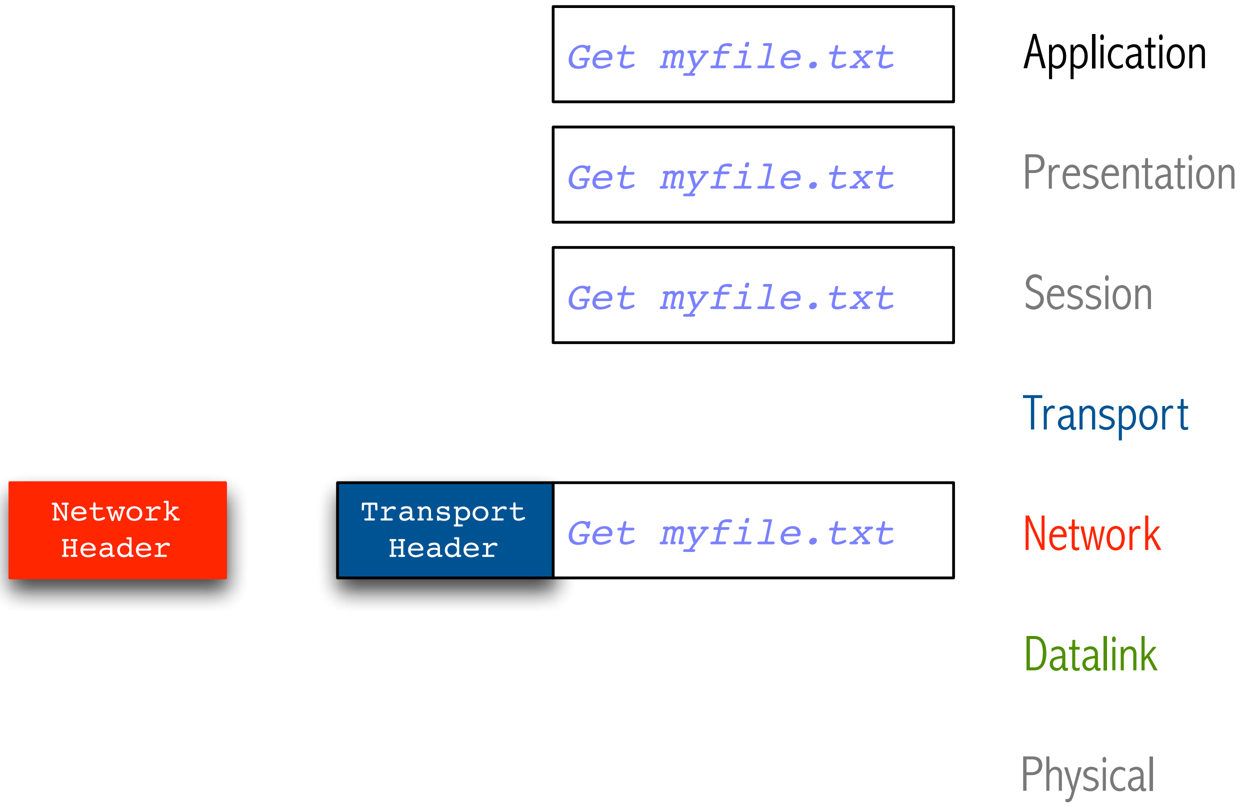
Session

Transport

Network

Datalink

Physical



Get myfile.txt Application

Get myfile.txt Presentation

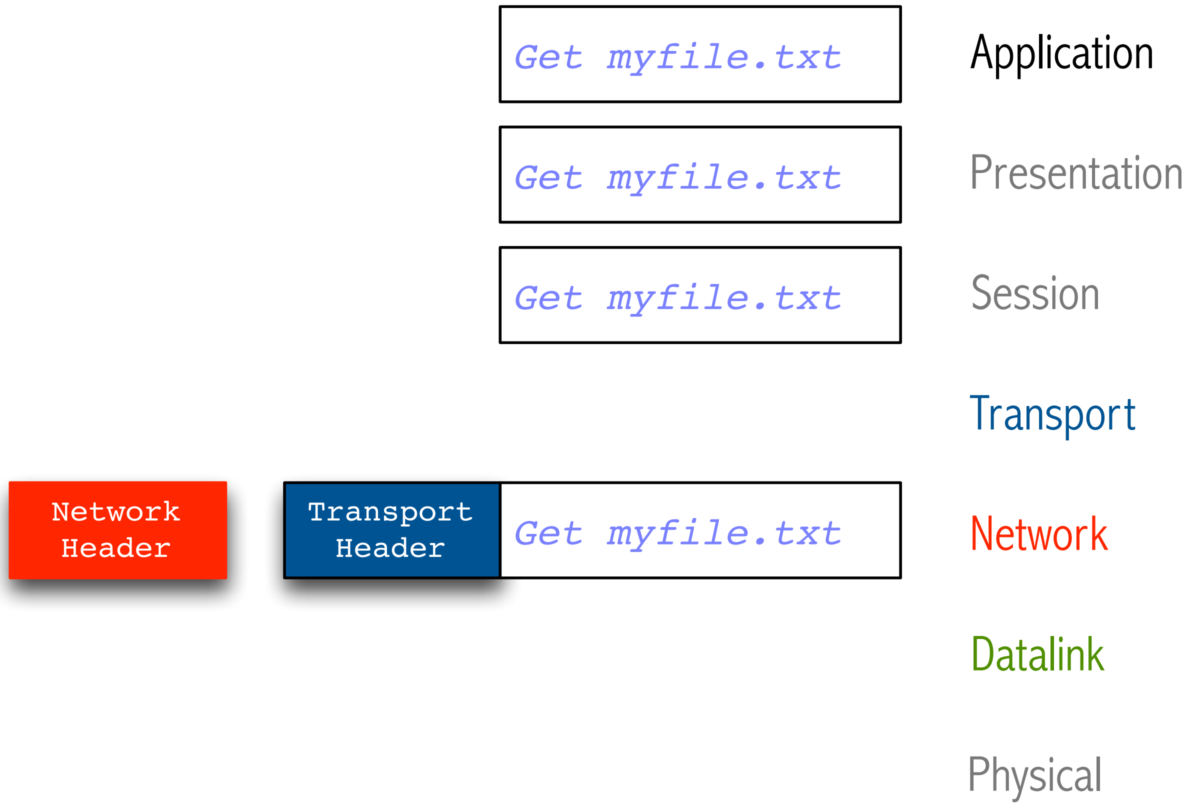
Get myfile.txt Session

Transport

Network Header Transport Header Get myfile.txt Network

Datalink

Physical



Get myfile.txt

Application

Get myfile.txt

Presentation

Get myfile.txt

Session

Transport

Network Header

Transport Header

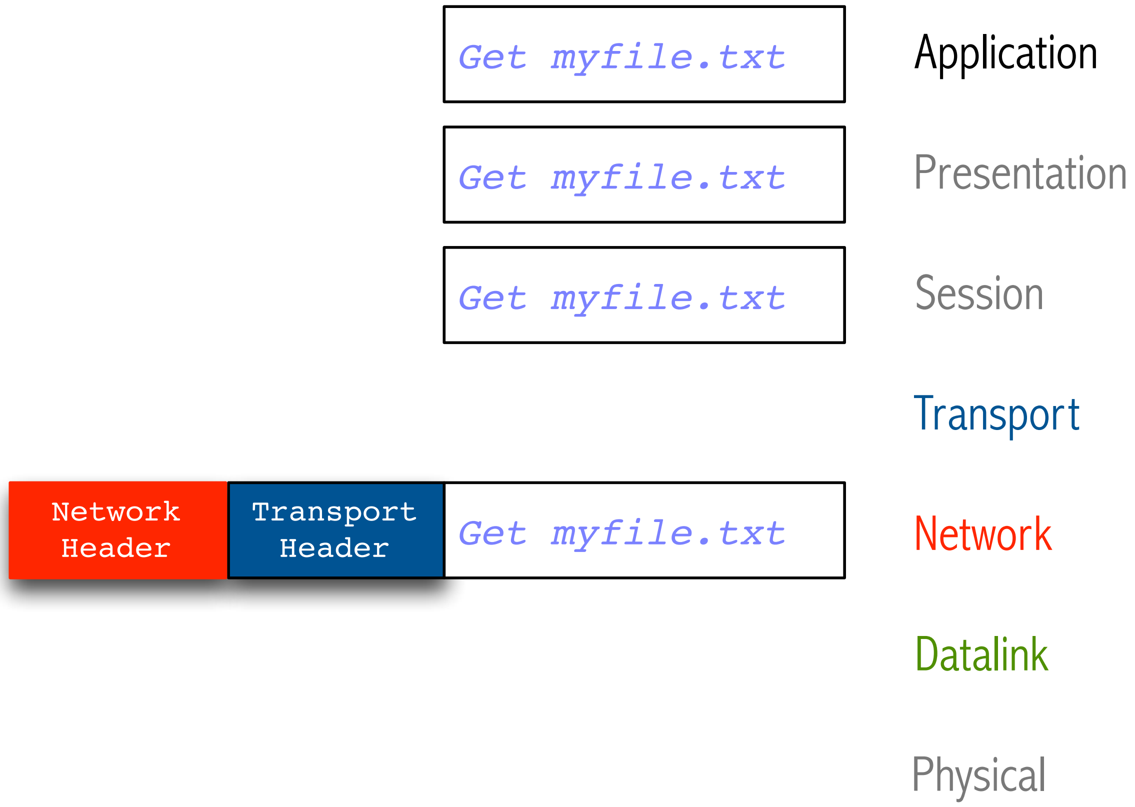
Get myfile.txt

Network

Datalink

Physical

Encapsulation at layer 3



Get myfile.txt Application

Get myfile.txt Presentation

Get myfile.txt Session

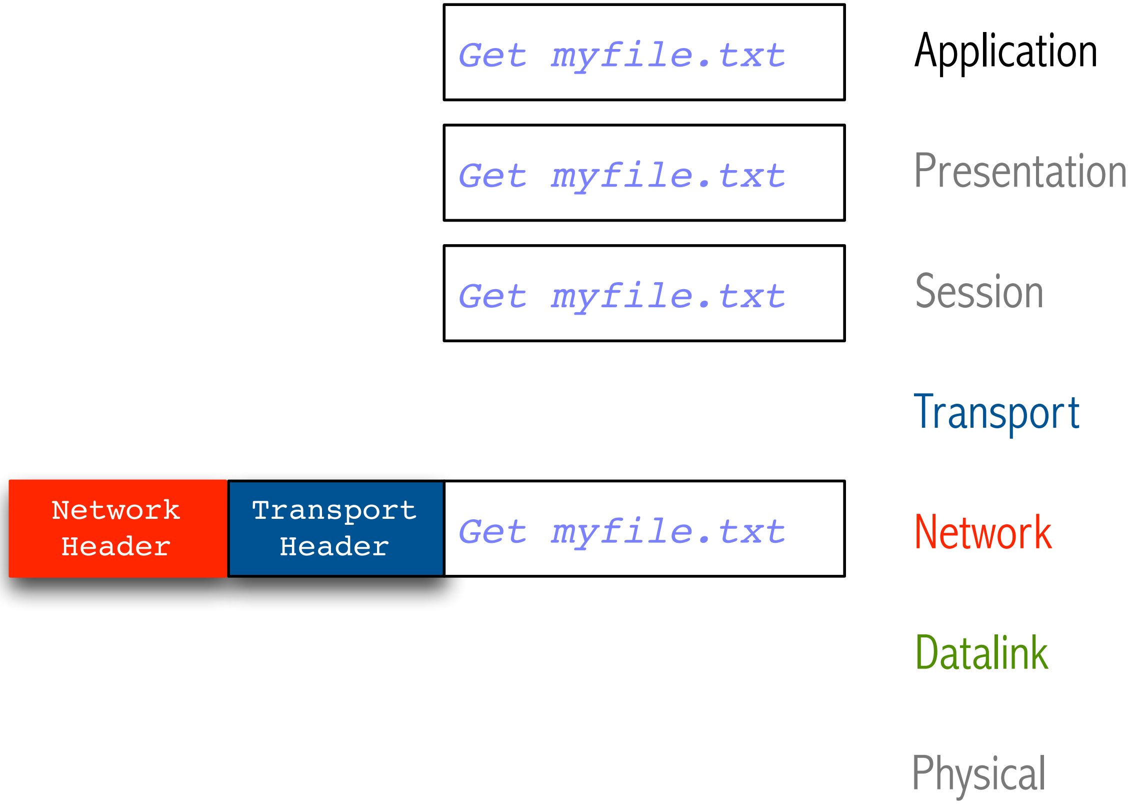
Transport

Network Header | Transport Header | *Get myfile.txt* Network

Datalink

Physical

Encapsulation at layer 3



Get myfile.txt — Application

Get myfile.txt — Presentation

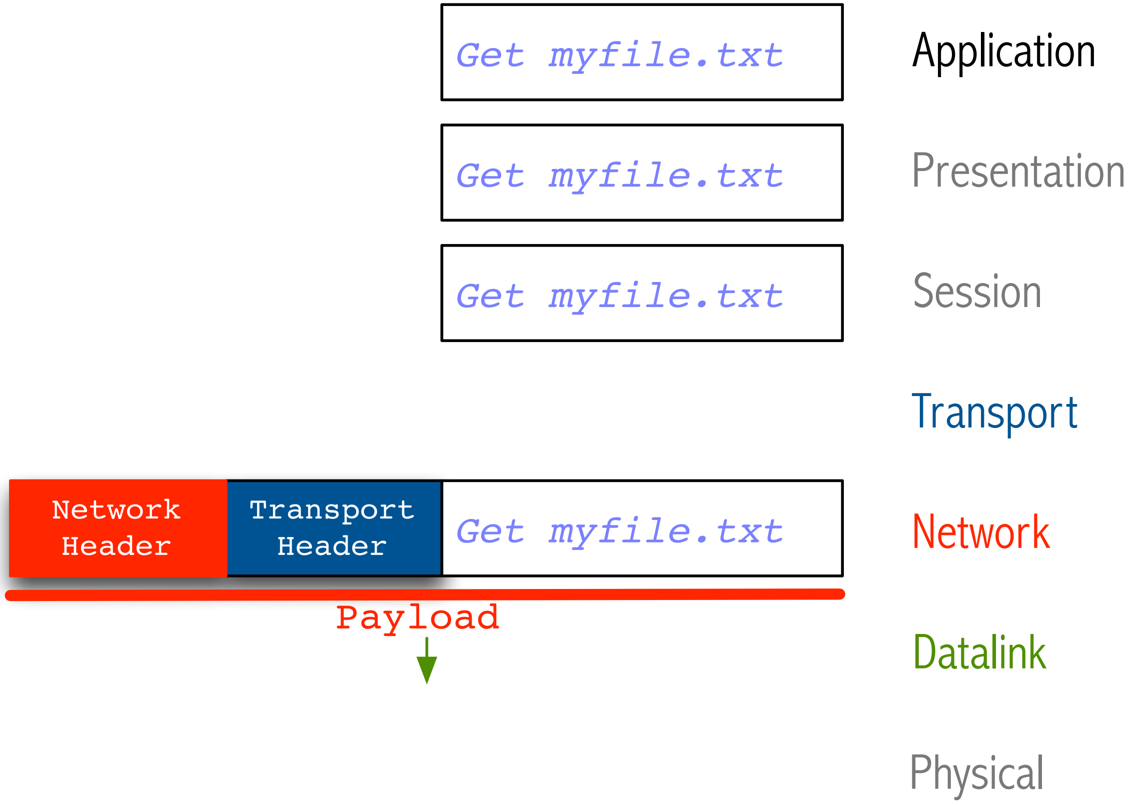
Get myfile.txt — Session

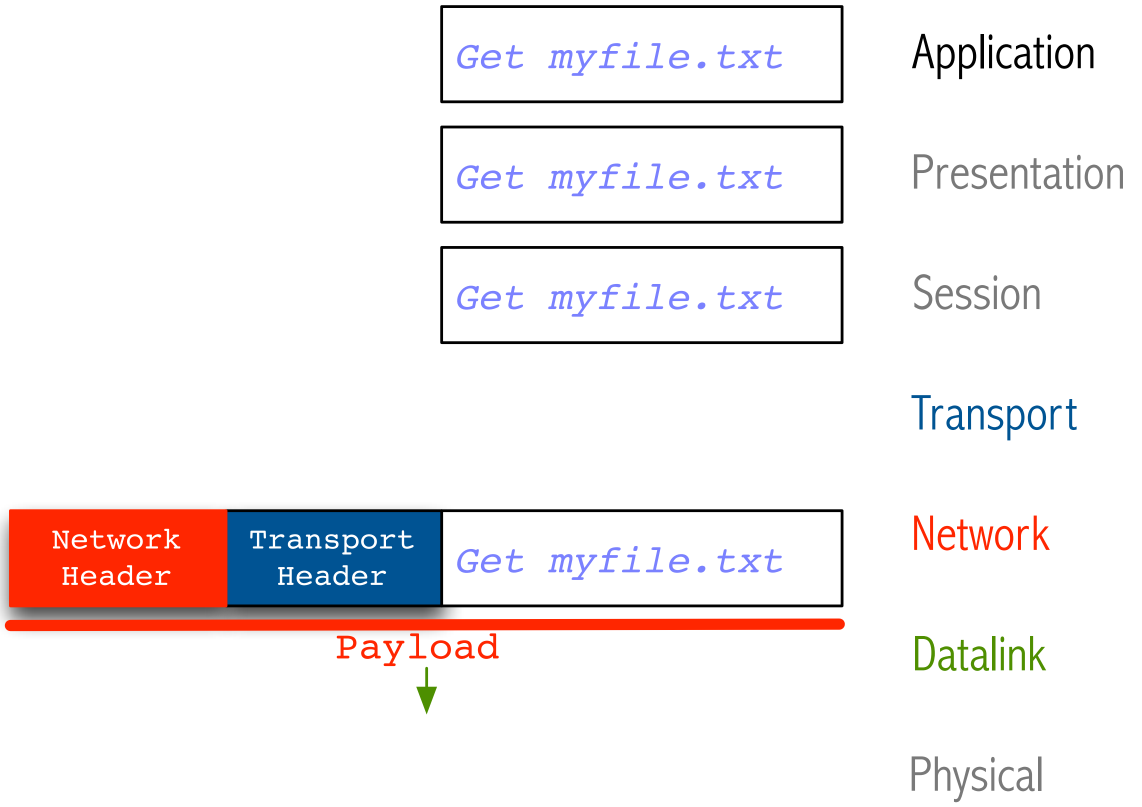
Transport

Network Header | Transport Header | Get myfile.txt — Network

Datalink

Physical


Get myfile.txt
Application
Get myfile.txt
Presentation
Get myfile.txt
Session
Transport
Network Header
Transport Header
Get myfile.txt
Network
Payload
Datalink
Physical



Get myfile.txt

Application

Get myfile.txt

Presentation

Get myfile.txt

Session

Transport

Network Header

Transport Header

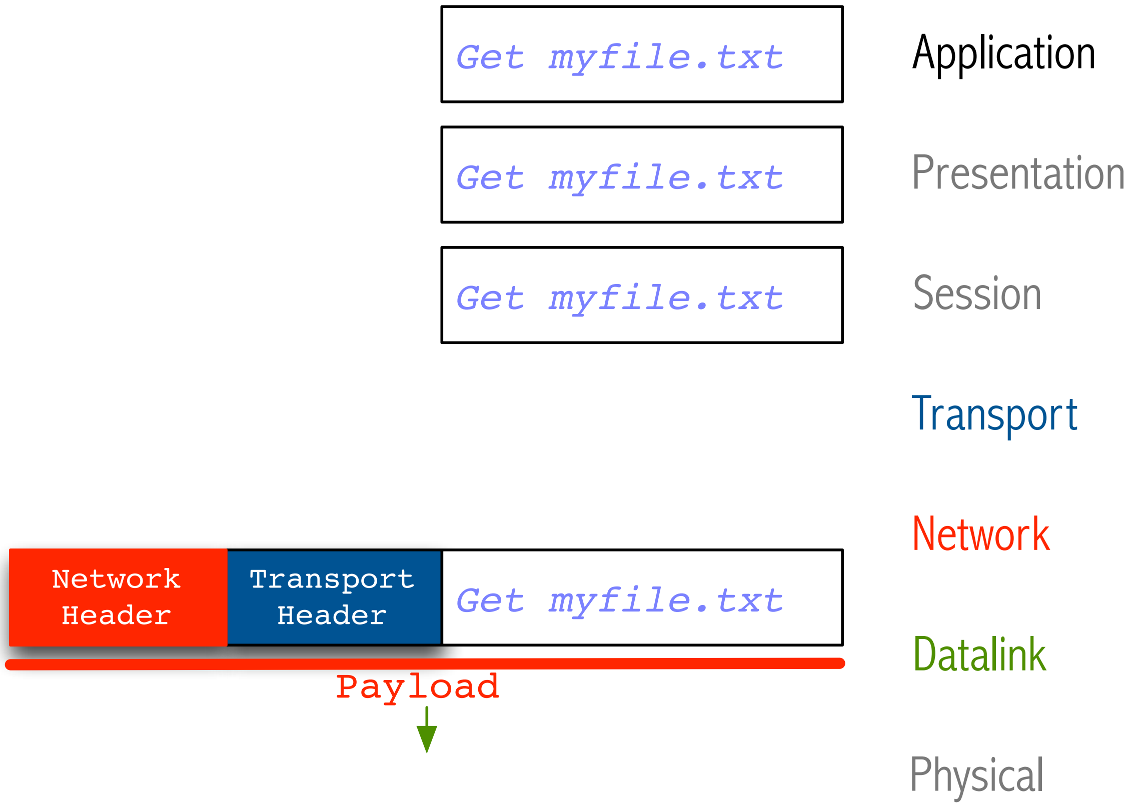
Get myfile.txt

Network

Payload

Datalink

Physical



Get myfile.txt Application

Get myfile.txt Presentation

Get myfile.txt Session

Transport

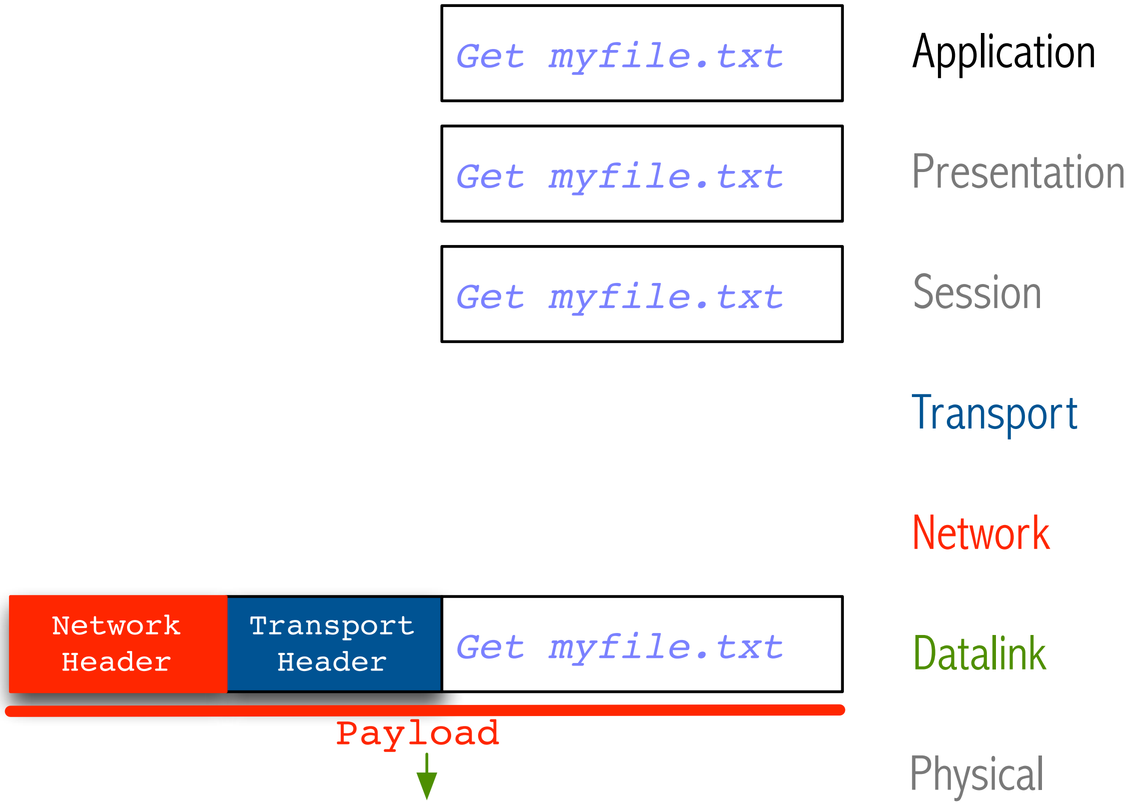
Network

Network Header | Transport Header | *Get myfile.txt*

Datalink

Payload

Physical



Get myfile.txt

Application

Get myfile.txt

Presentation

Get myfile.txt

Session

Transport

Network

Network Header

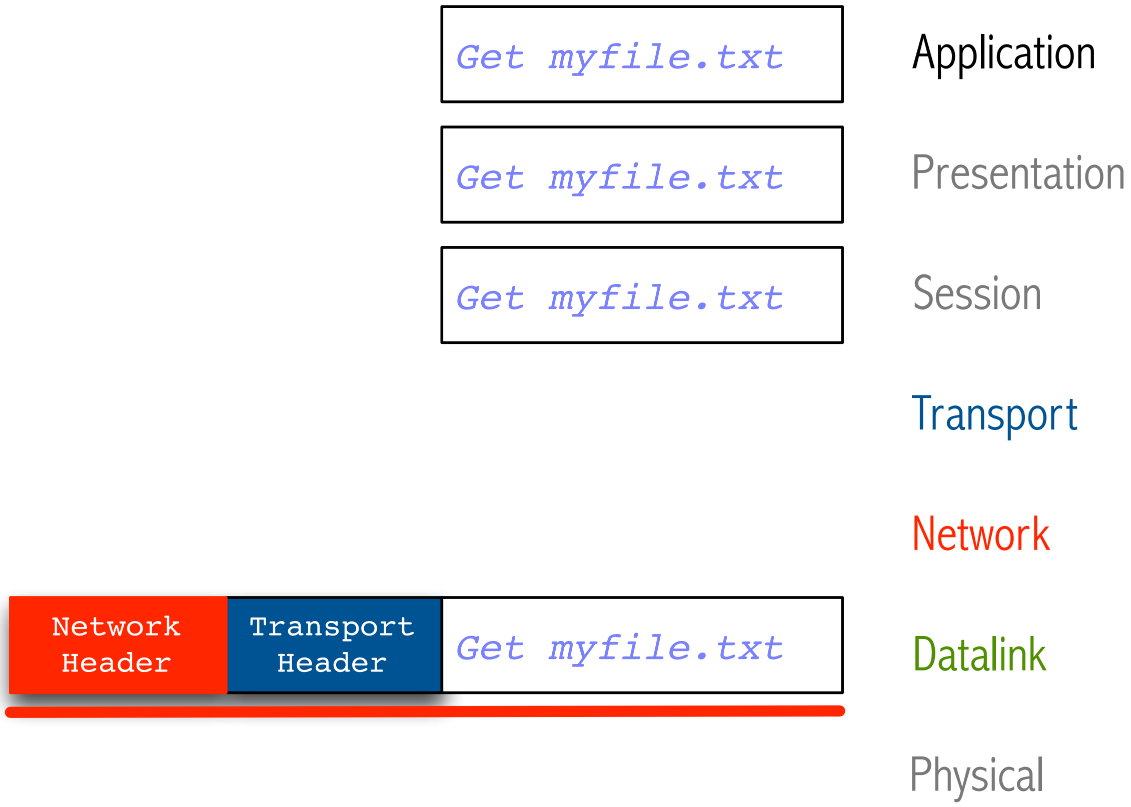
Transport Header

Get myfile.txt

Datalink

Payload

Physical



Get myfile.txt

Application

Get myfile.txt

Presentation

Get myfile.txt

Session

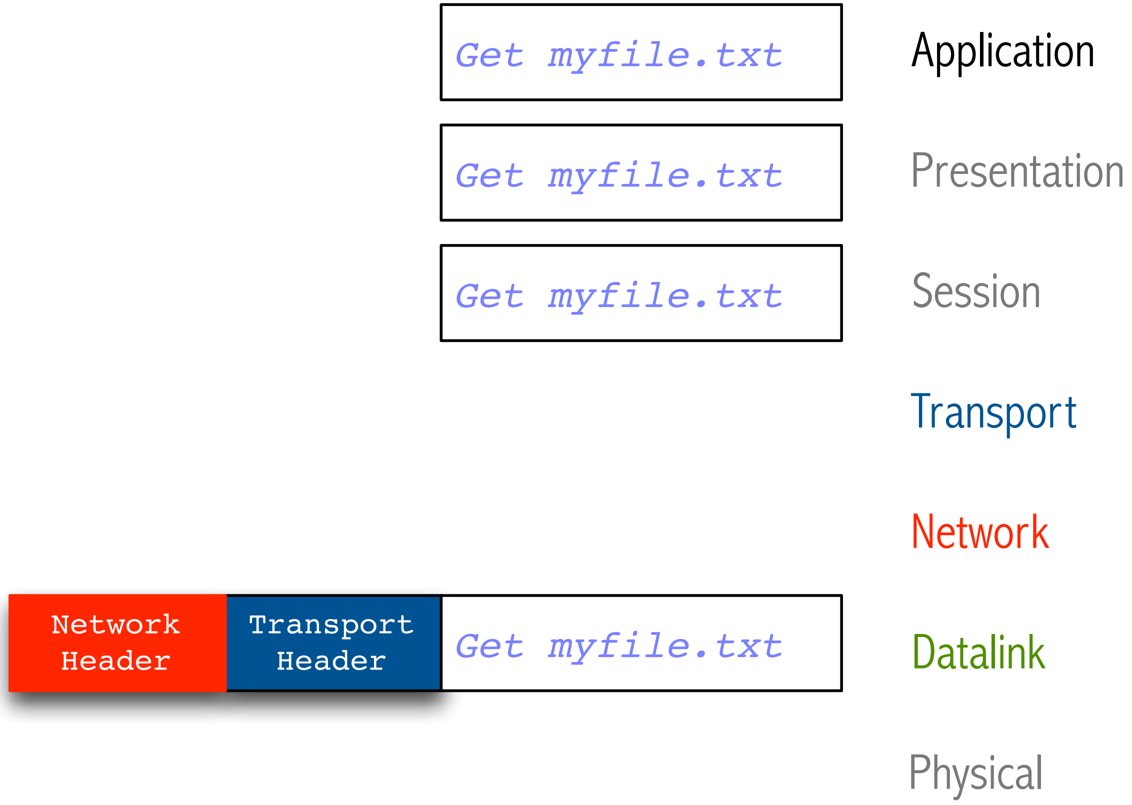
Transport

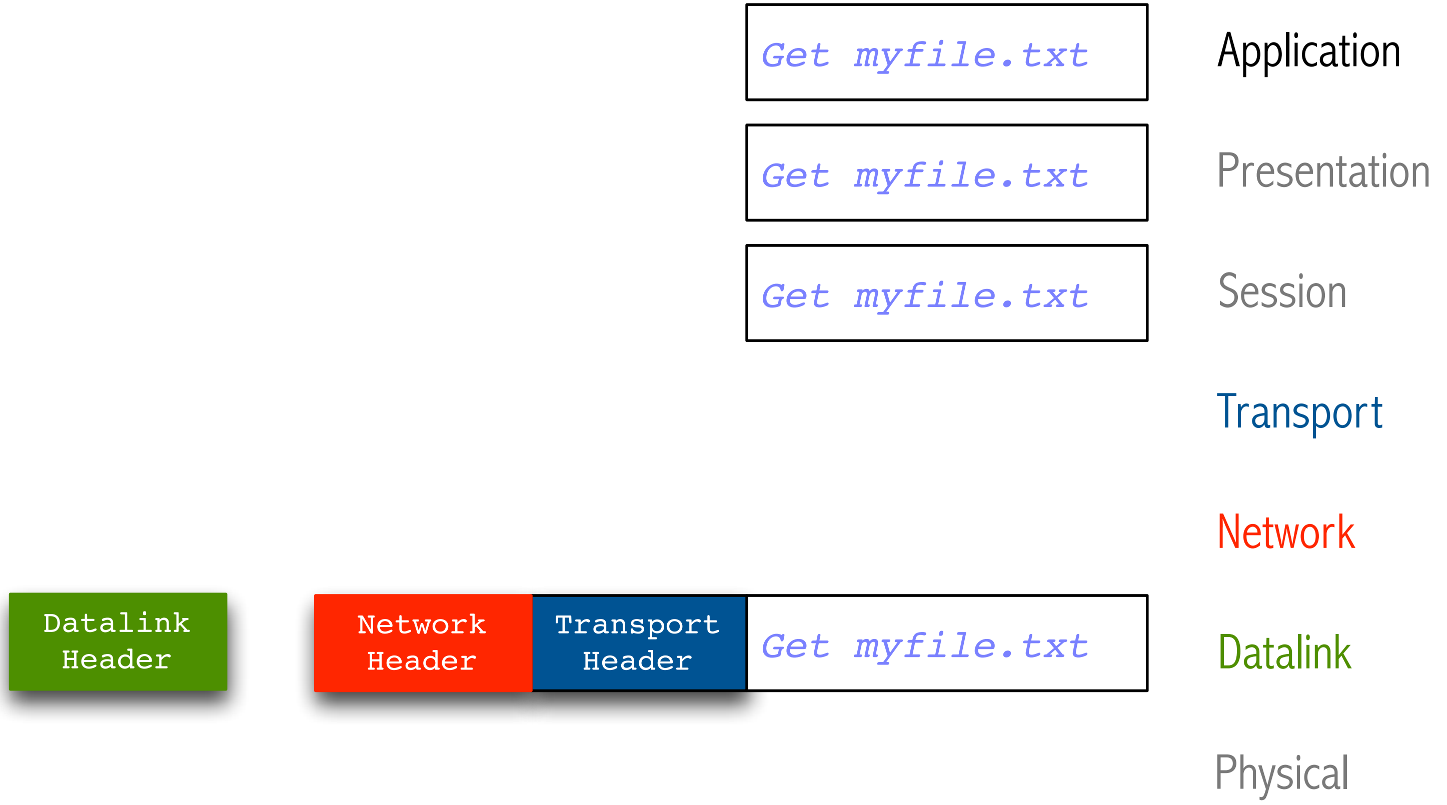
Network

Network Header | Transport Header | Get myfile.txt

Datalink

Physical


Get myfile.txt
Application
Get myfile.txt
Presentation
Get myfile.txt
Session
Transport
Network
Network Header
Transport Header
Get myfile.txt
Datalink
Physical



Get myfile.txt

Application

Get myfile.txt

Presentation

Get myfile.txt

Session

Transport

Network

Datalink Header

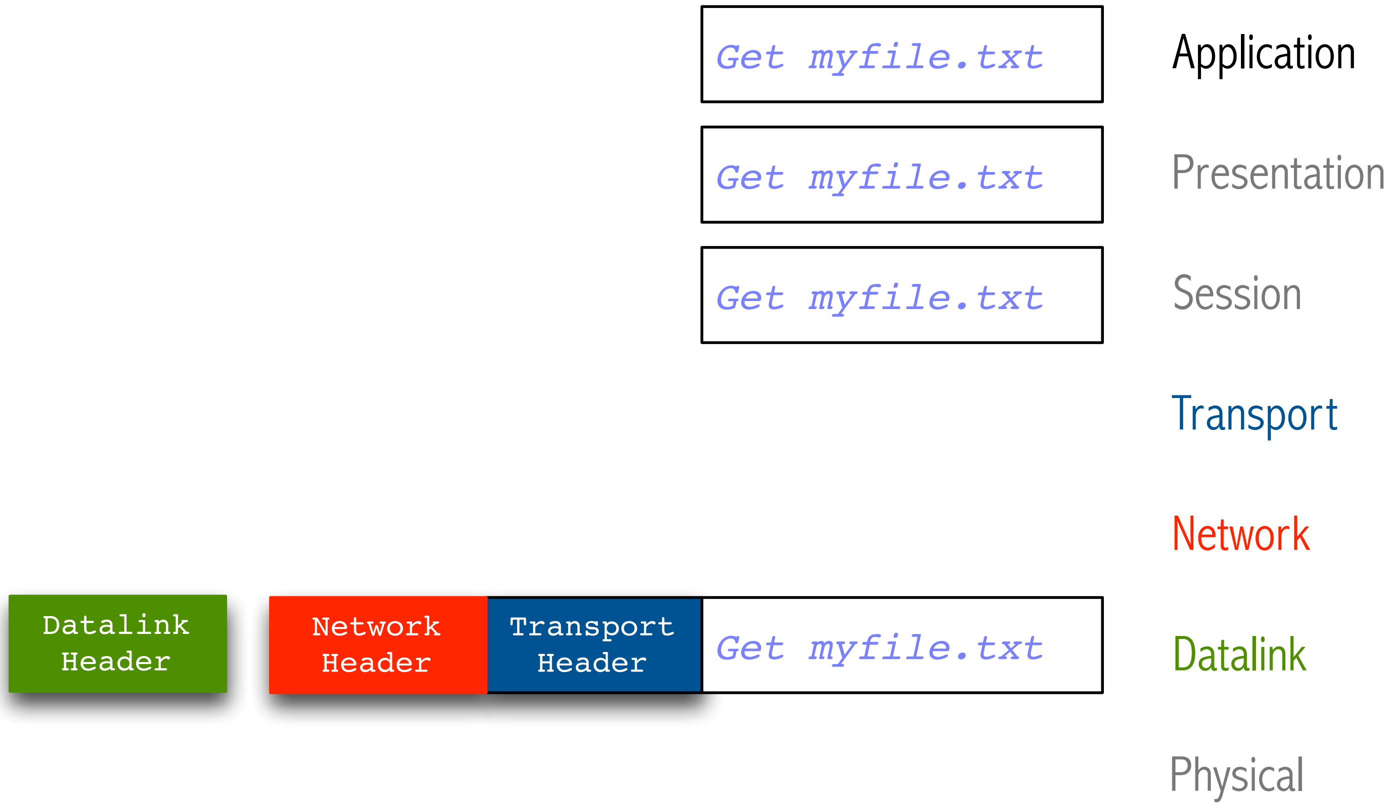
Network Header

Transport Header

Get myfile.txt

Datalink

Physical



Get myfile.txt

Application

Get myfile.txt

Presentation

Get myfile.txt

Session

Transport

Network

Datalink Header

Network Header

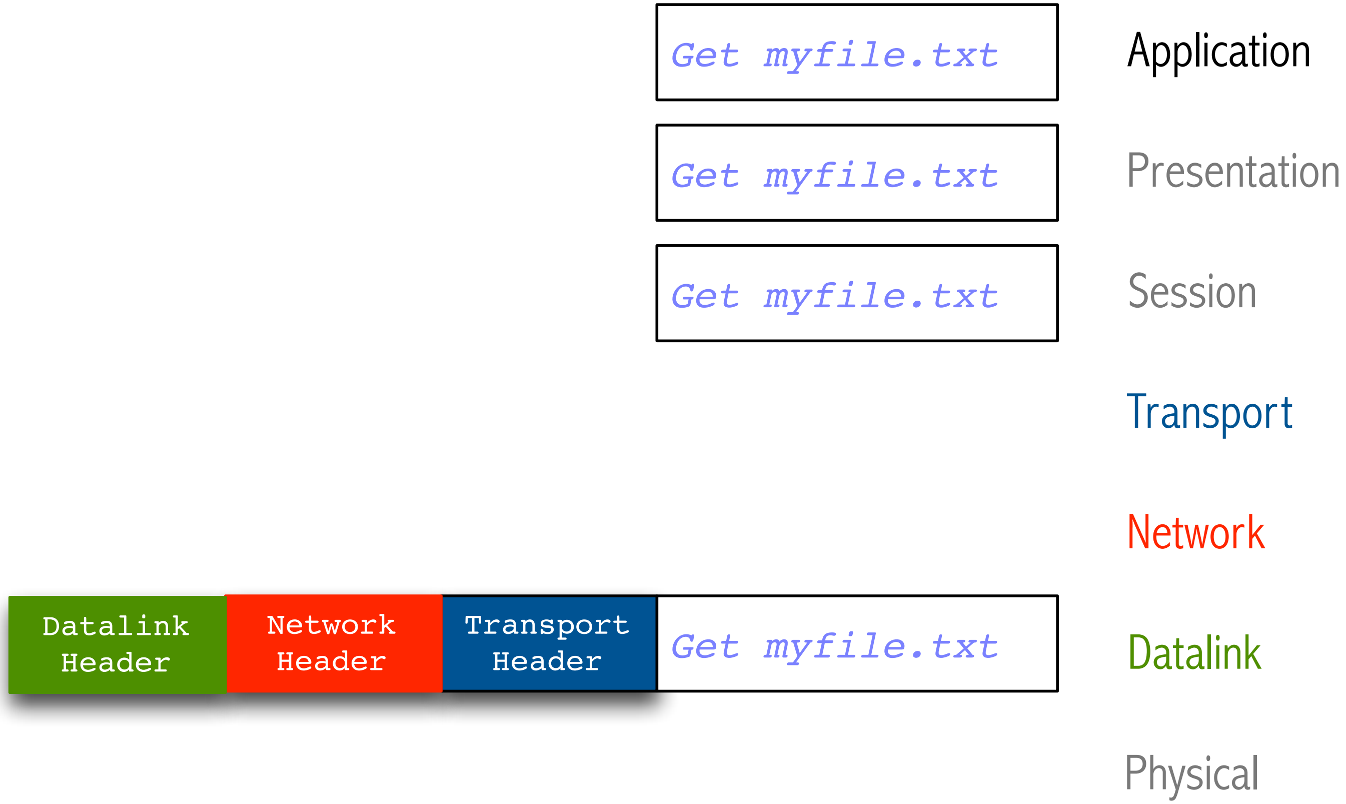
Transport Header

Get myfile.txt

Datalink

Physical

Encapsulation at layer 2



Get myfile.txt Application

Get myfile.txt Presentation

Get myfile.txt Session

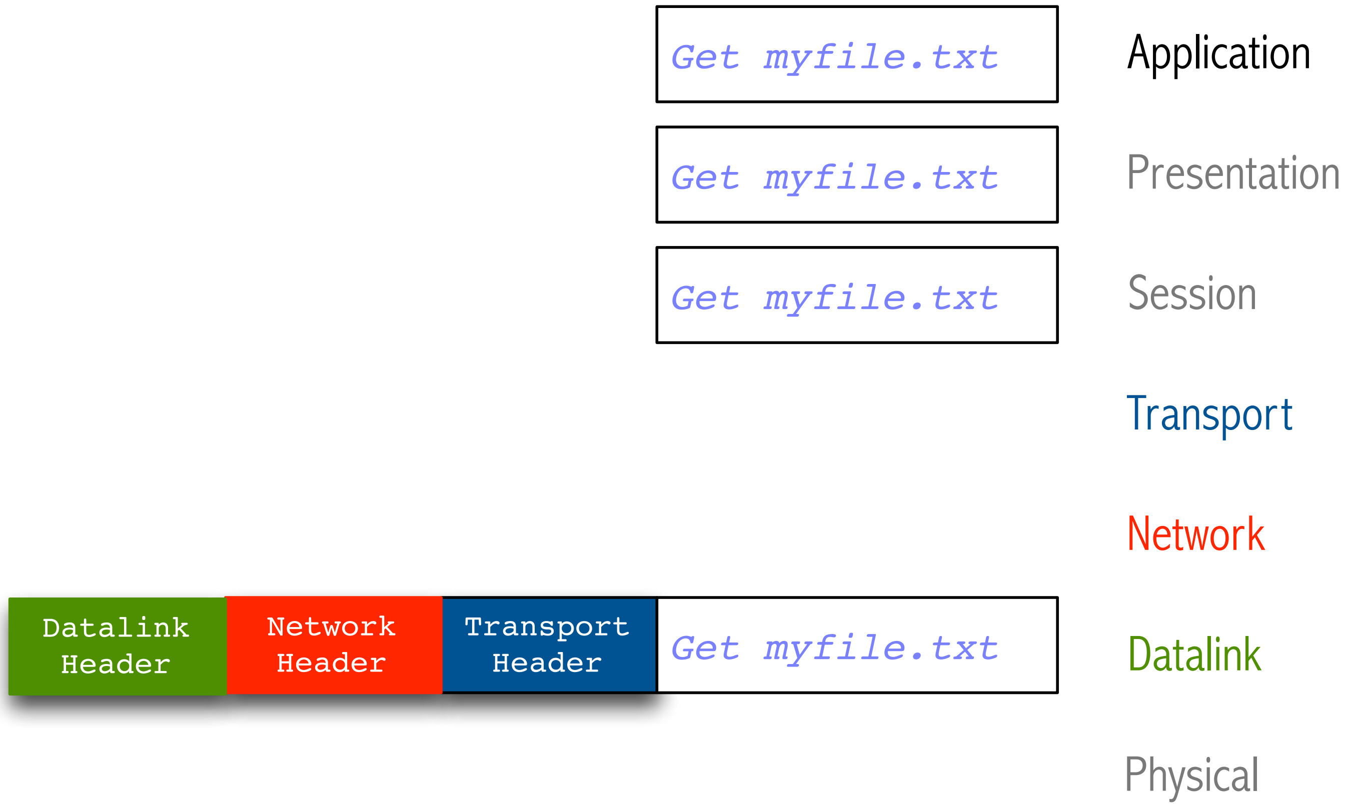
Transport

Network

Datalink Header | Network Header | Transport Header | *Get myfile.txt* Datalink

Physical

Encapsulation at layer 2



Get myfile.txt — Application

Get myfile.txt — Presentation

Get myfile.txt — Session

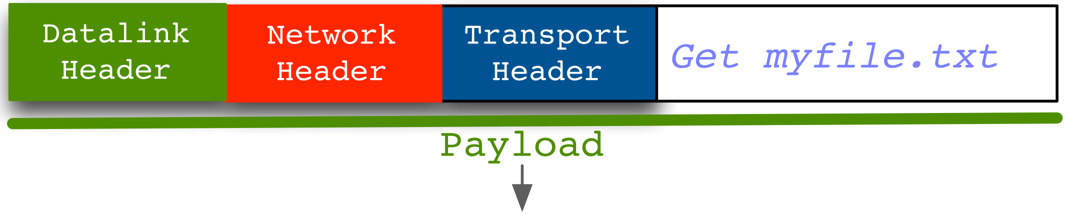
Transport

Network

Datalink Header | Network Header | Transport Header | Get myfile.txt — Datalink

Physical

Encapsulation at layer 2



Application

Presentation

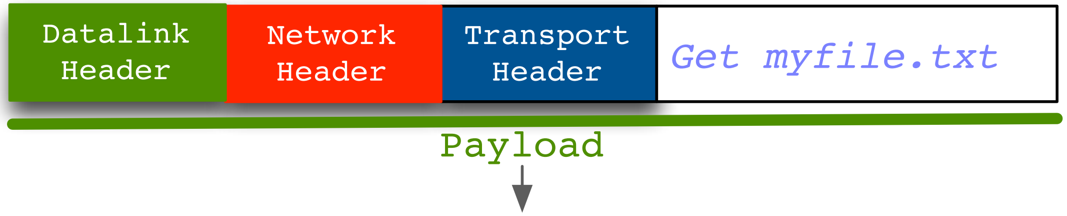
Session

Transport

Network

Datalink

Physical



Application

Presentation

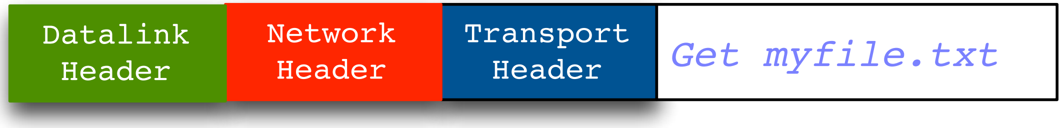
Session

Transport

Network

Datalink

Physical



Datalink Header

Network Header

Transport Header

Get myfile.txt

Application

Presentation

Session

Transport

Network

Datalink

Physical

Application

Presentation

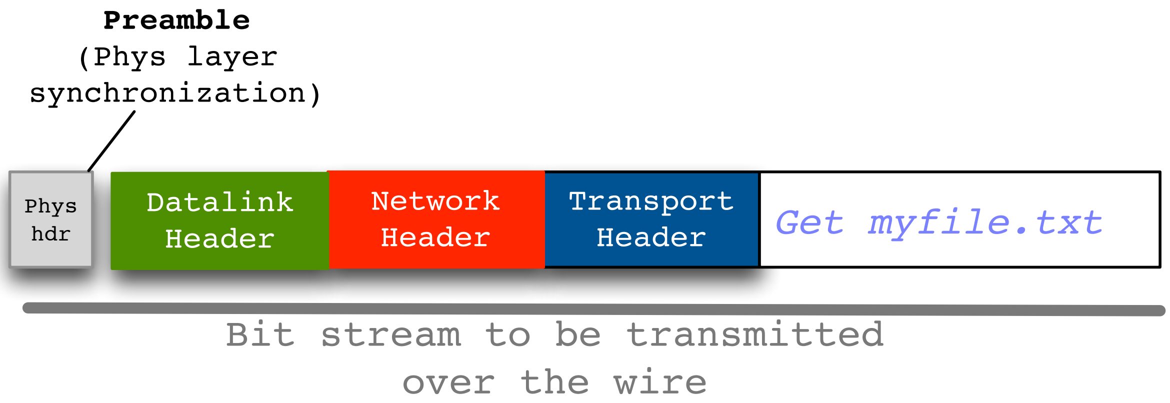
Session

Transport

Network

Datalink

Physical



Preamble
(Phys layer
synchronization)

Phys
hdr

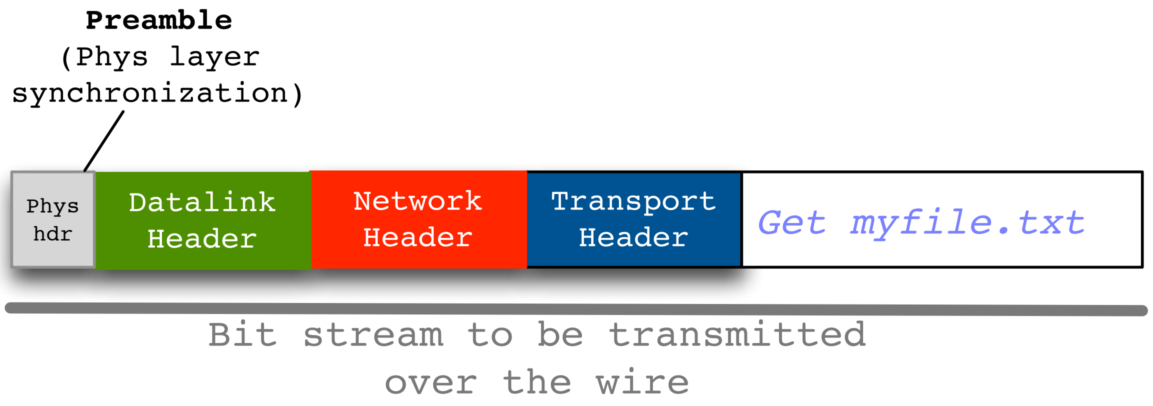
Datalink
Header

Network
Header

Transport
Header

Get myfile.txt

Bit stream to be transmitted
over the wire



Preamble
(Phys layer
synchronization)

Bit stream to be transmitted
over the wire

Application

Presentation

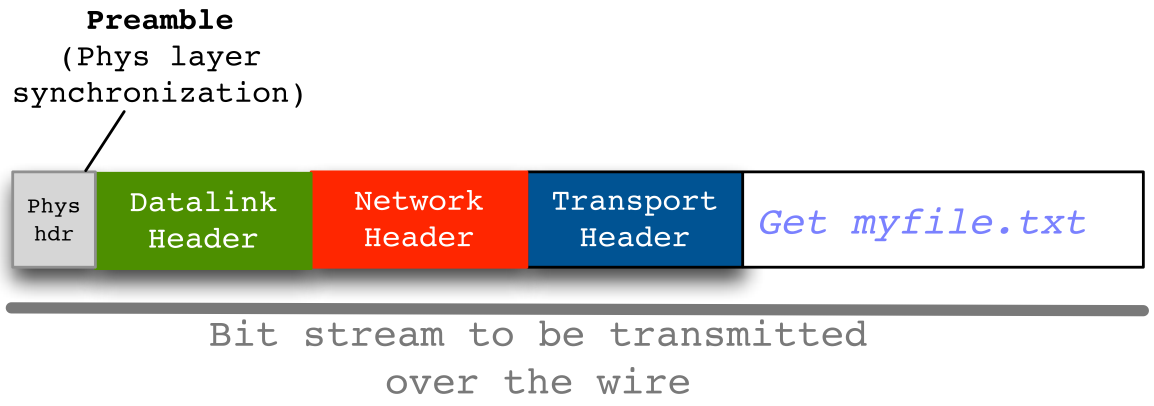
Session

Transport

Network

Datalink

Physical



Preamble
(Phys layer
synchronization)

Phys hdr

Datalink Header

Network Header

Transport Header

Get myfile.txt

Bit stream to be transmitted
over the wire

Application

Presentation

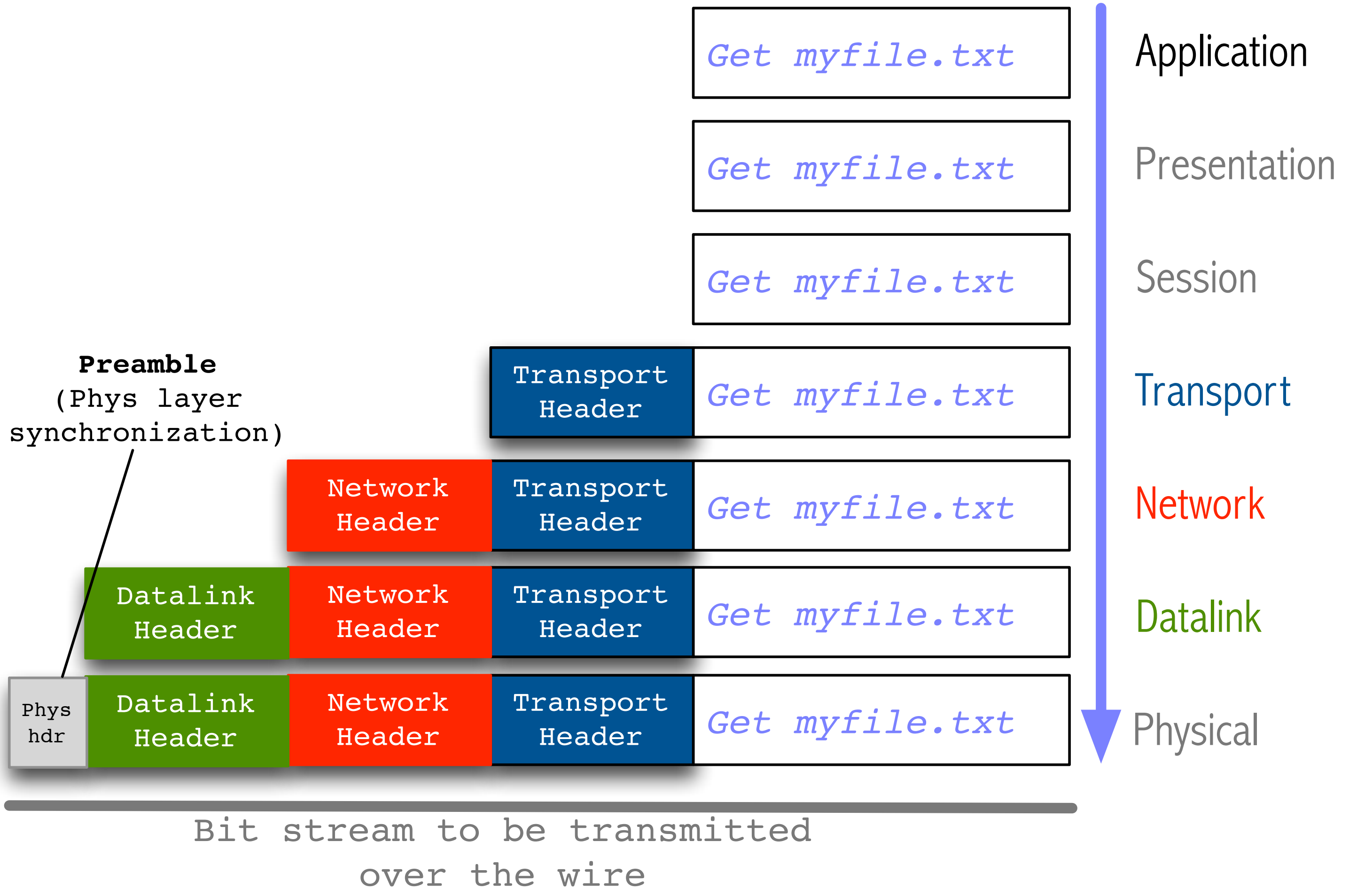
Session

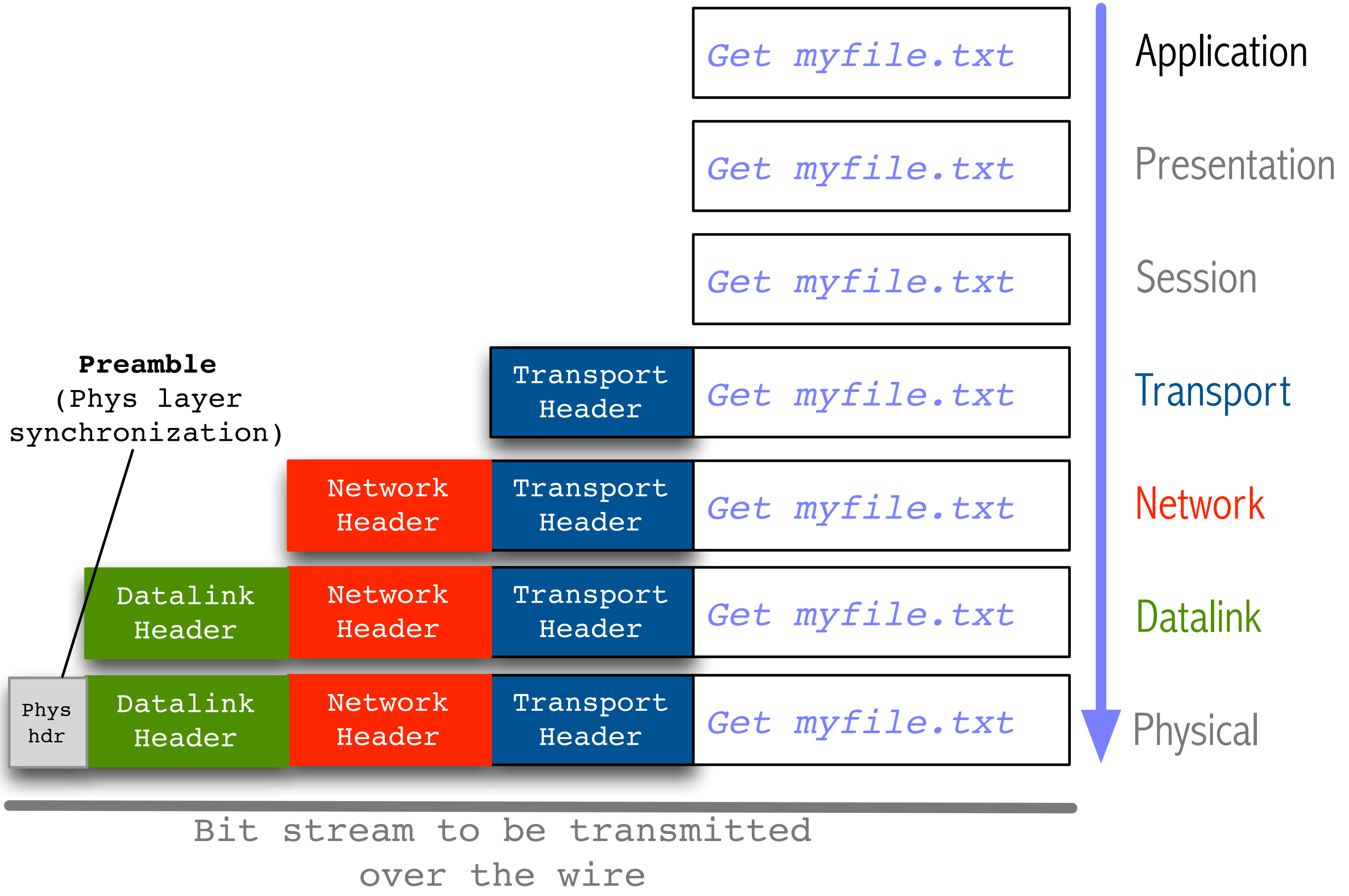
Transport

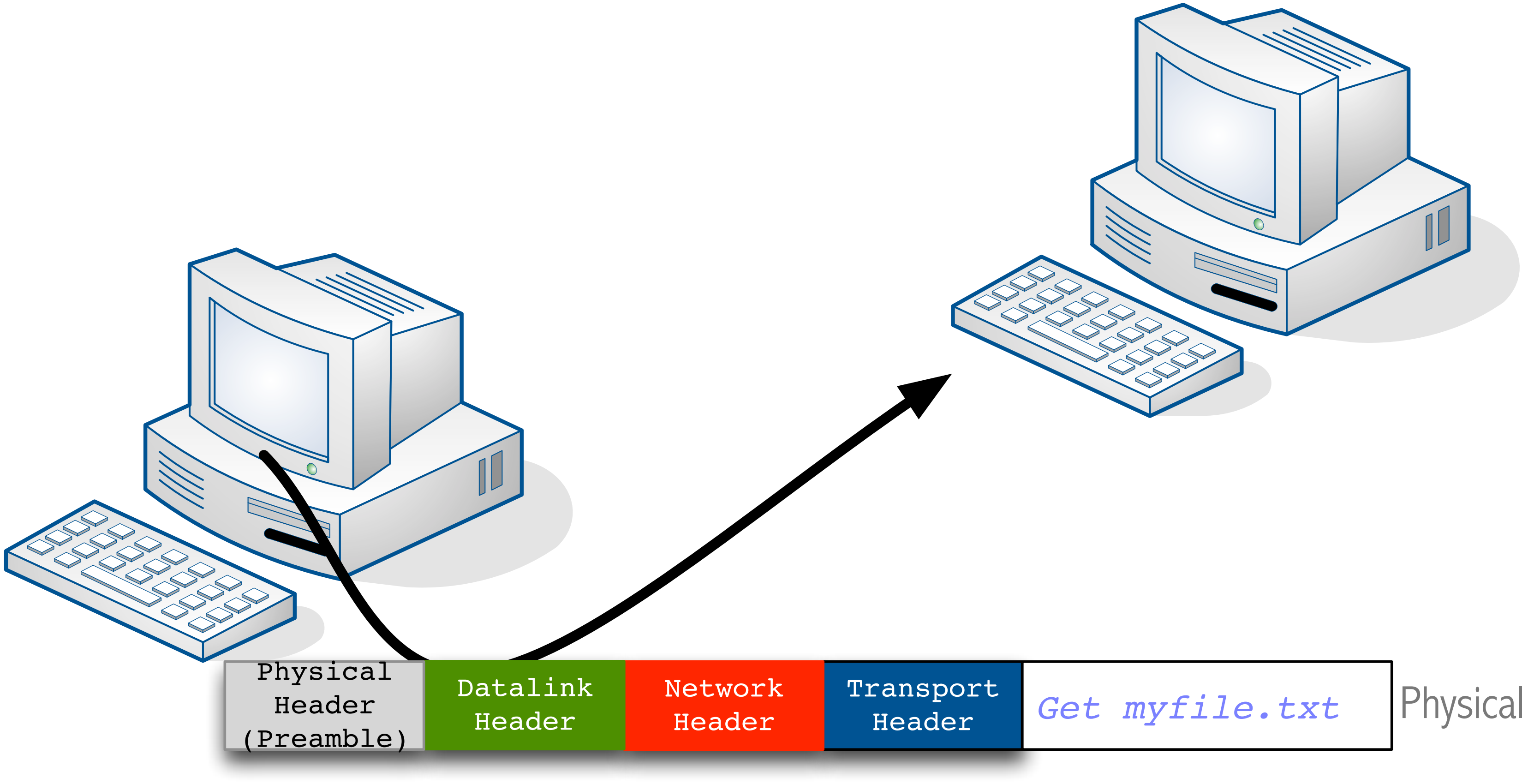
Network

Datalink

Physical


Get myfile.txt
Application
Get myfile.txt
Presentation
Get myfile.txt
Session
Transport Header
Get myfile.txt
Transport
Network Header
Transport Header
Get myfile.txt
Network
Datalink Header
Network Header
Transport Header
Get myfile.txt
Datalink
Phys hdr
Datalink Header
Network Header
Transport Header
Get myfile.txt
Physical
Preamble
(Phys layer synchronization)
Bit stream to be transmitted over the wire


Application
Presentation
Session
Transport
Network
Datalink
Physical
Get myfile.txt
Transport Header
Network Header
Datalink Header
Phys hdr
Preamble
(Phys layer synchronization)
Bit stream to be transmitted over the wire



Physical Header (Preamble)

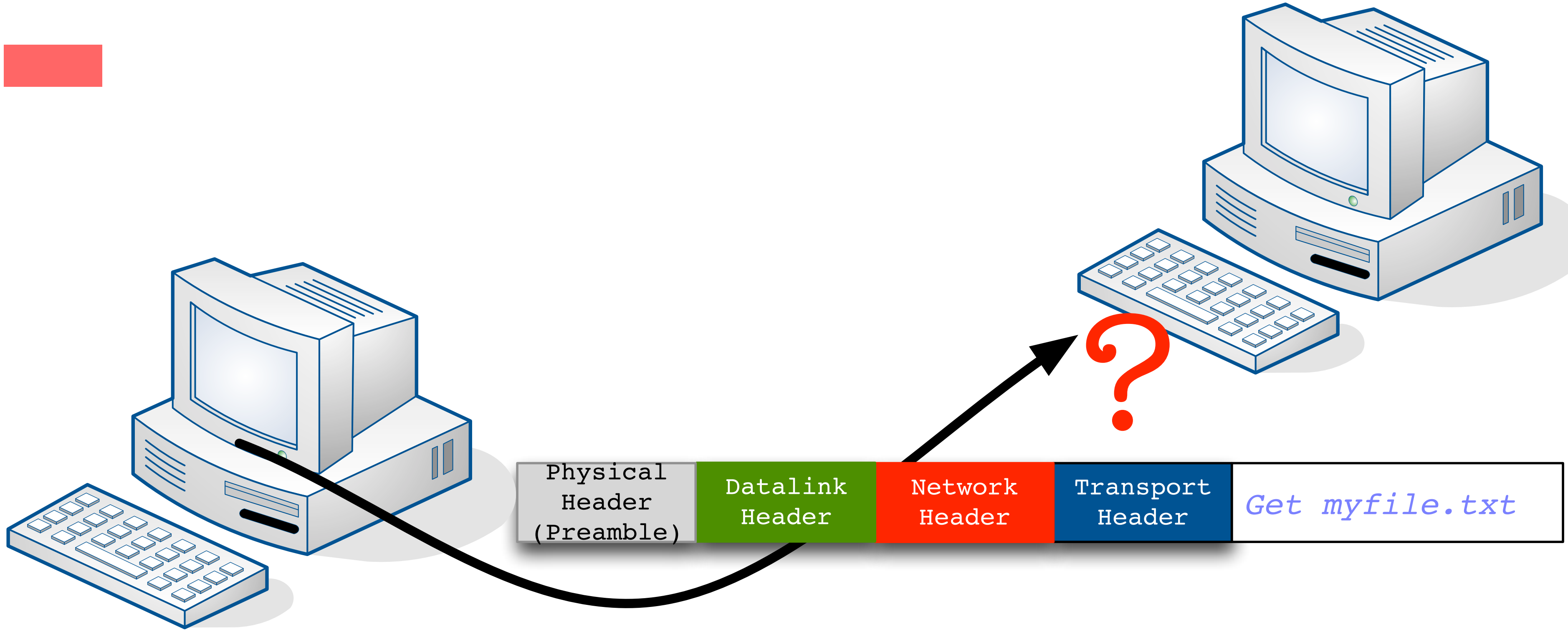
Datalink Header

Network Header

Transport Header

Get myfile.txt

Physical



Physical Header (Preamble)

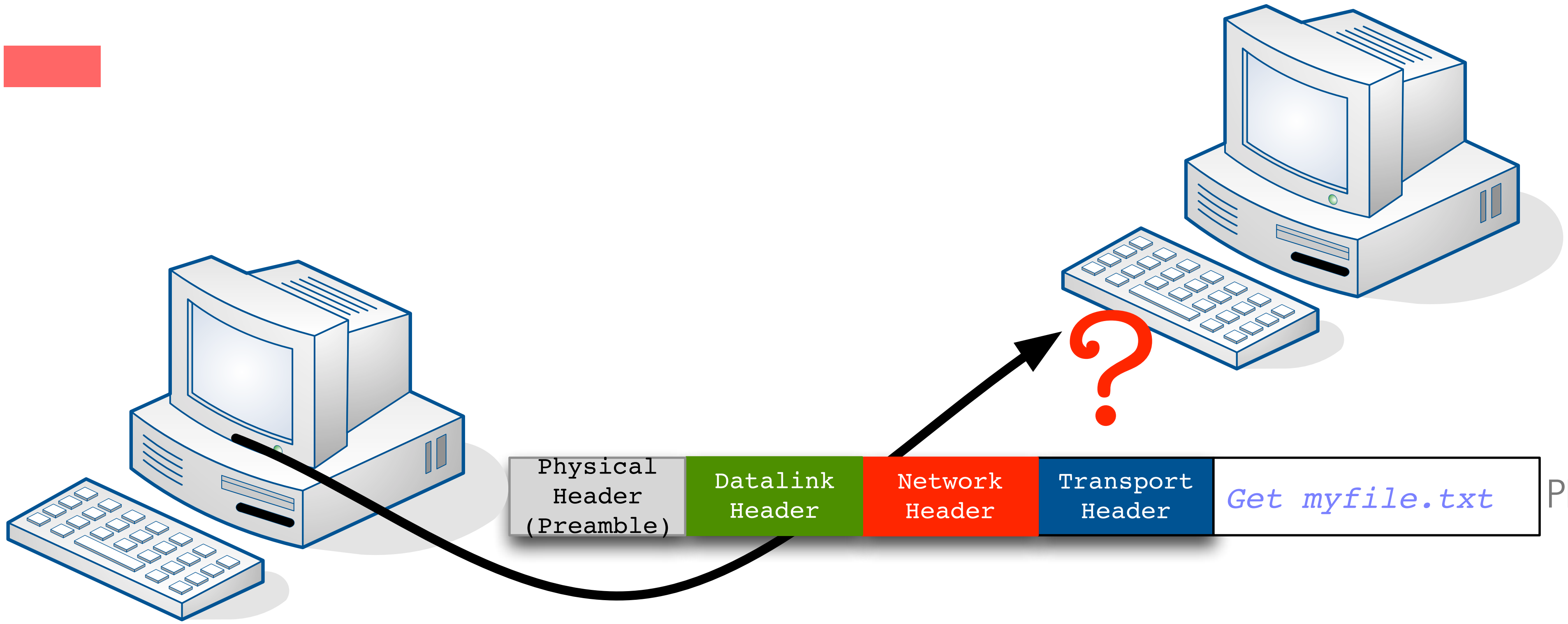
Datalink Header

Network Header

Transport Header

Get myfile.txt

Physical



Physical Header (Preamble)

Datalink Header

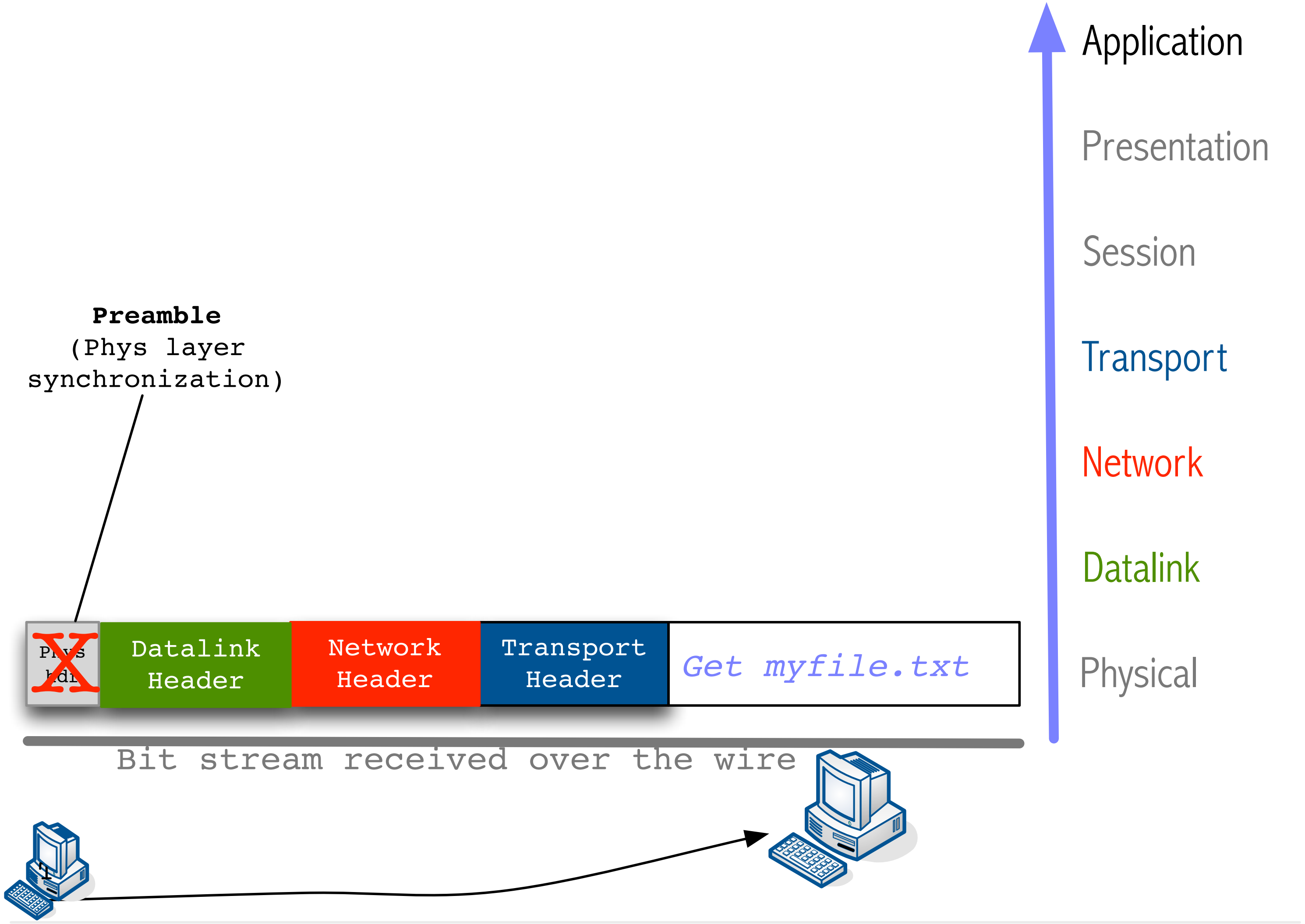
Network Header

Transport Header

Get myfile.txt

Physical

De-encapsulation at layer 1 (Supress the preamble)



Preamble
(Phys layer
synchronization)

~~Phys hdr~~

Datalink Header

Network Header

Transport Header

Get myfile.txt

Bit stream received over the wire

Application

Presentation

Session

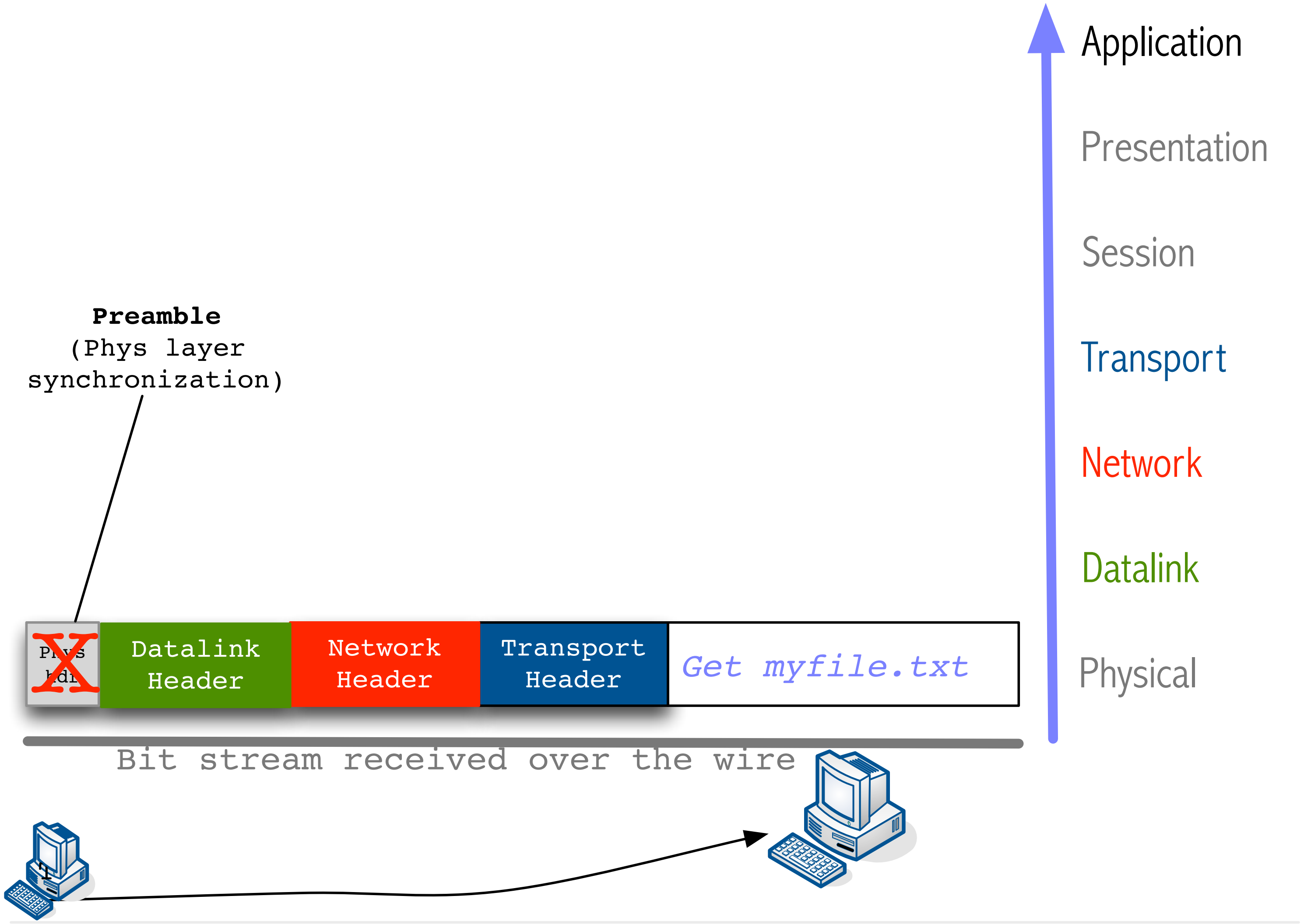
Transport

Network

Datalink

Physical

De-encapsulation at layer 1 (Supress the preamble)



Preamble
(Phys layer
synchronization)

~~Phys hdr~~

Datalink Header

Network Header

Transport Header

Get myfile.txt

Bit stream received over the wire

Application

Presentation

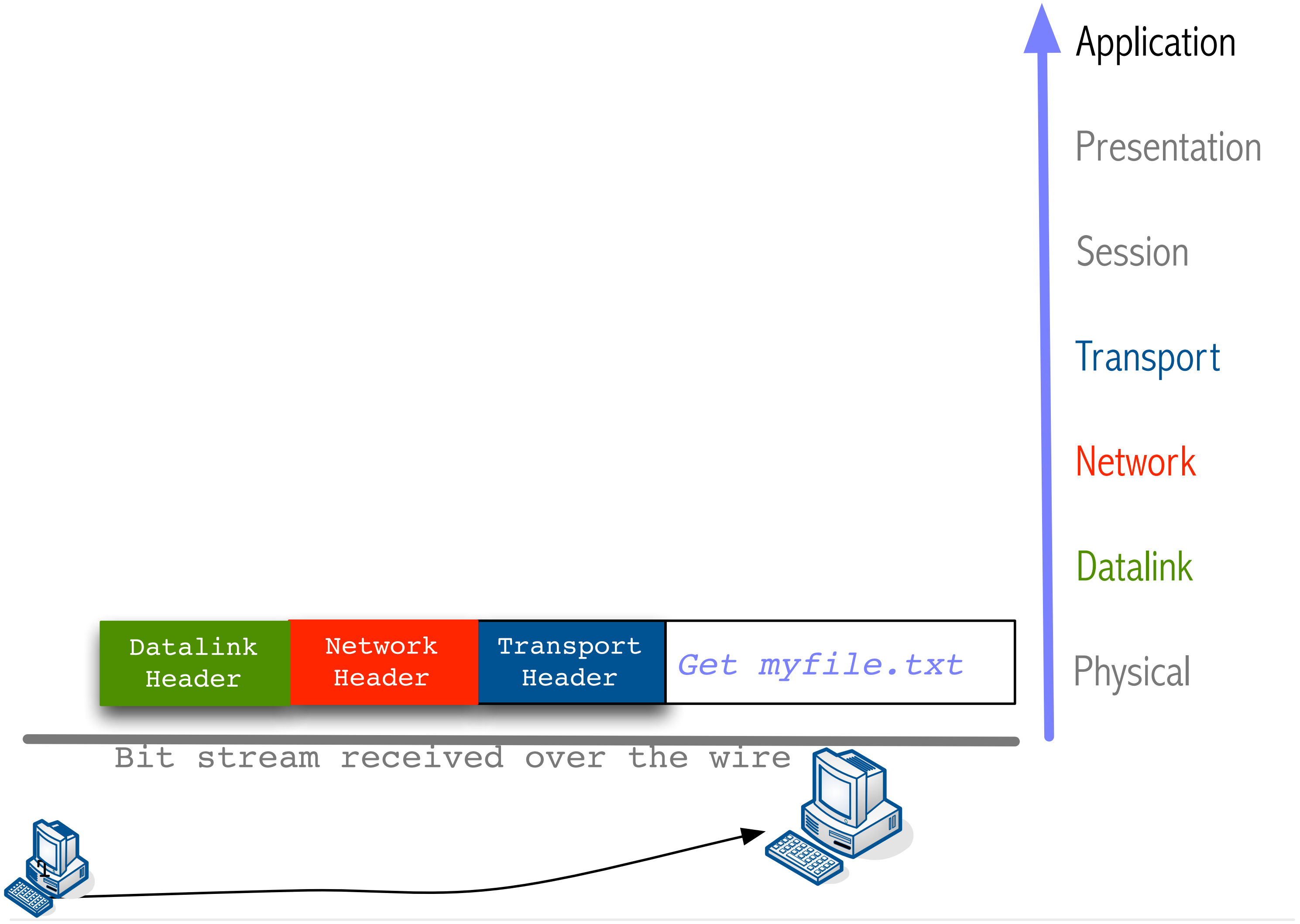
Session

Transport

Network

Datalink

Physical



Application

Presentation

Session

Transport

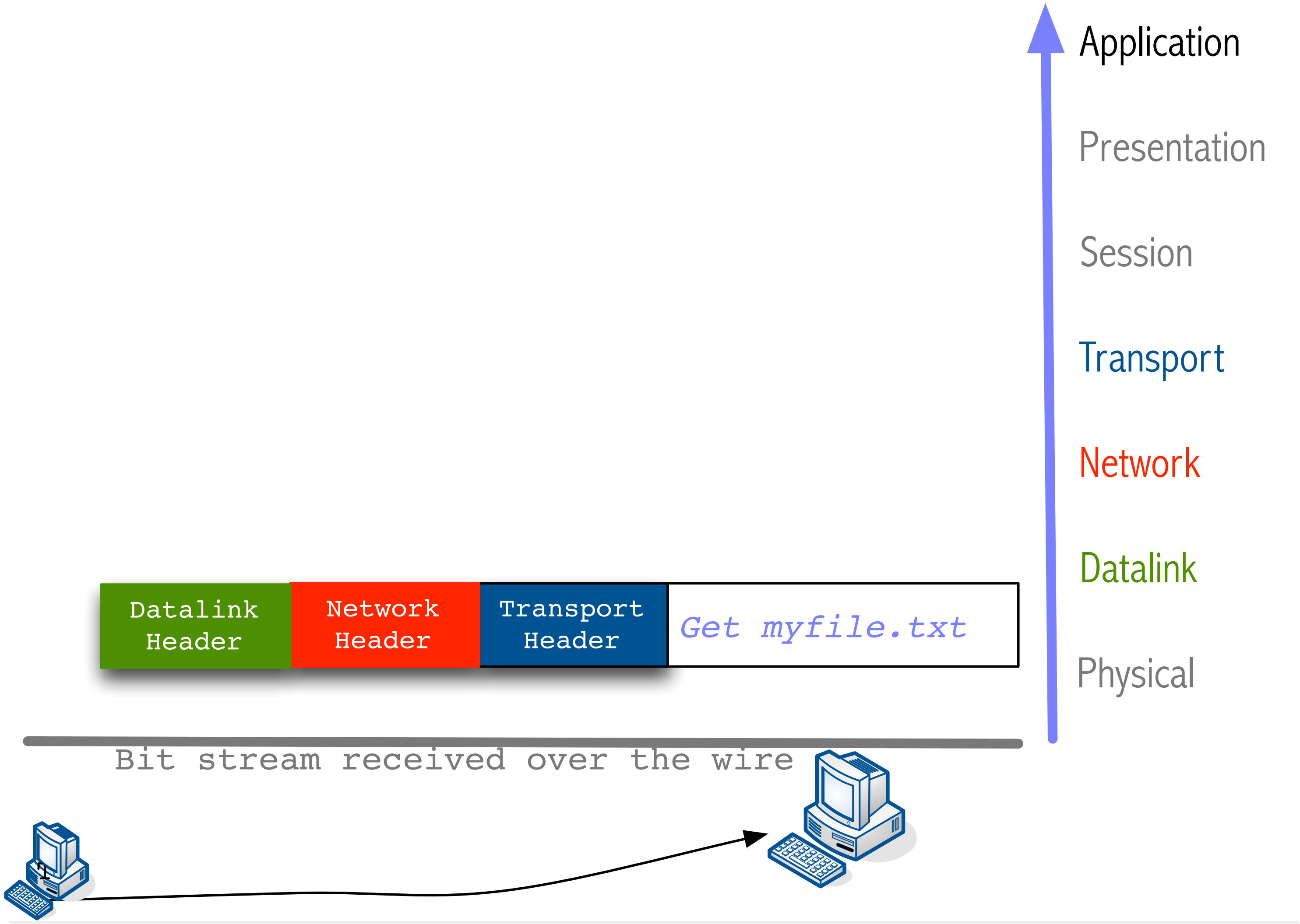
Network

Datalink

Physical

Datalink Header | Network Header | Transport Header | *Get myfile.txt*

Bit stream received over the wire



Application

Presentation

Session

Transport

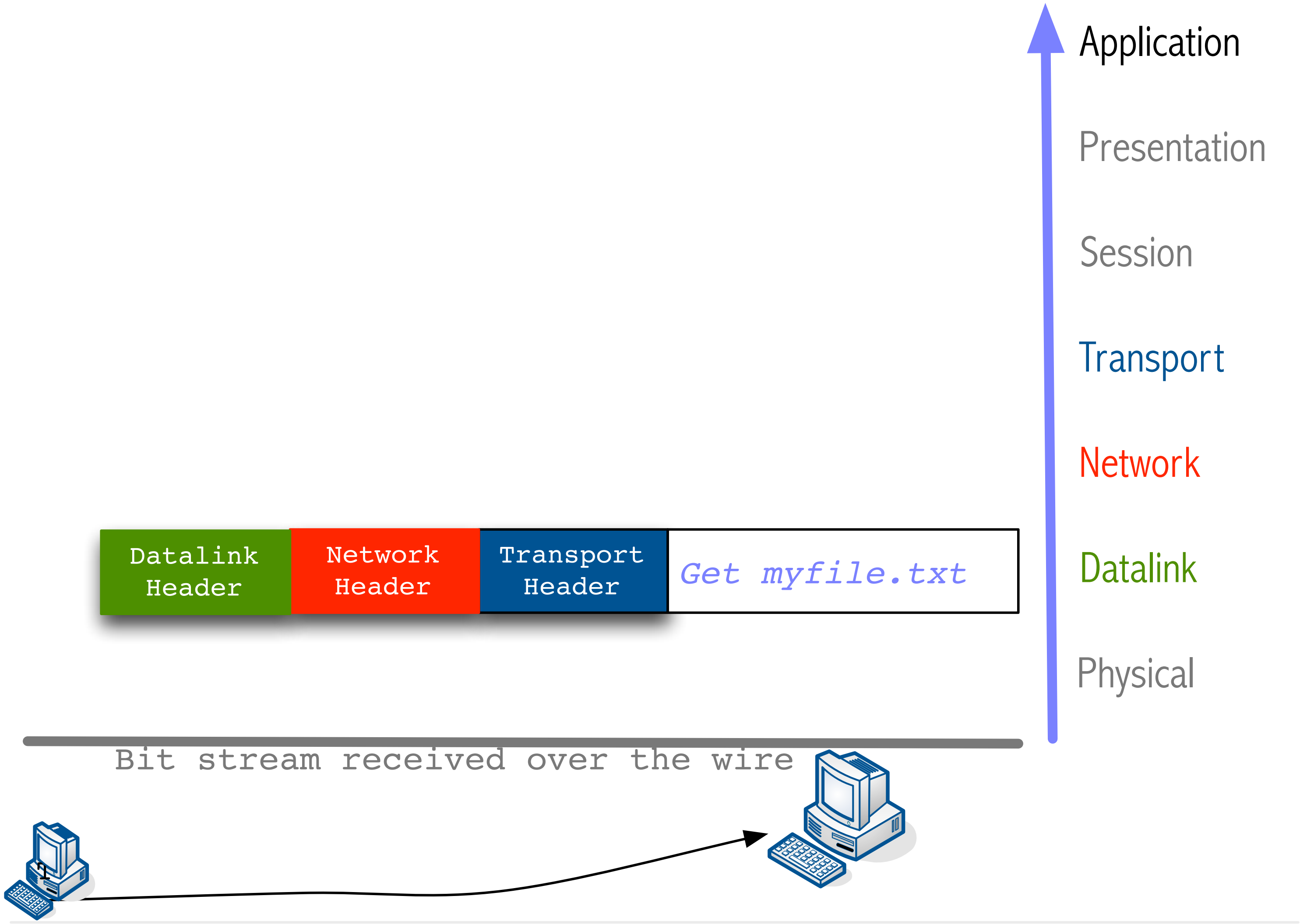
Network

Datalink

Physical

Datalink Header | Network Header | Transport Header | *Get myfile.txt*

Bit stream received over the wire



Application

Presentation

Session

Transport

Network

Datalink

Physical

Datalink Header

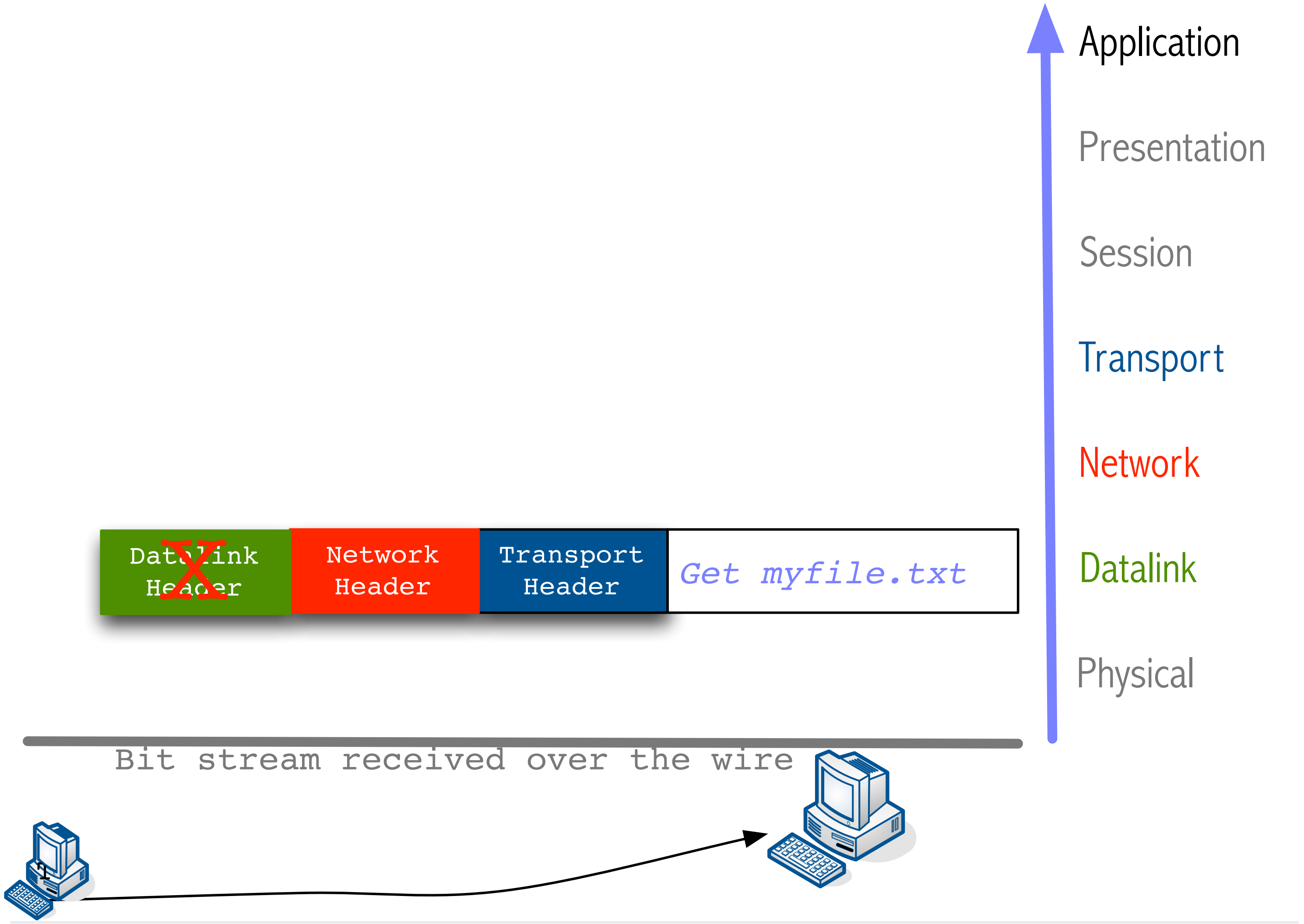
Network Header

Transport Header

Get myfile.txt

Bit stream received over the wire

De-encapsulation at layer 2



Application

Presentation

Session

Transport

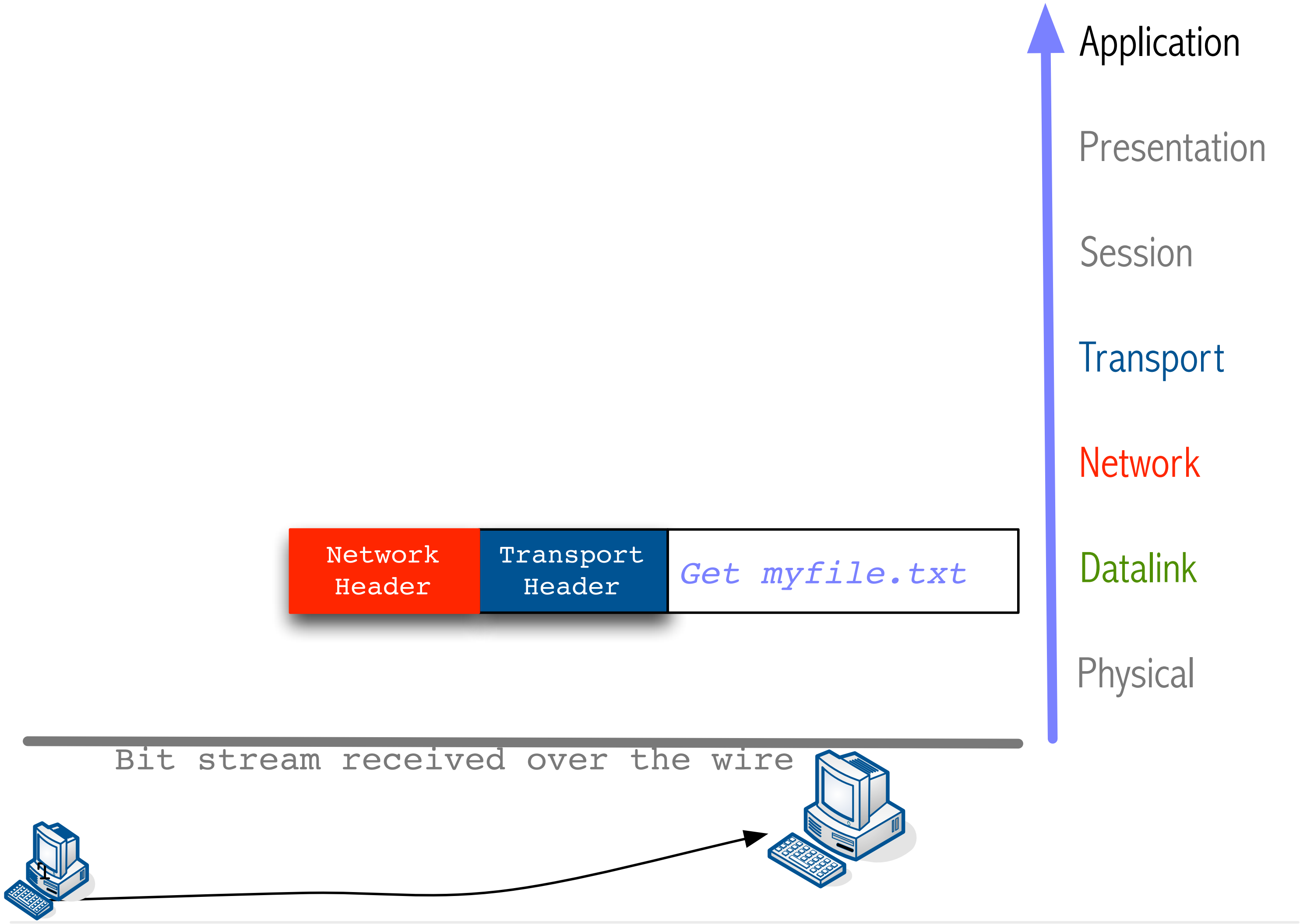
Network

Datalink

Physical

~~Datalink Header~~ | Network Header | Transport Header | *Get myfile.txt*

Bit stream received over the wire



Application

Presentation

Session

Transport

Network

Datalink

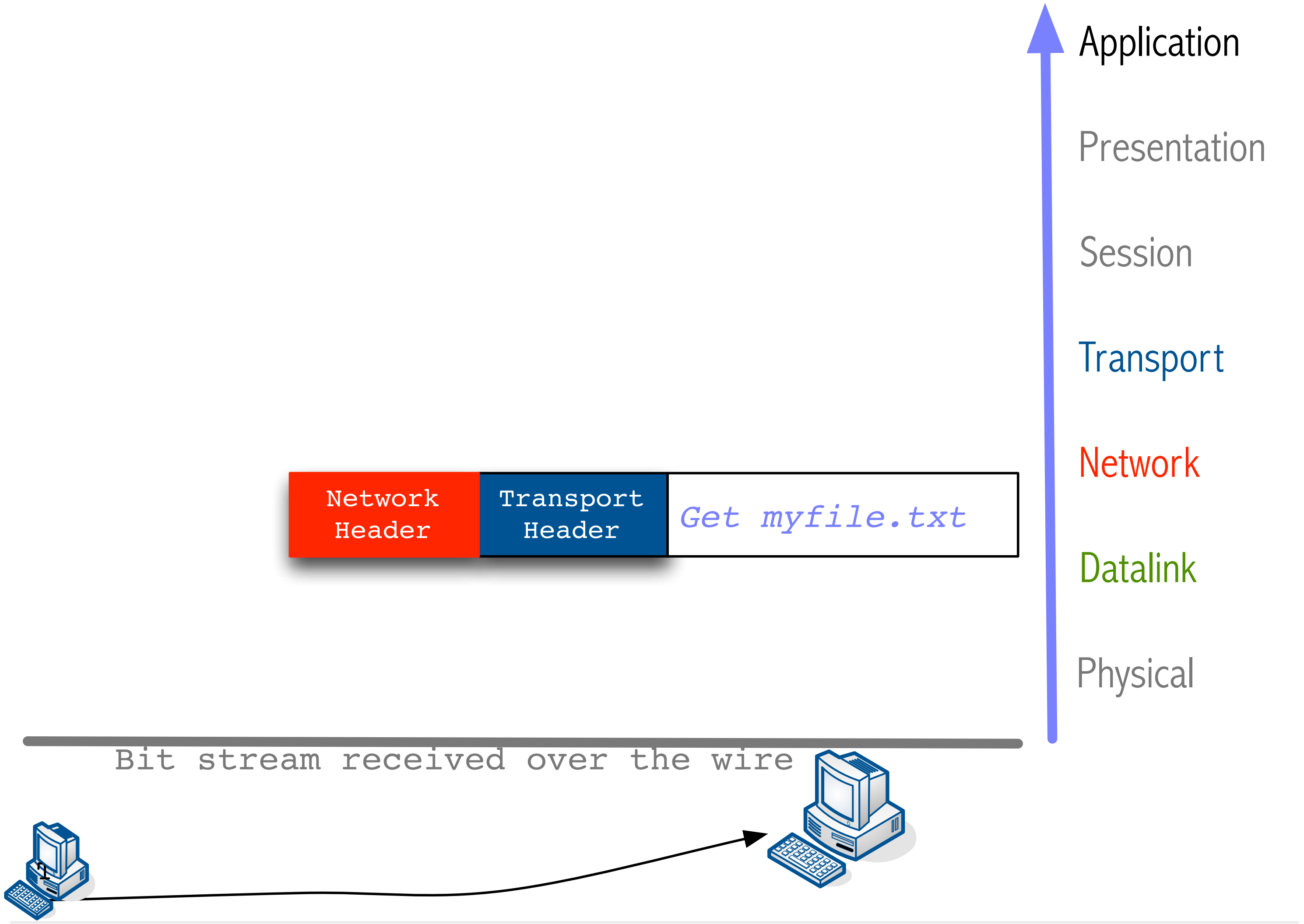
Physical

Network Header

Transport Header

Get myfile.txt

Bit stream received over the wire



Application

Presentation

Session

Transport

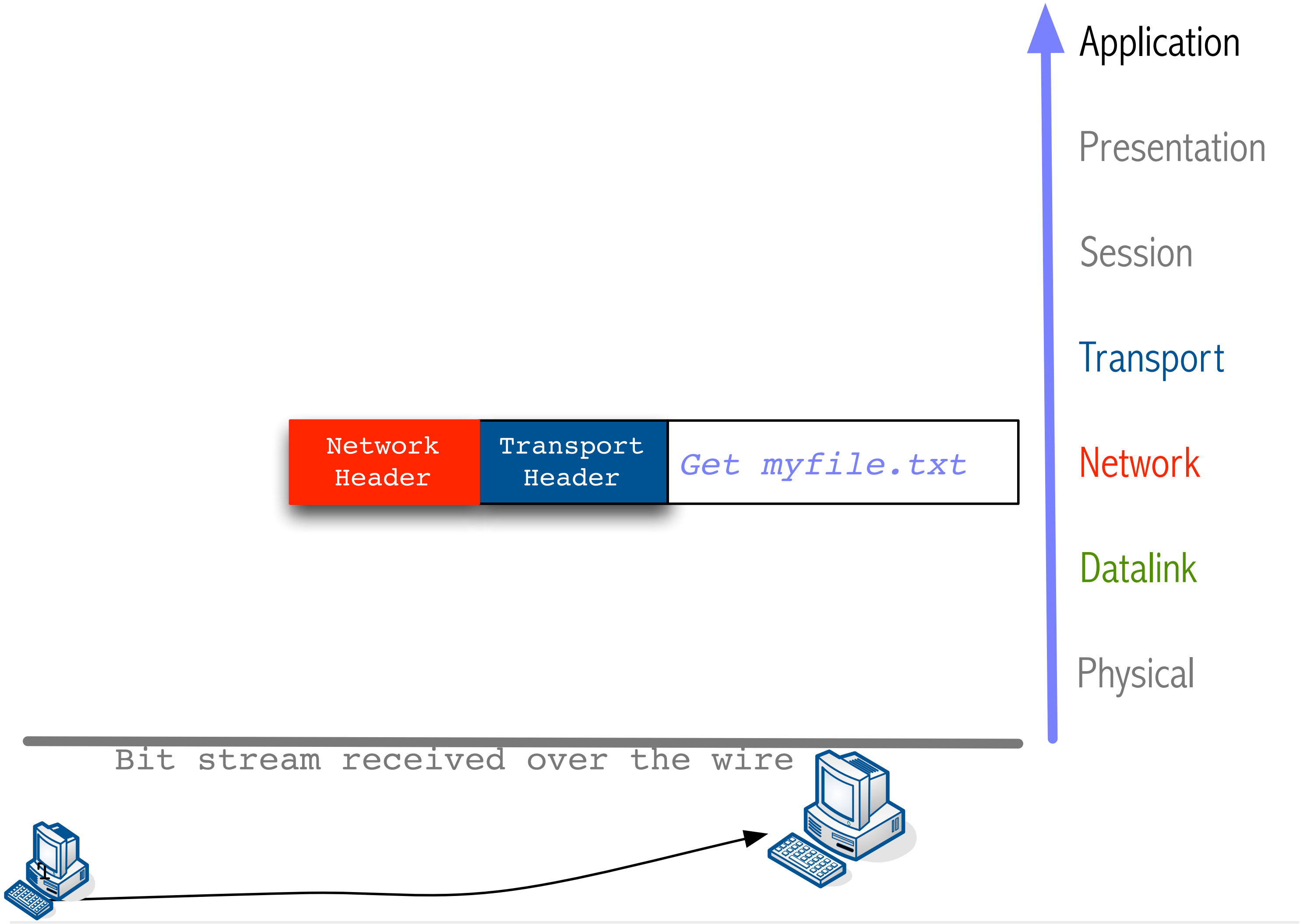
Network

Datalink

Physical

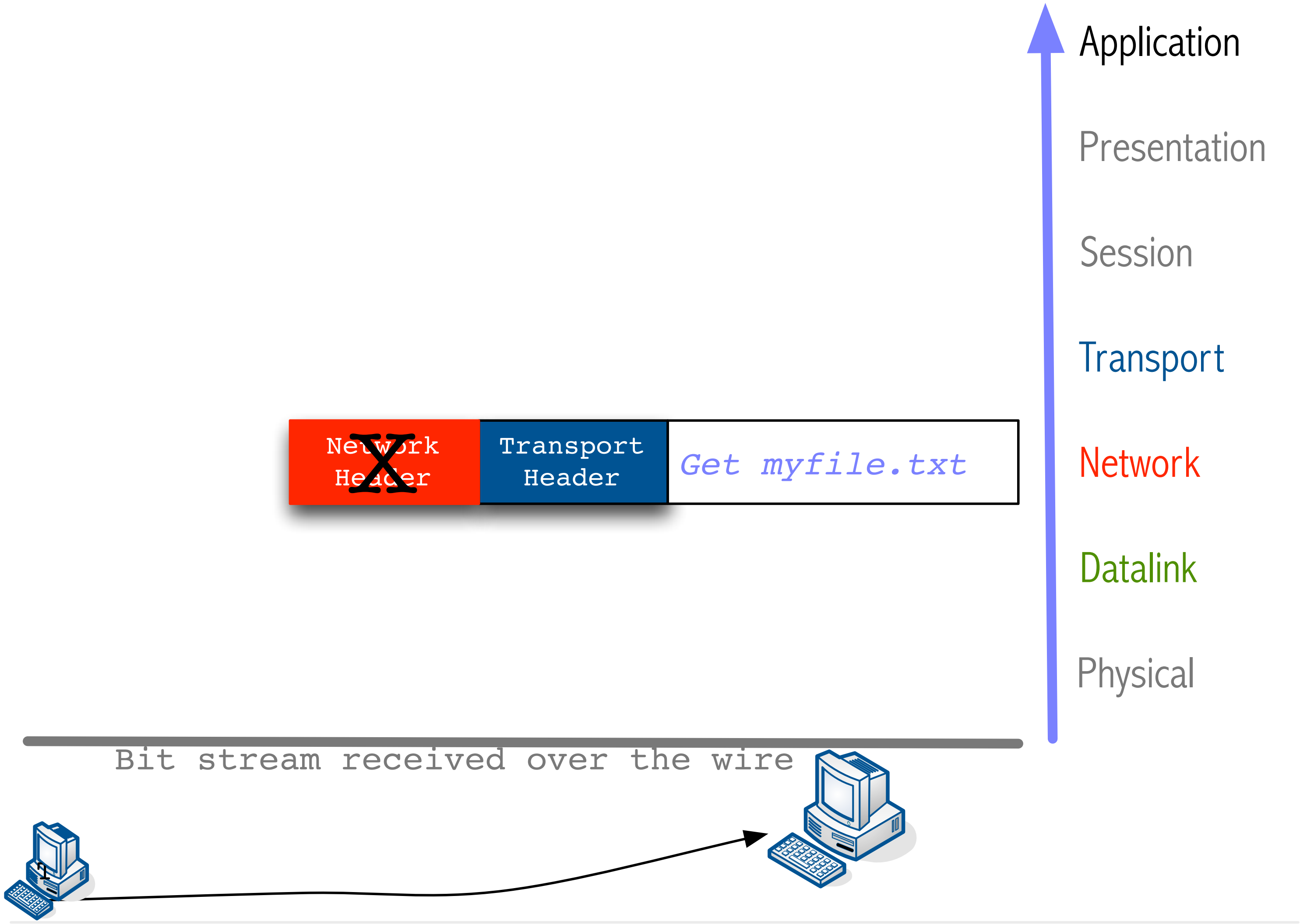
Network Header | Transport Header | *Get myfile.txt*

Bit stream received over the wire



Network Header
Transport Header
Get myfile.txt
Application
Presentation
Session
Transport
Network
Datalink
Physical
Bit stream received over the wire

De-encapsulation at layer 3



~~Network Header~~ | Transport Header | *Get myfile.txt*

Application

Presentation

Session

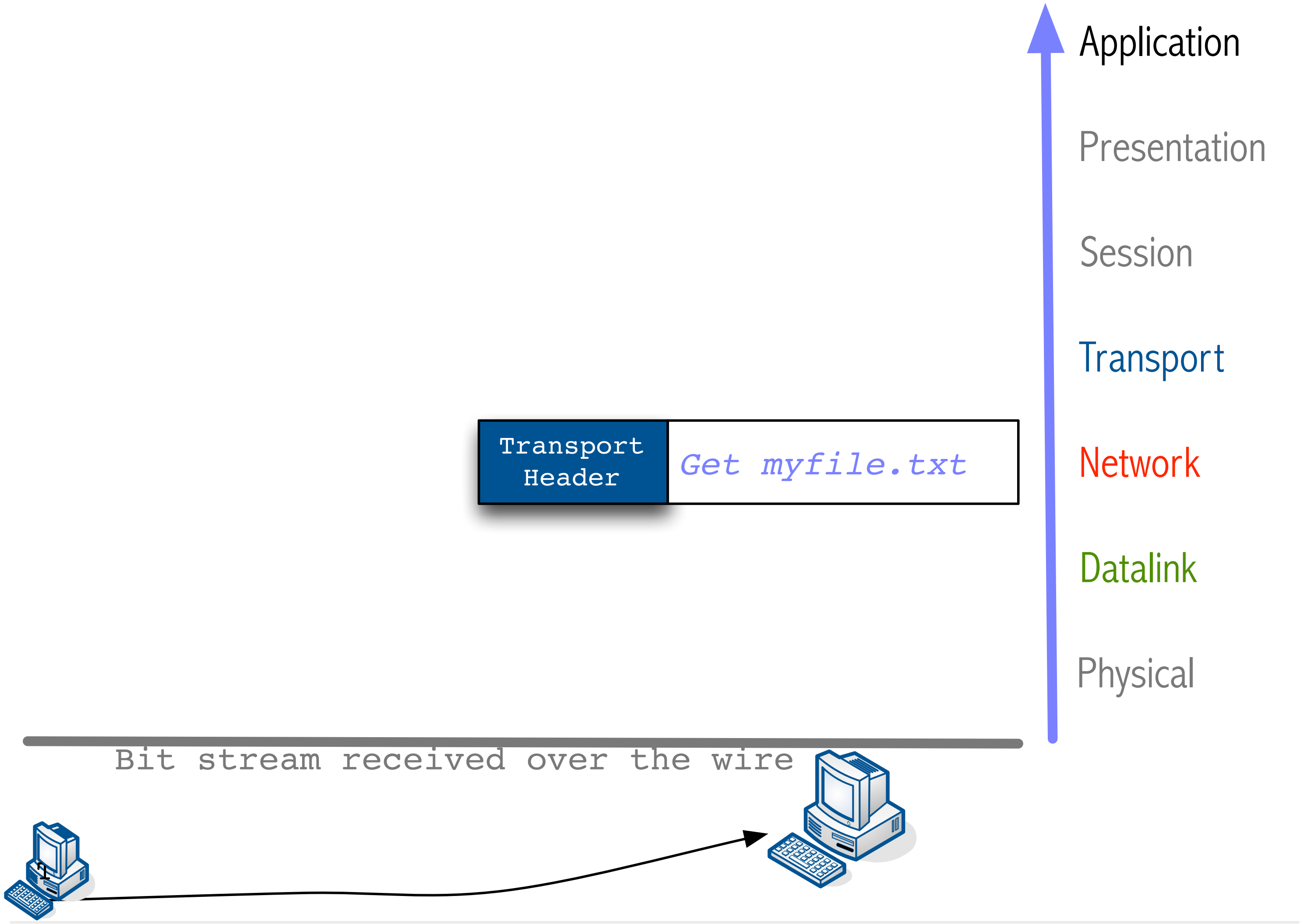
Transport

Network

Datalink

Physical

Bit stream received over the wire



Application

Presentation

Session

Transport

Network

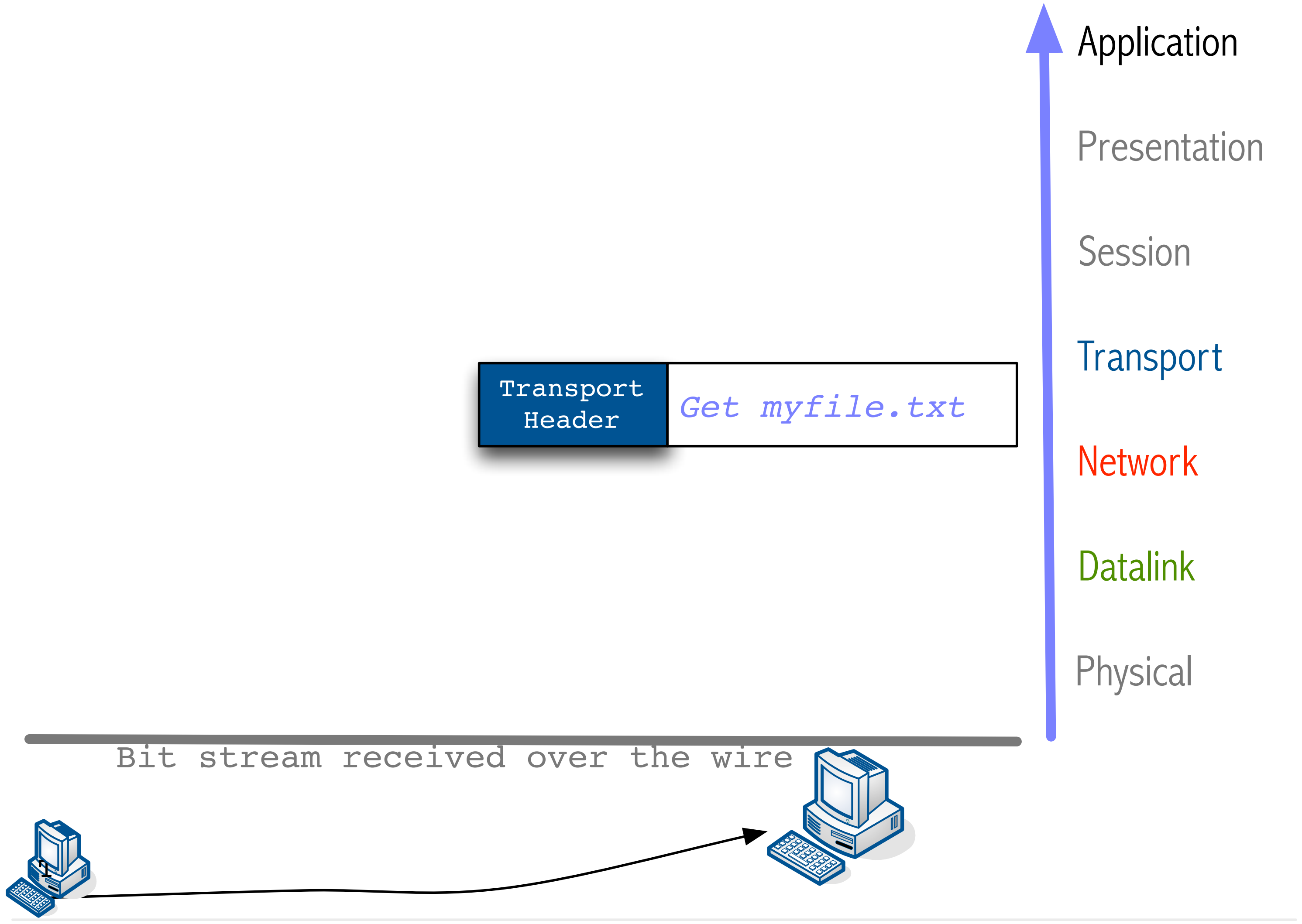
Datalink

Physical

Transport Header

Get myfile.txt

Bit stream received over the wire



Application

Presentation

Session

Transport

Network

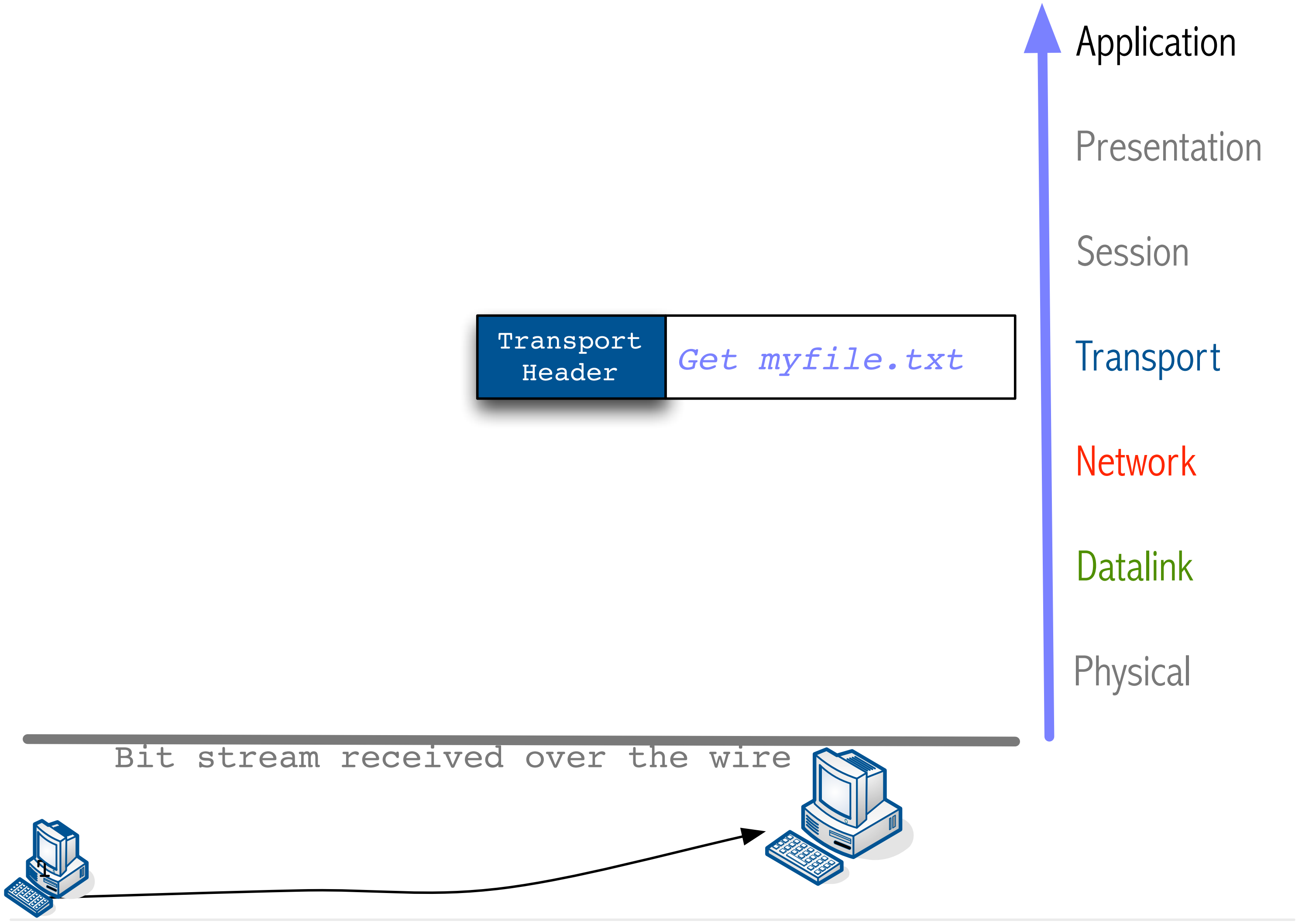
Datalink

Physical

Transport Header

Get myfile.txt

Bit stream received over the wire



Transport Header

Get myfile.txt

Application

Presentation

Session

Transport

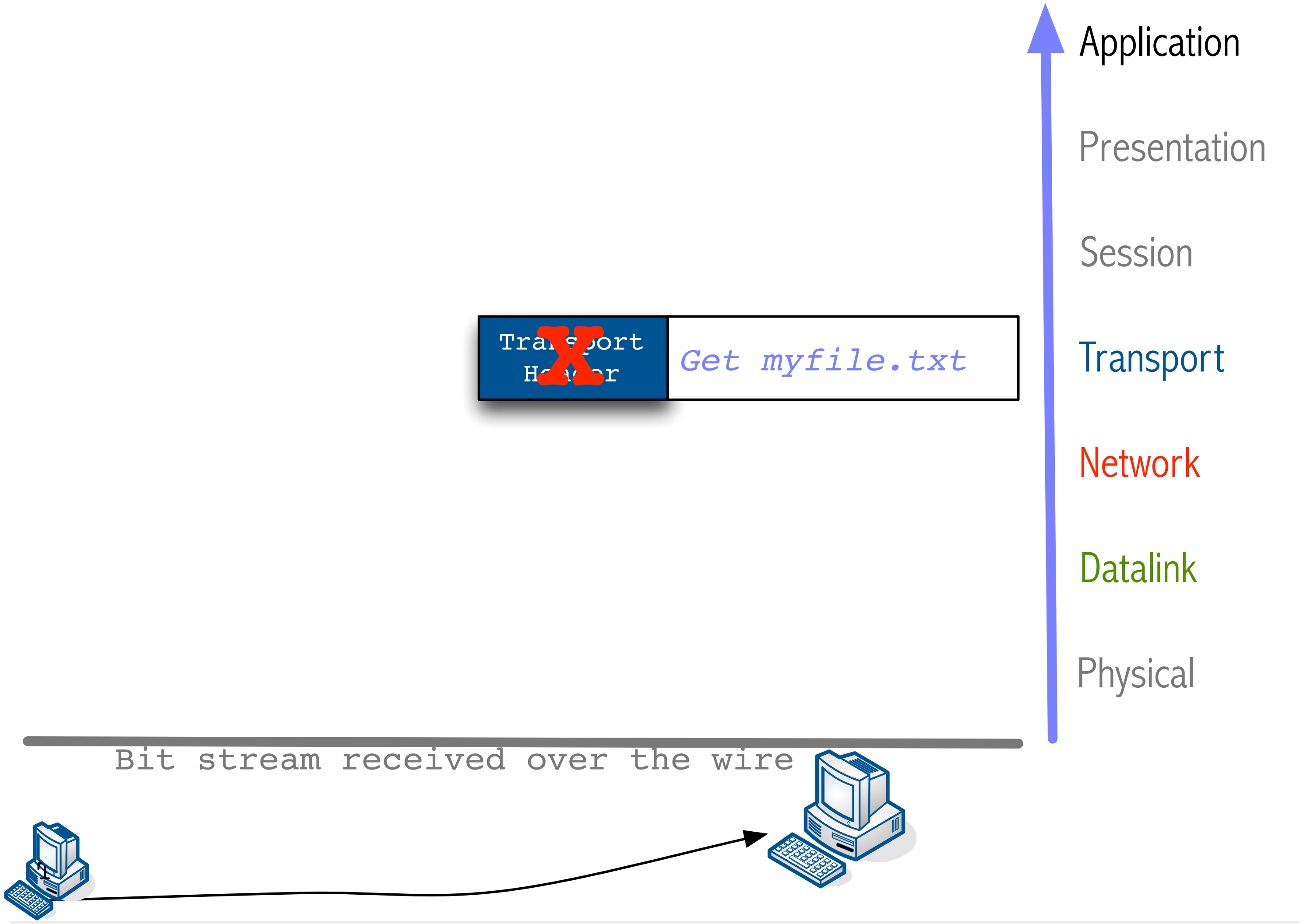
Network

Datalink

Physical

Bit stream received over the wire

De-encapsulation at layer 4



Application

Presentation

Session

Transport

Network

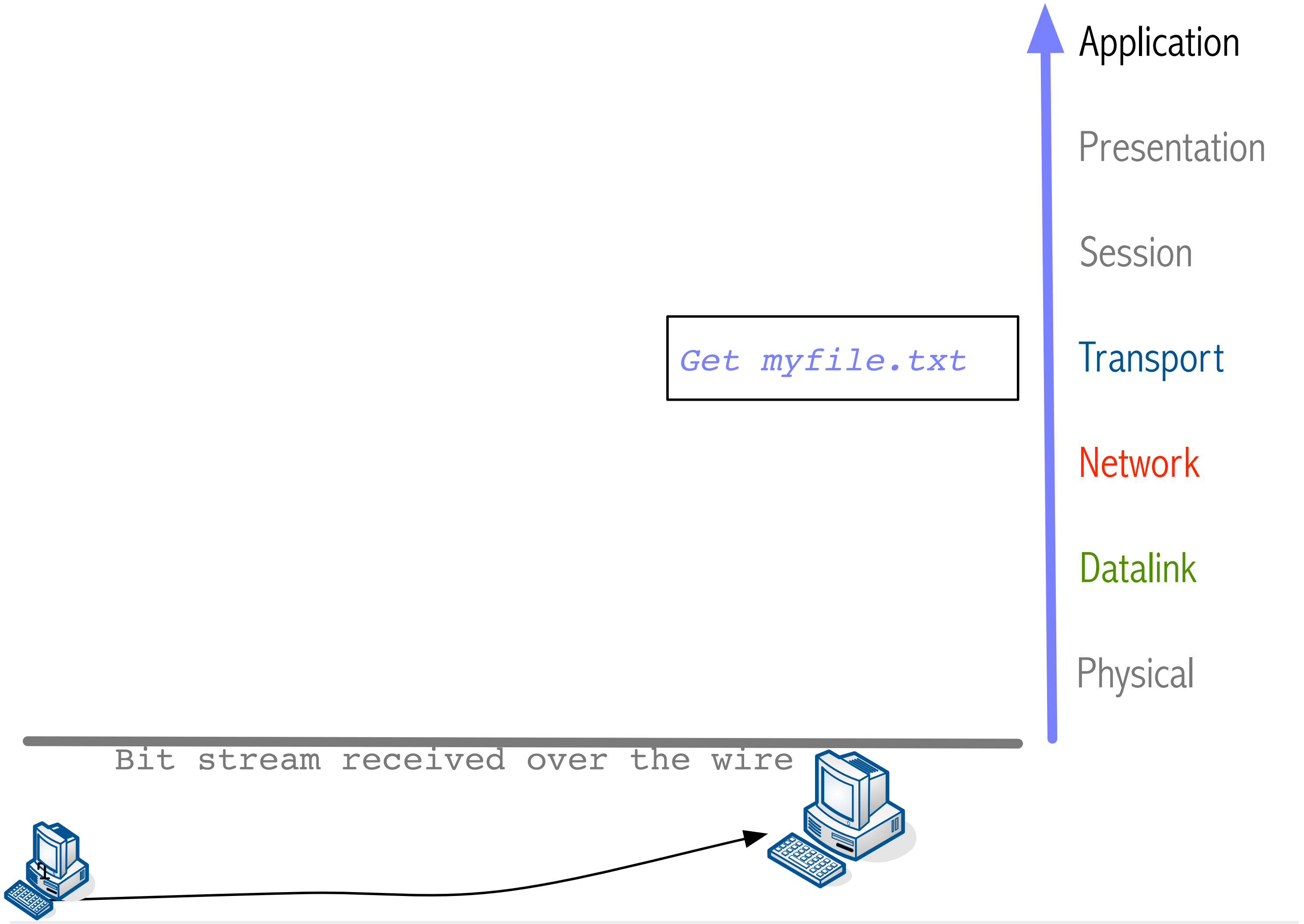
Datalink

Physical

Transport Header

Get myfile.txt

Bit stream received over the wire



Application

Presentation

Session

Transport

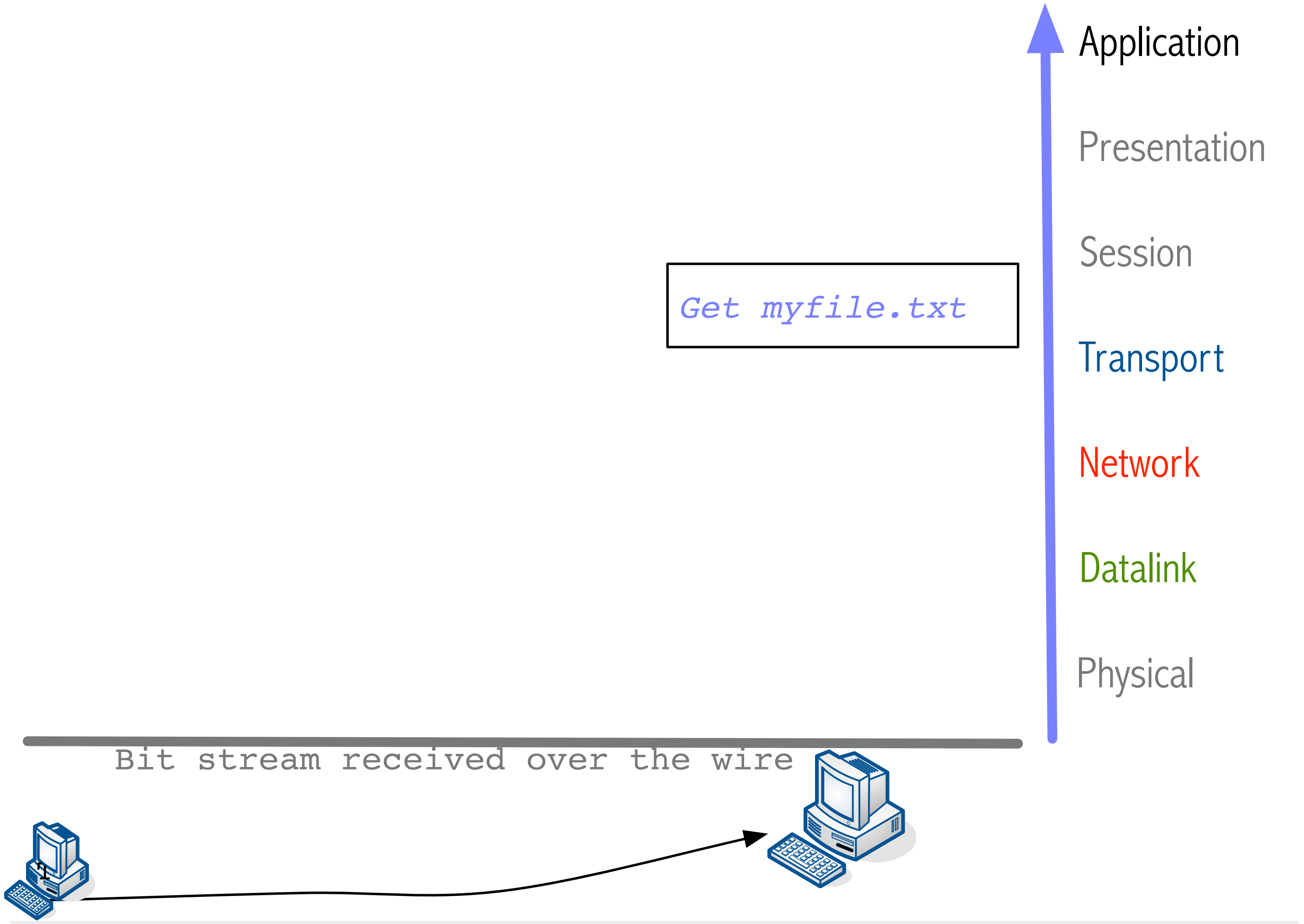
Network

Datalink

Physical

`Get myfile.txt`

`Bit stream received over the wire`



Application

Presentation

Session

Transport

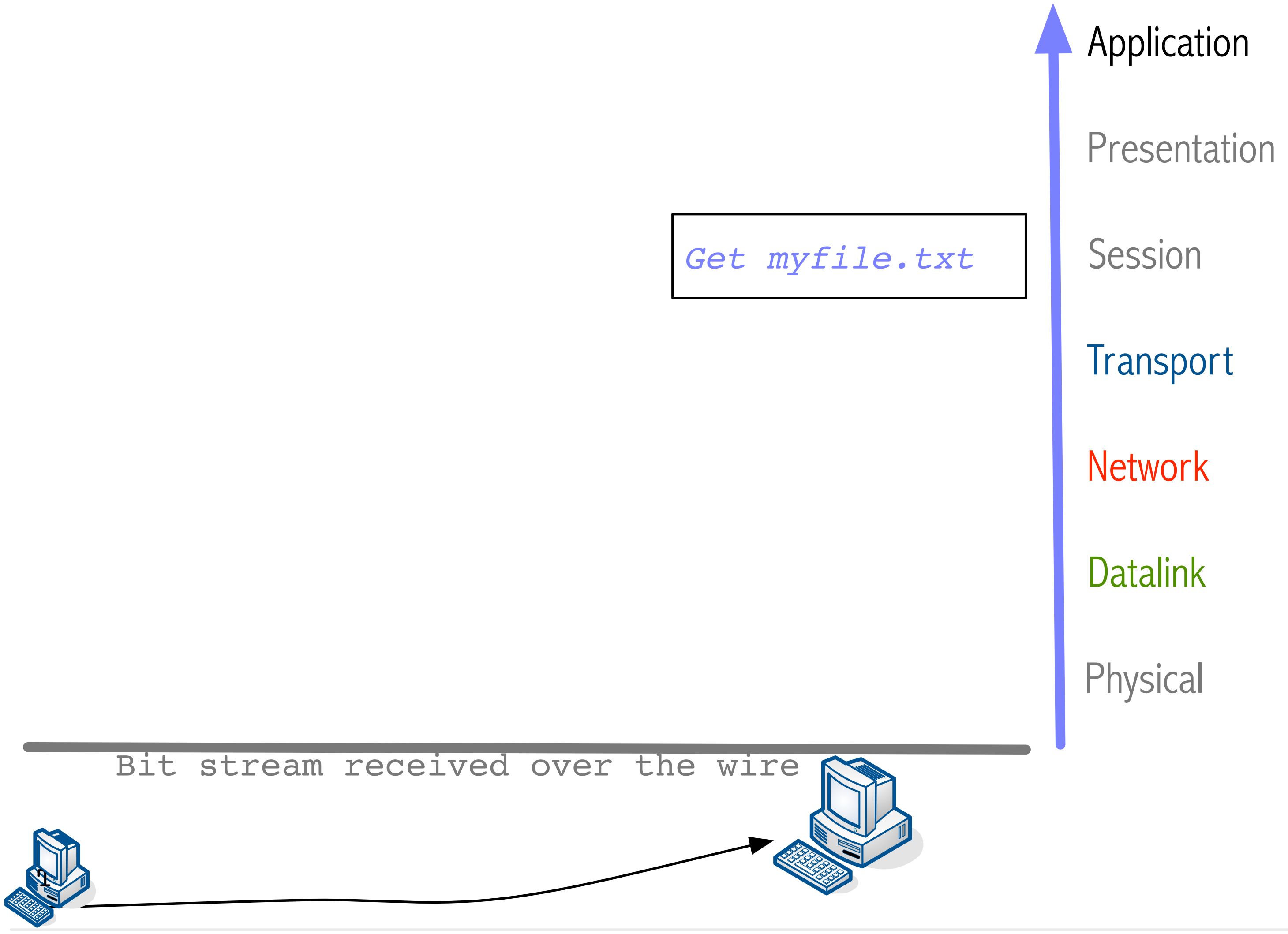
Network

Datalink

Physical

`Get myfile.txt`

`Bit stream received over the wire`



Get myfile.txt

Application

Presentation

Session

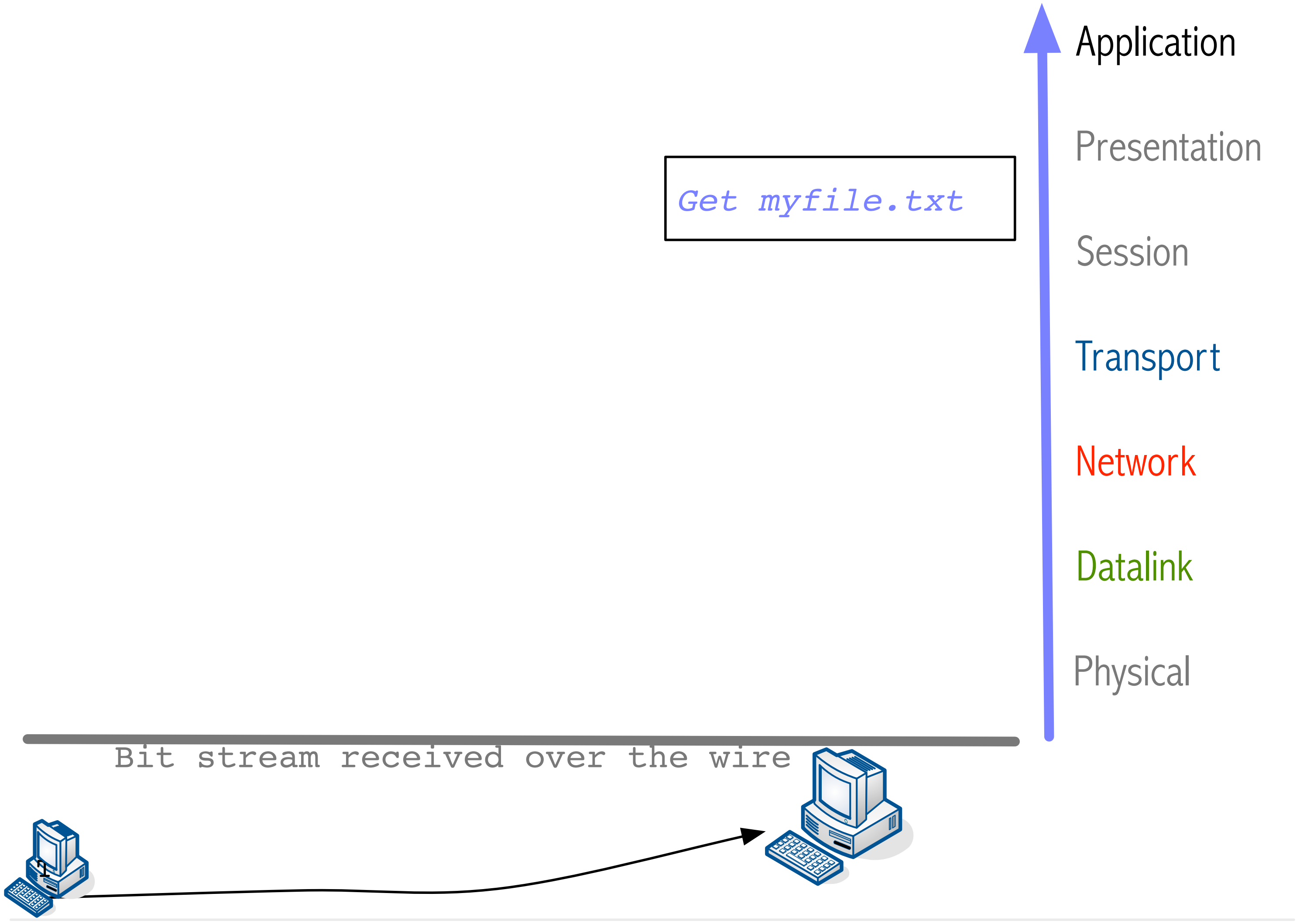
Transport

Network

Datalink

Physical

Bit stream received over the wire



Application

Presentation

Session

Transport

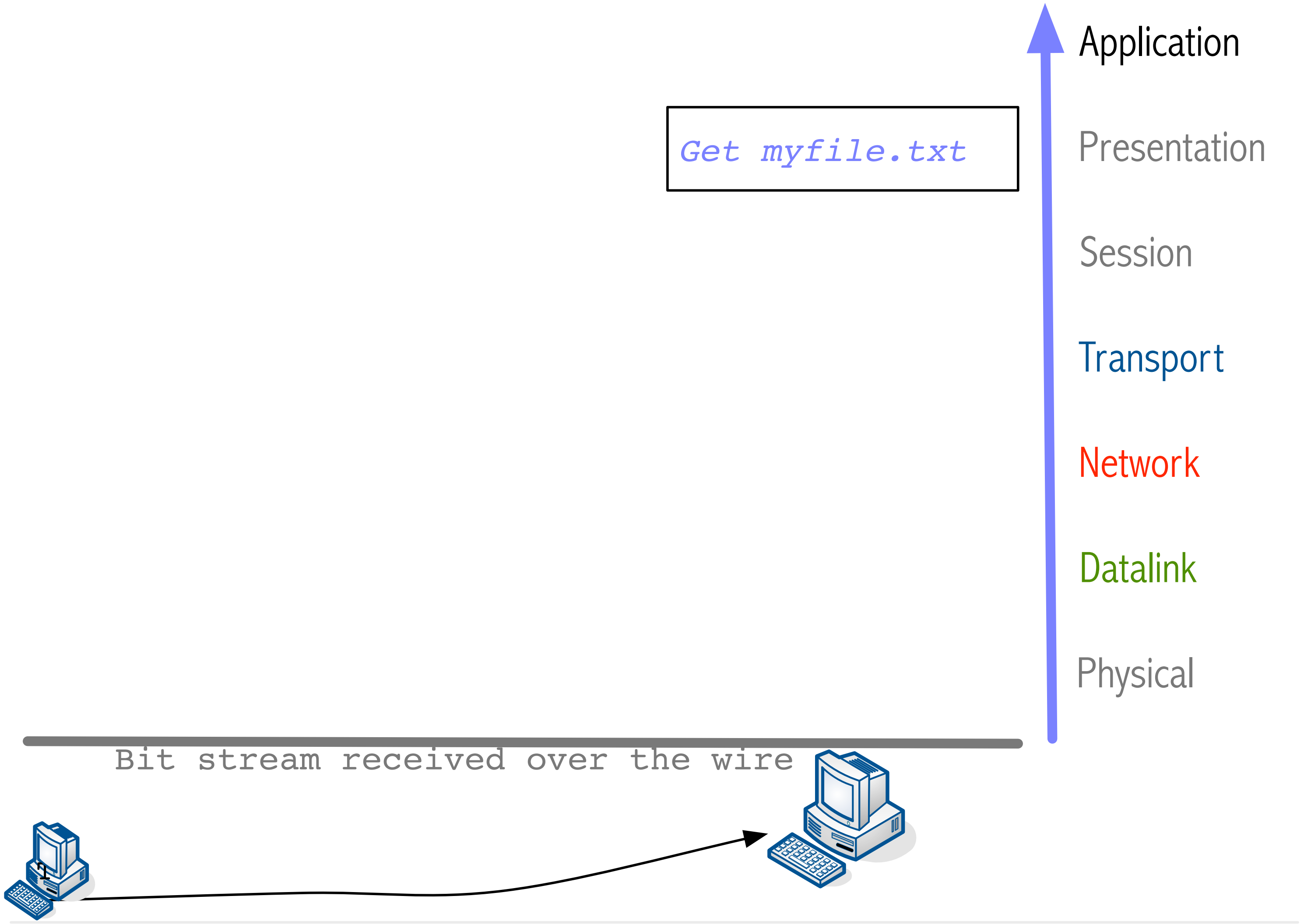
Network

Datalink

Physical

`Get myfile.txt`

`Bit stream received over the wire`



Get myfile.txt

Application

Presentation

Session

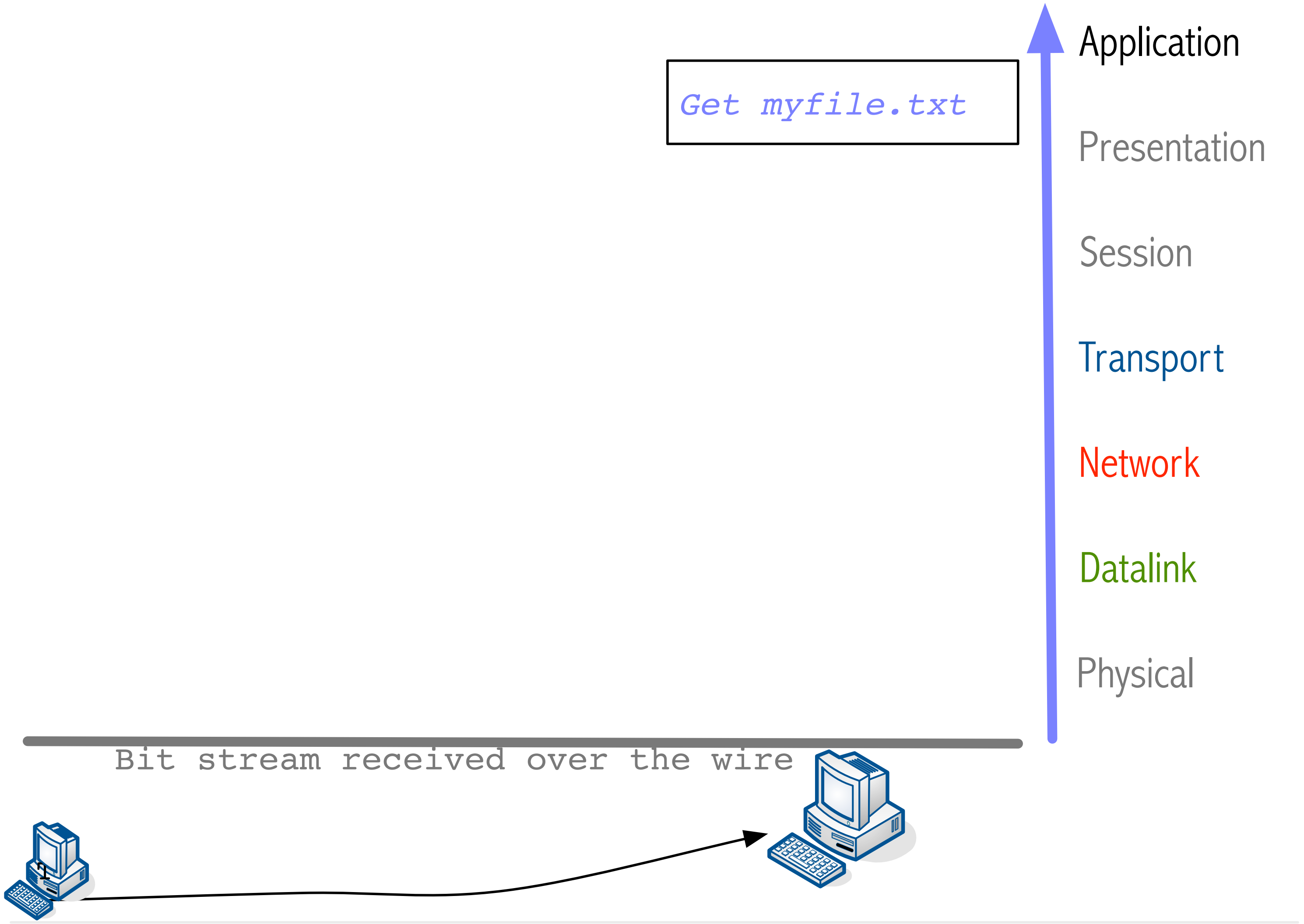
Transport

Network

Datalink

Physical

Bit stream received over the wire



Get myfile.txt

Application

Presentation

Session

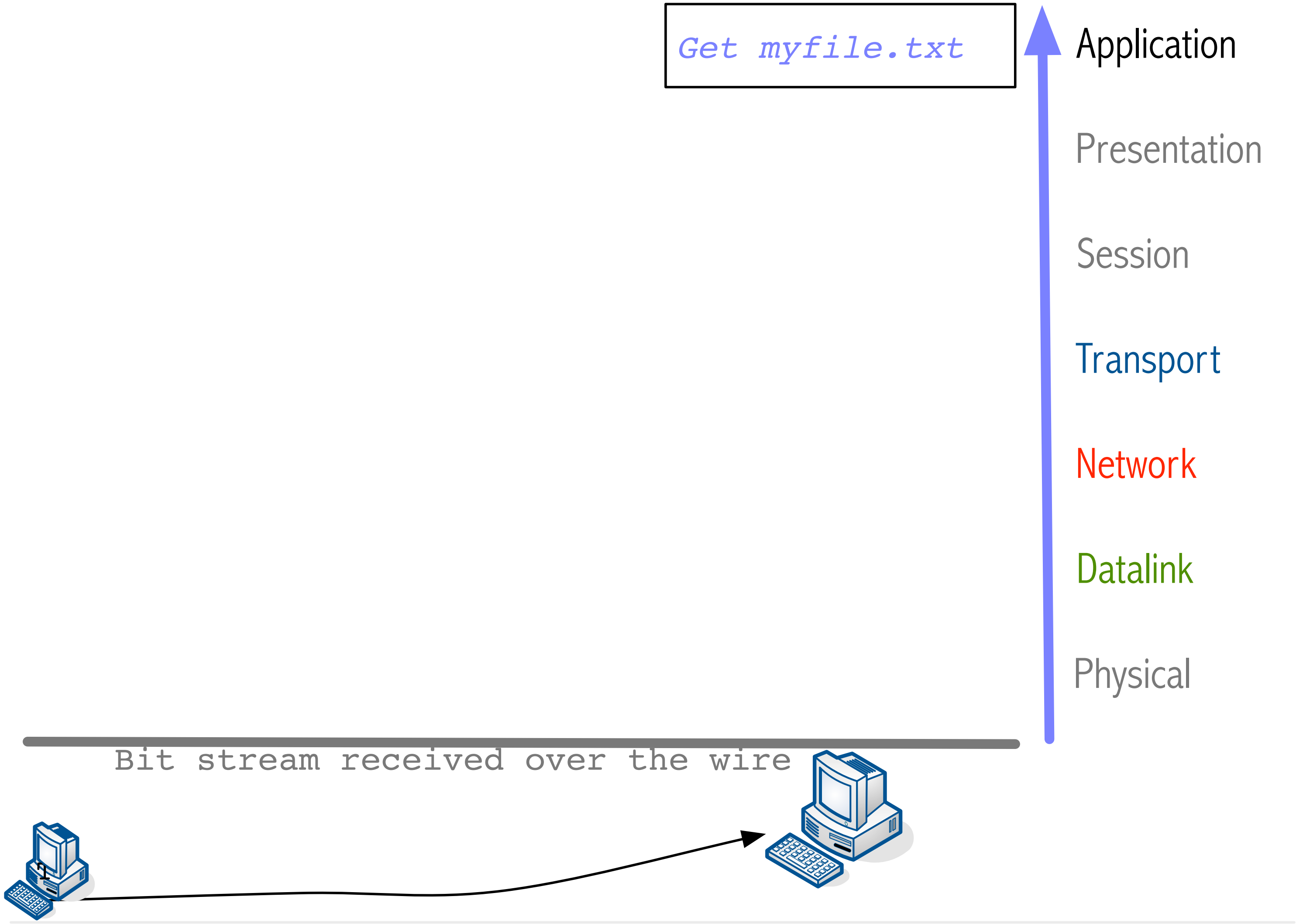
Transport

Network

Datalink

Physical

Bit stream received over the wire



`Get myfile.txt`

Application

Presentation

Session

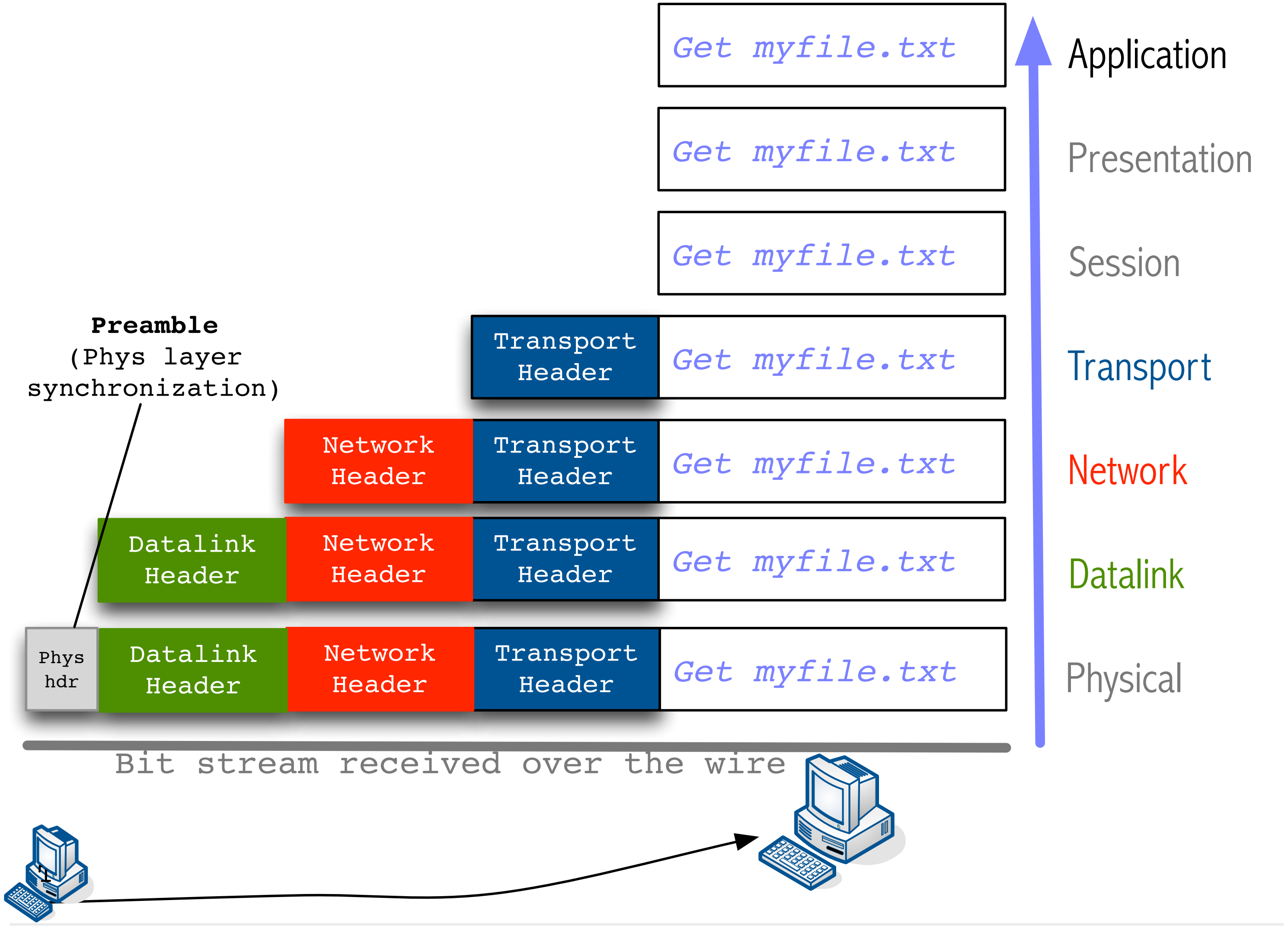
Transport

Network

Datalink

Physical

`Bit stream received over the wire`



Get myfile.txt — Application

Get myfile.txt — Presentation

Get myfile.txt — Session

Transport Header | Get myfile.txt — Transport

Network Header | Transport Header | Get myfile.txt — Network

Datalink Header | Network Header | Transport Header | Get myfile.txt — Datalink

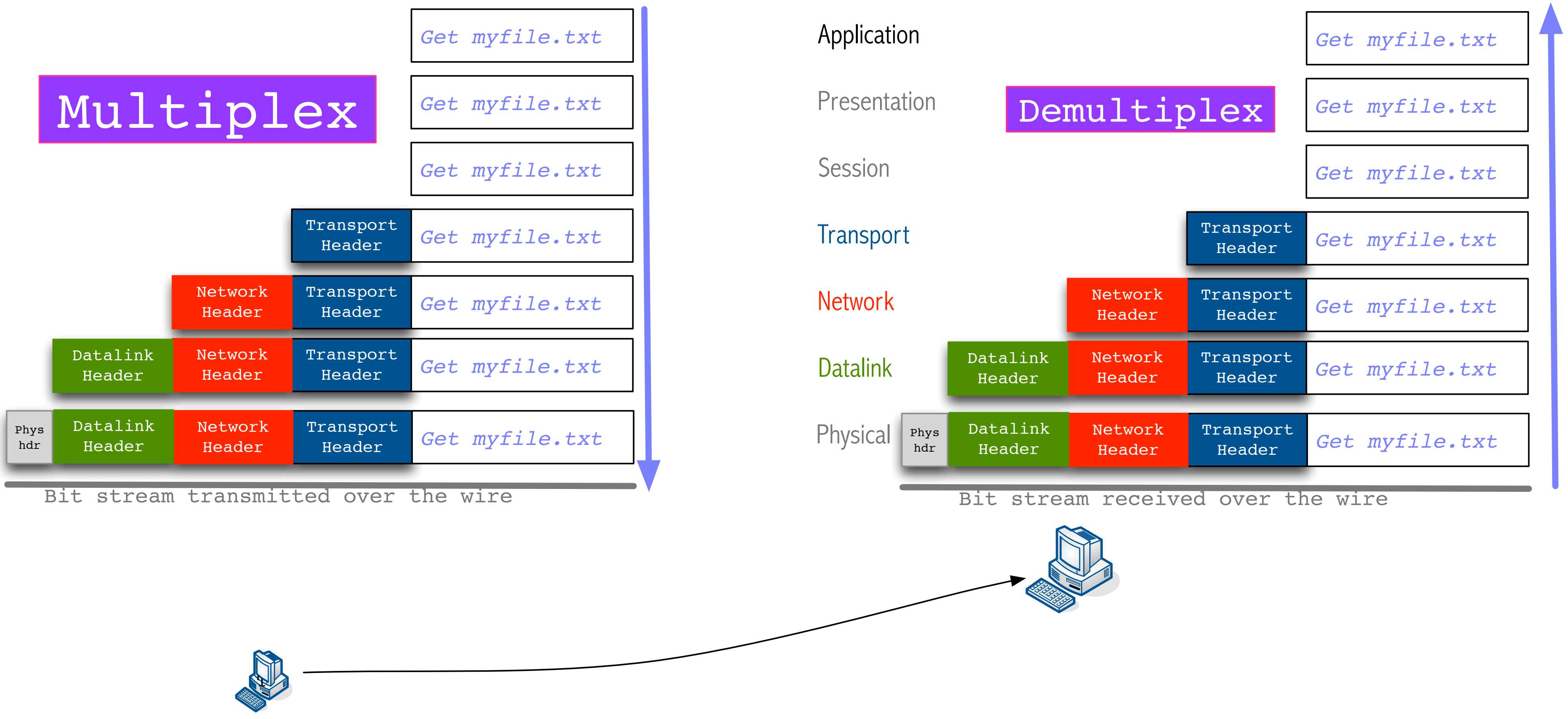
Phys hdr | Datalink Header | Network Header | Transport Header | Get myfile.txt — Physical

Preamble
(Phys layer
synchronization)

Bit stream received over the wire

Multiplexing

- Transmitter multiplexes several flows by having each layer add its header which contains addressing information



Multiplex

Get myfile.txt

Get myfile.txt

Get myfile.txt

Transport Header | Get myfile.txt

Network Header | Transport Header | Get myfile.txt

Datalink Header | Network Header | Transport Header | Get myfile.txt

Phys hdr | Datalink Header | Network Header | Transport Header | Get myfile.txt

Bit stream transmitted over the wire

Application

Presentation

Session

Transport

Network

Datalink

Physical

Demultiplex

Get myfile.txt

Get myfile.txt

Get myfile.txt

Transport Header | Get myfile.txt

Network Header | Transport Header | Get myfile.txt

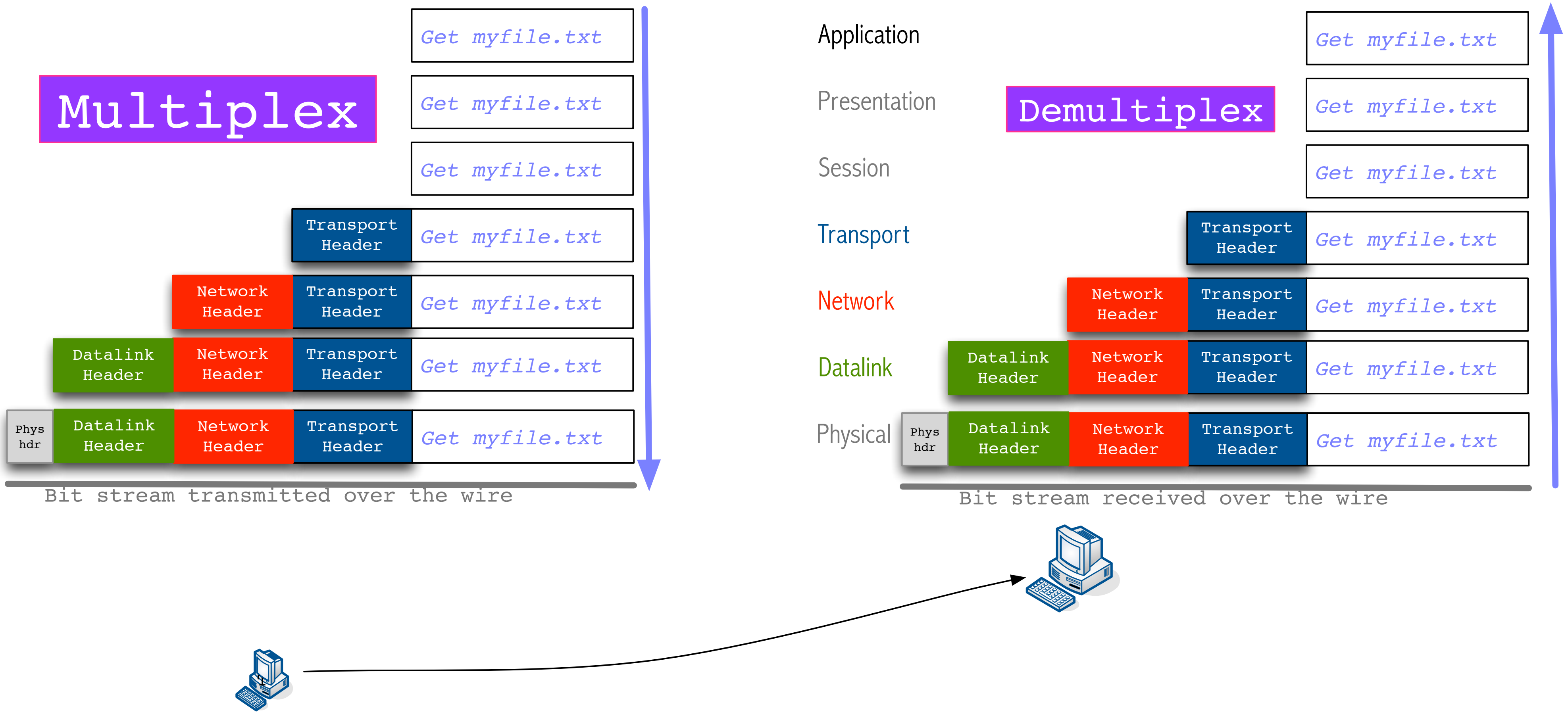
Datalink Header | Network Header | Transport Header | Get myfile.txt

Phys hdr | Datalink Header | Network Header | Transport Header | Get myfile.txt

Bit stream received over the wire

Demultiplexing

- □ Receiver demultiplexes several flows by having each layer analyze its header which contains addressing information about the upper-layer protocol that is to receive the payload



Multiplex

Get myfile.txt

Get myfile.txt

Get myfile.txt

Transport Header | Get myfile.txt

Network Header | Transport Header | Get myfile.txt

Datalink Header | Network Header | Transport Header | Get myfile.txt

Phys hdr | Datalink Header | Network Header | Transport Header | Get myfile.txt

Bit stream transmitted over the wire

Application — Get myfile.txt

Presentation — Demultiplex — Get myfile.txt

Session — Get myfile.txt

Transport — Transport Header | Get myfile.txt

Network — Network Header | Transport Header | Get myfile.txt

Datalink — Datalink Header | Network Header | Transport Header | Get myfile.txt

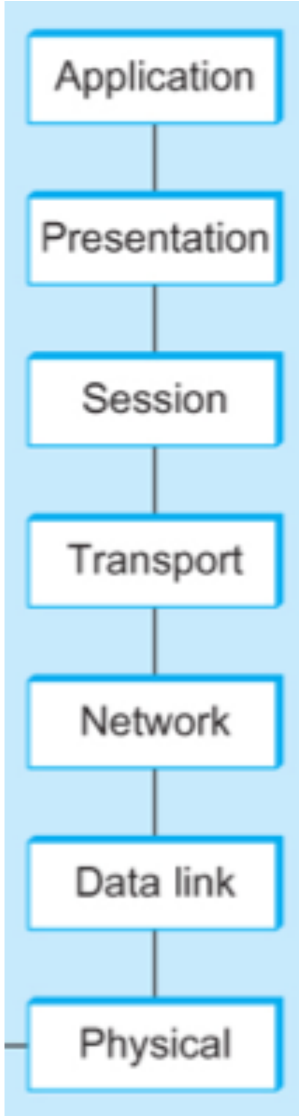
Physical — Phys hdr | Datalink Header | Network Header | Transport Header | Get myfile.txt

Bit stream received over the wire

Description of Layers

- **Physical Layer**
  - Handles the transmission of raw bits over a communication link
- **Data Link Layer**
  - Collects a stream of bits into a larger aggregate called a *frame*
  - Network adaptor along with device driver in OS implement the protocol in this layer
  - Frames are actually delivered to hosts
- **Network Layer**
  - Handles routing among nodes within a packet-switched network
  - Unit of data exchanged between nodes in this layer is called a *packet*

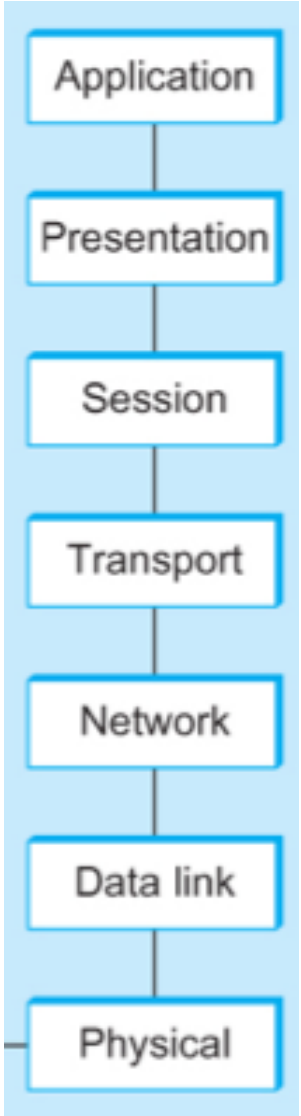
The lower three layers are implemented **on all network nodes**



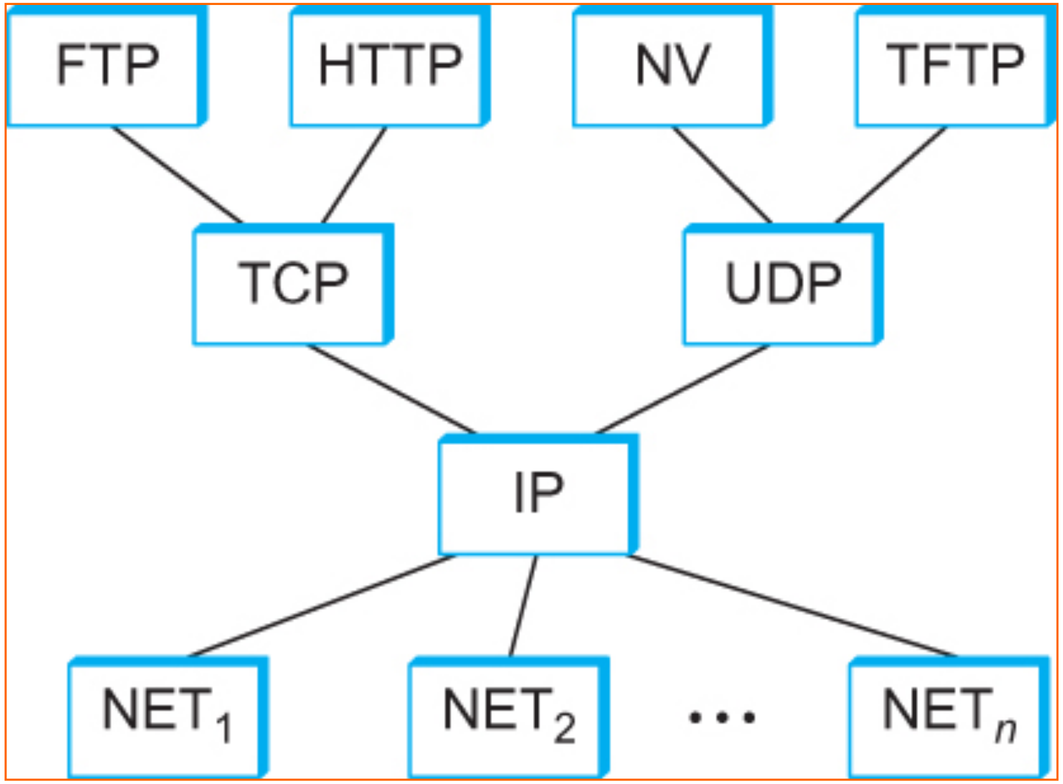
Description of Layers

- Transport Layer
  - Implements a process-to-process channel
  - Unit of data exchanges in this layer is called a *message*
- Session Layer
  - Provides a name space that is used to tie together the potentially different transport streams that are part of a single application
- Presentation Layer
  - Concerned about the format of data exchanged between peers
- Application Layer
  - Standardize common type of exchanges

The transport layer and the higher layers typically run only on end-hosts and not on the intermediate switches and routers



Internet Architecture



FTP HTTP NV TFTP

TCP UDP

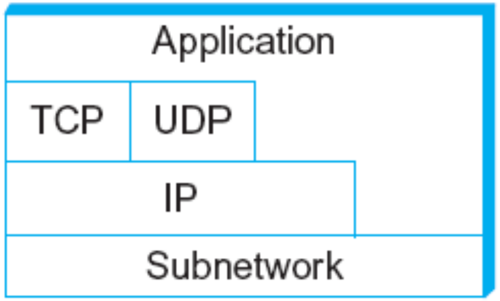
IP

NET_1 NET_2 … NET_n

Example of an

Internet Protocol Graph

TCP/IP architecture!



Application

TCP UDP

IP

Subnetwork

Alternative view of the Internet architecture. The "Network" layer shown here is sometimes referred to as the "sub-network" or "link" layer.

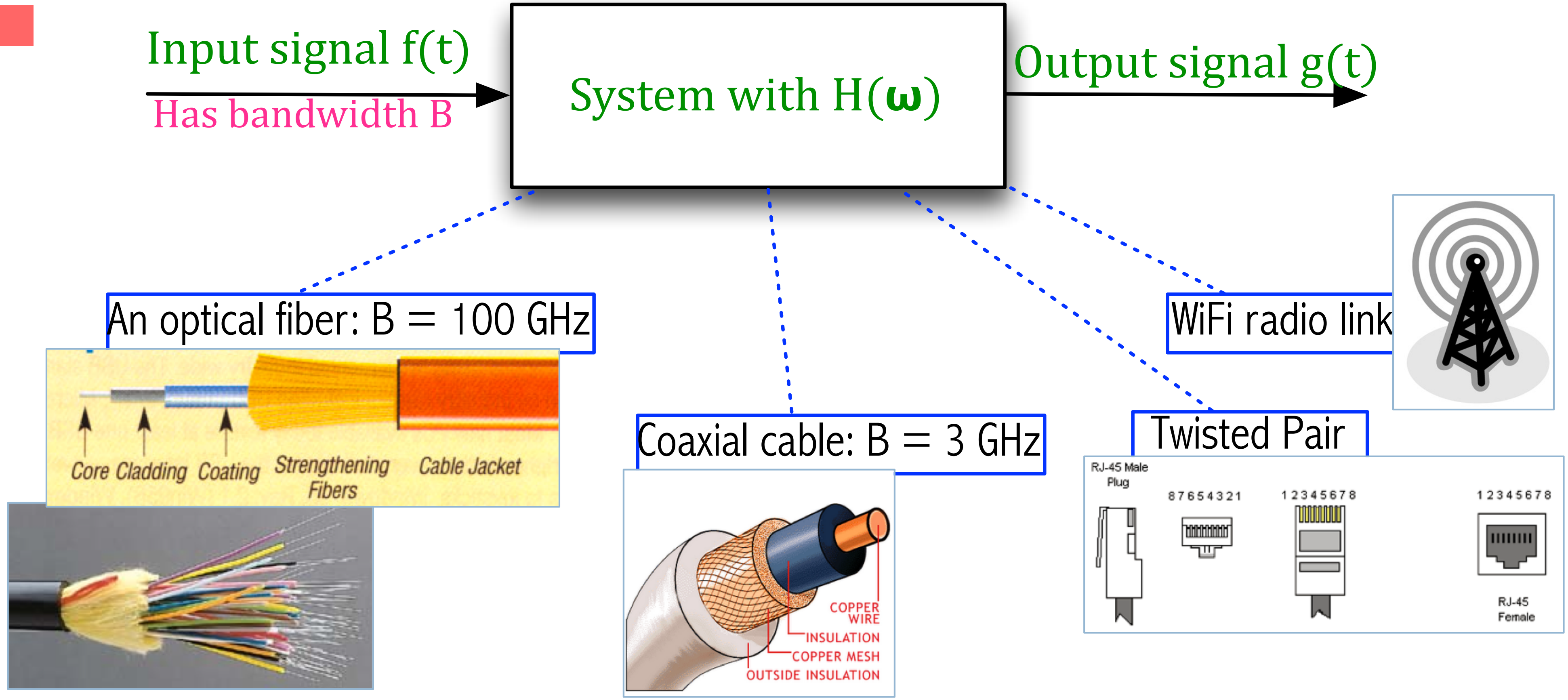
Internet Architecture

- Defined by **IETF** (Internet Engineering Task Force)
- Three main features
  - Does **not imply strict layering**. The application is free to bypass the defined transport layers and to directly use IP or other underlying networks
  - An **hour-glass** shape – wide at the top, narrow in the middle and wide at the bottom. IP serves as the focal point for the architecture
  - In order for a new protocol to be officially included in the architecture, there needs to be both a protocol specification and at least one (and preferably two) representative implementations of the specification

Network Performance: Networks must be fast

What are the essential network performance metrics: bandwidth and latency

Transmission media AS systems



Input signal $f(t)$

Has bandwidth B

System with $H(\boldsymbol{\omega})$

Output signal $g(t)$

An optical fiber: $B = 100$ GHz

Coaxial cable: $B = 3$ GHz

Twisted Pair

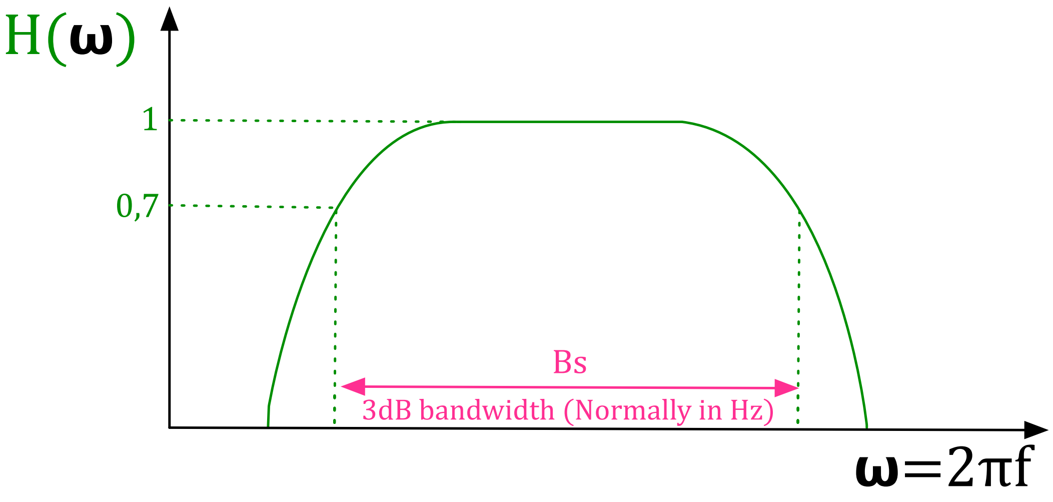
WiFi radio link

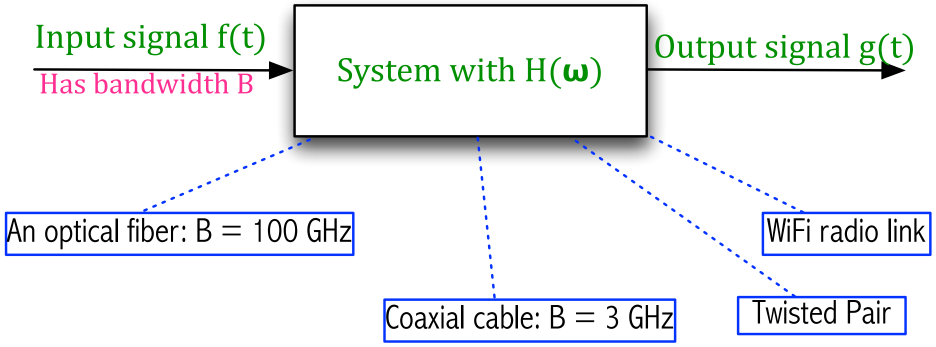
Bandwidth is a property of every system

The width of the accepted frequency band

- Measured in Hz = 1/s
- In networking bps means bits per second
  - bits/s
  - bits · Hz

- Higher bandwidth (Hz) allows a higher transmission speed





Input signal f(t)

Has bandwidth B

System with H(ω)

Output signal g(t)

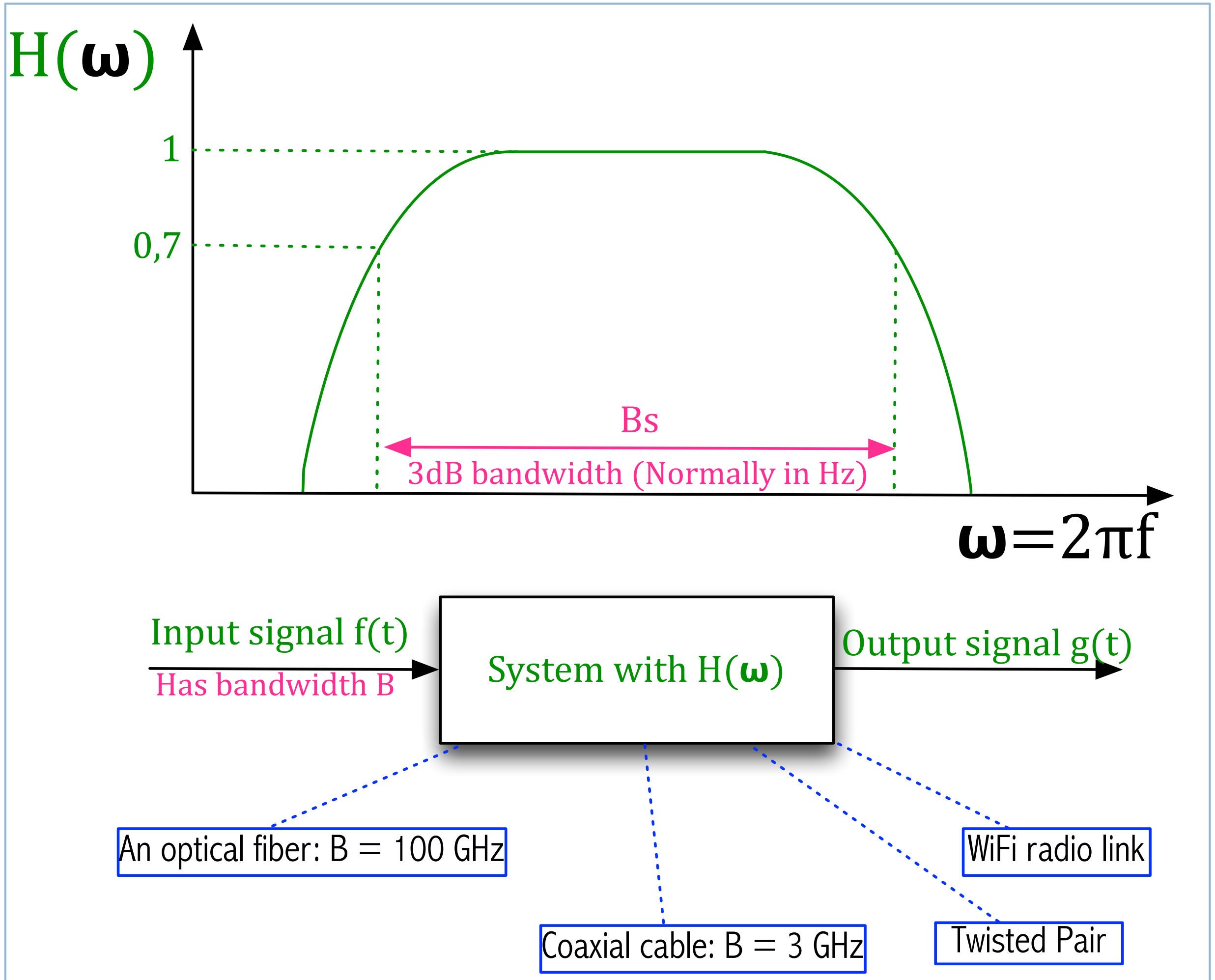
An optical fiber: B = 100 GHz

Coaxial cable: B = 3 GHz

Twisted Pair

WiFi radio link

Bandwidth and transmission speed



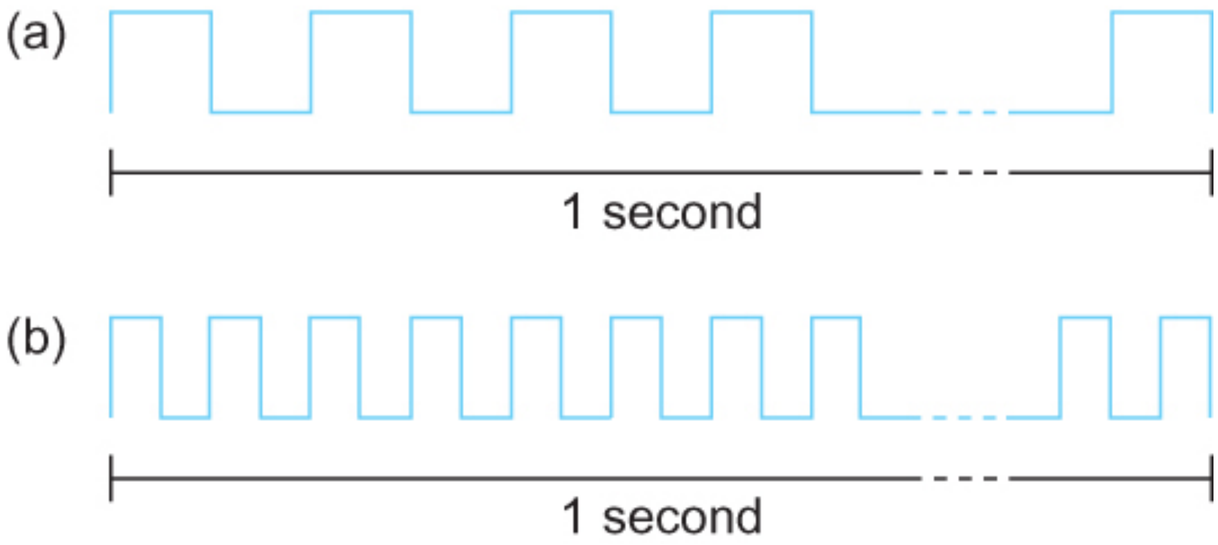
If B > Bs => output will have big distortion
If Bs > B => output will have little distortion

- If the band can hold the signal, it passes cleanly on to the output
- Bs in Hz is the inverse of the signaling SPEED in seconds
- High bps needs high bandwidth => high speed demands high bandwidth
- High bandwidth permits high speed transmission

Performance

- **Bandwidth**
  - Width of the frequency band
  - Number of bits per second that can be transmitted over a communication link in a given period of time
  - The inverse of Bw is the time it takes to transmit one bit
    - 1Mbps = 1M bits/s = 10^6 bits/s → T_{transm}1 bit = 10^{-6} s/bit · 1bit = 1μs
  - Higher bandwidth (speed) means shorter transmission times
- Multipliers used in expressing speeds (ratios)
  - 1Kbps = 1K bits/s = 10^3 bits/s
  - 1Mbps = 1M bits/s = 10^6 bits/s
  - 1Gbps = 1G bits/s = 10^9 bits/s

Bandwidth



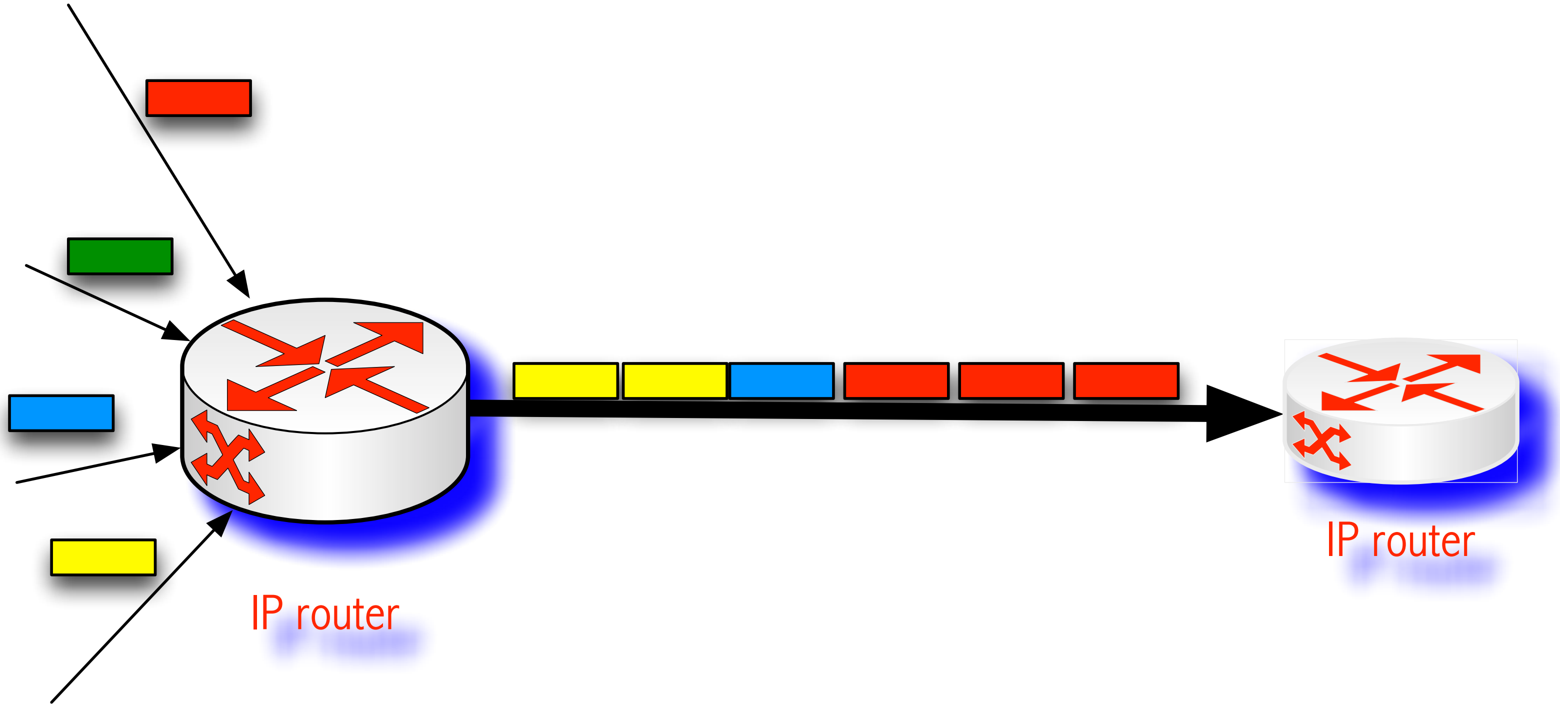
Bits transmitted at a particular bandwidth can be regarded as having some width:

(a) bits transmitted at 1Mbps (each bit 1 µs wide);

(b) bits transmitted at 2Mbps (each bit 0.5 µs wide).

Context for analyzing the components of performance in networks

An IP router is a form of statistical multiplexer



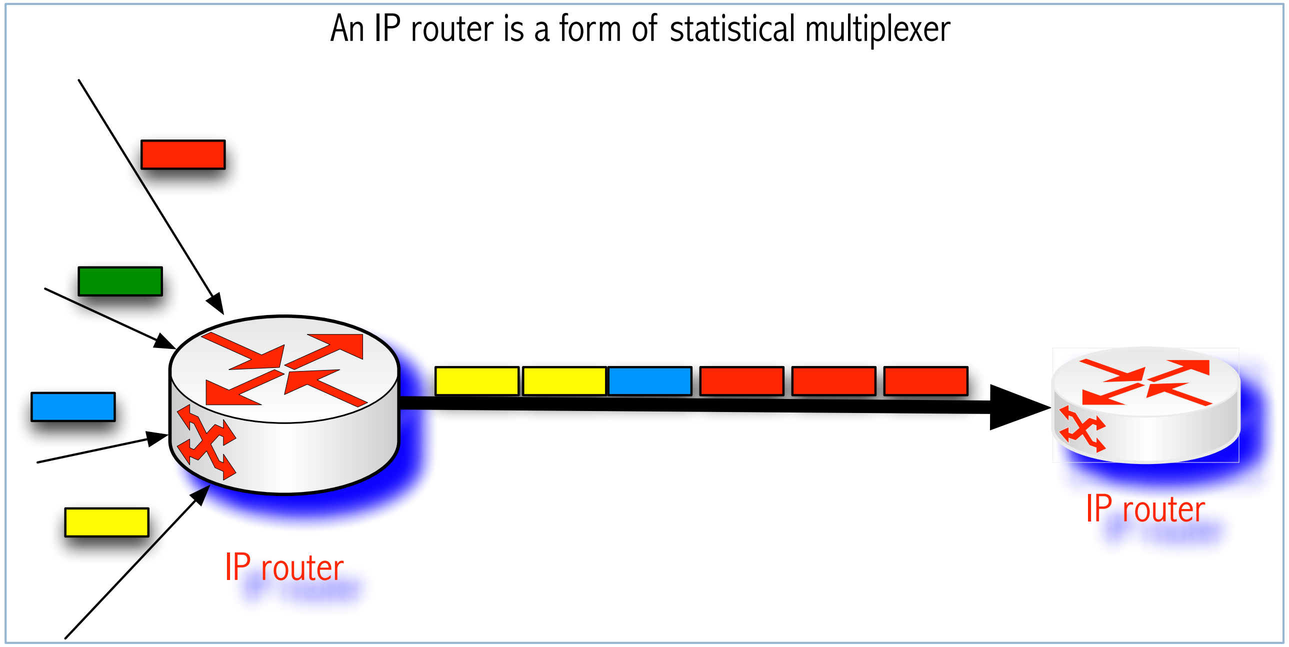
Performance

- Latency = Propagation + transmit + queue
- Propagation = distance/speed of light
- Transmit = size/bandwidth

- One bit transmission => propagation is important
- Large bytes transmission => bandwidth is important

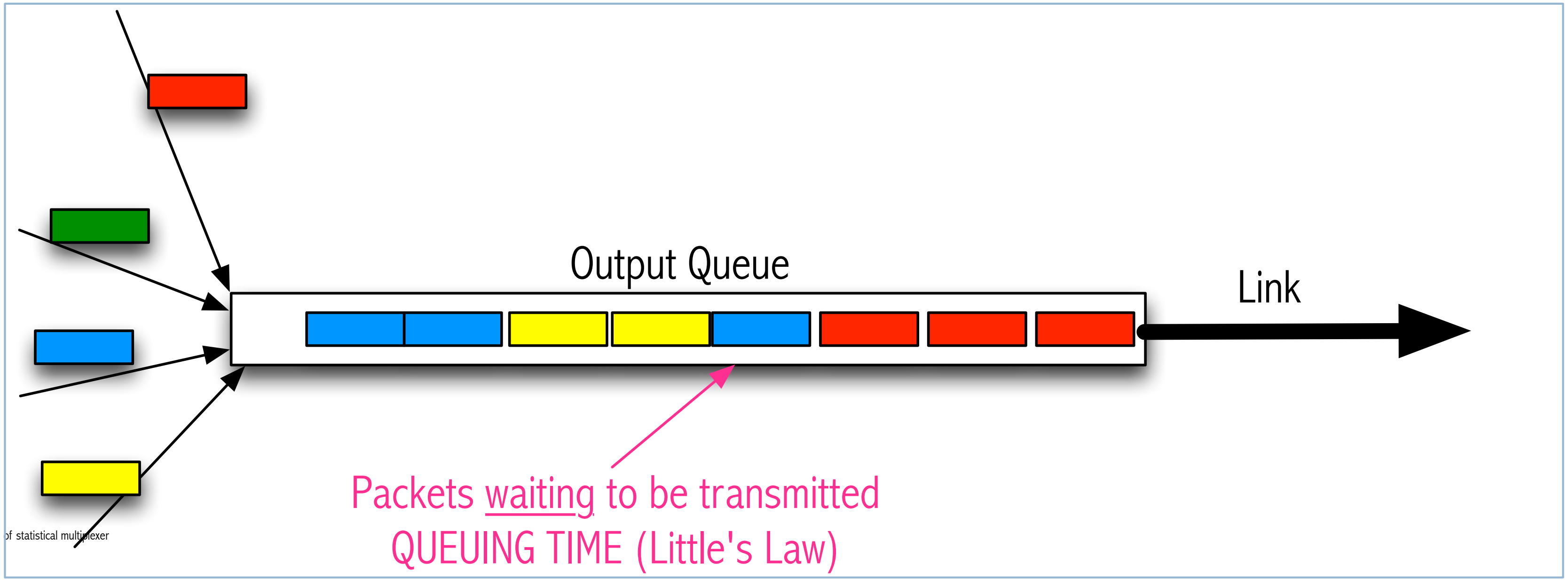
How much does a packet take to be *transferred* to its destination?

- □ Queuing time
- □ Transmission time
- □ Propagation time



Queuing time

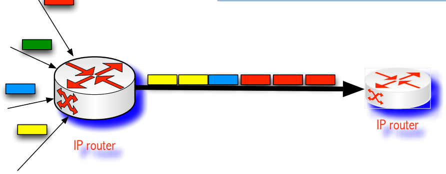
- □ The router has to enqueue the packets received from the various sources
  - Otherwise, they will be lost



Packets waiting to be transmitted

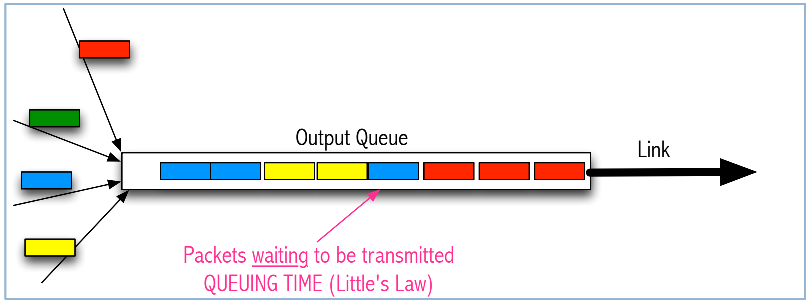
QUEUING TIME (Little's Law)

An IP router is a form of statistical multiplexer



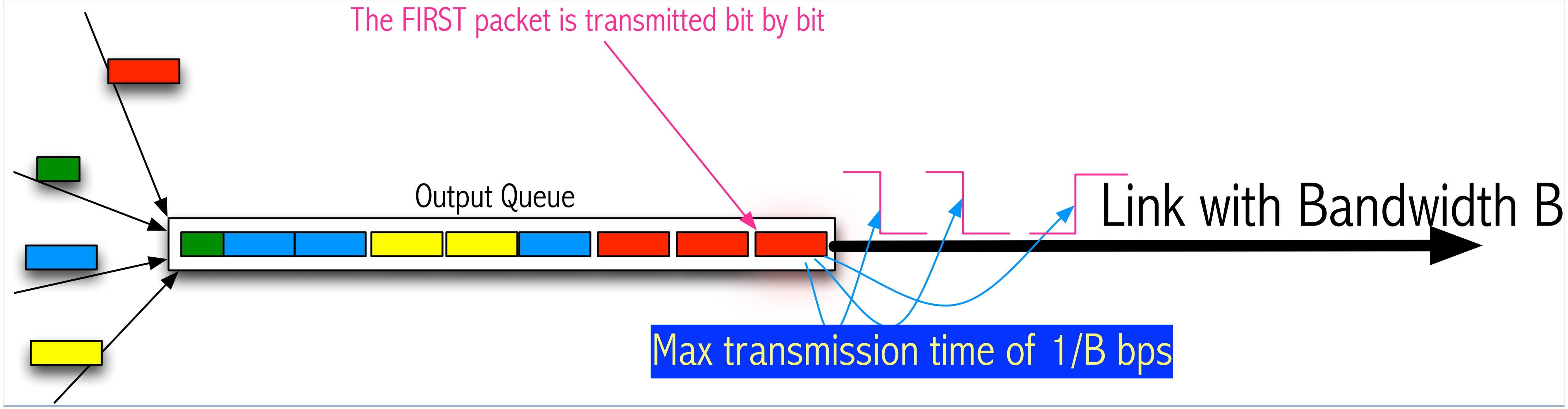
Transmission time

- The **packet** on the **first position in Queue** is transmitted
  - Packet bits are turned into signals (Electrical, Electromagnetic, Optical)
  - 1 bit from packet -> 1 signal waveform
  - At which **speed** (bps = bits/s) can this be done?
  - High Bw => High speed


Output Queue
Link
Packets waiting to be transmitted
QUEUING TIME (Little's Law)

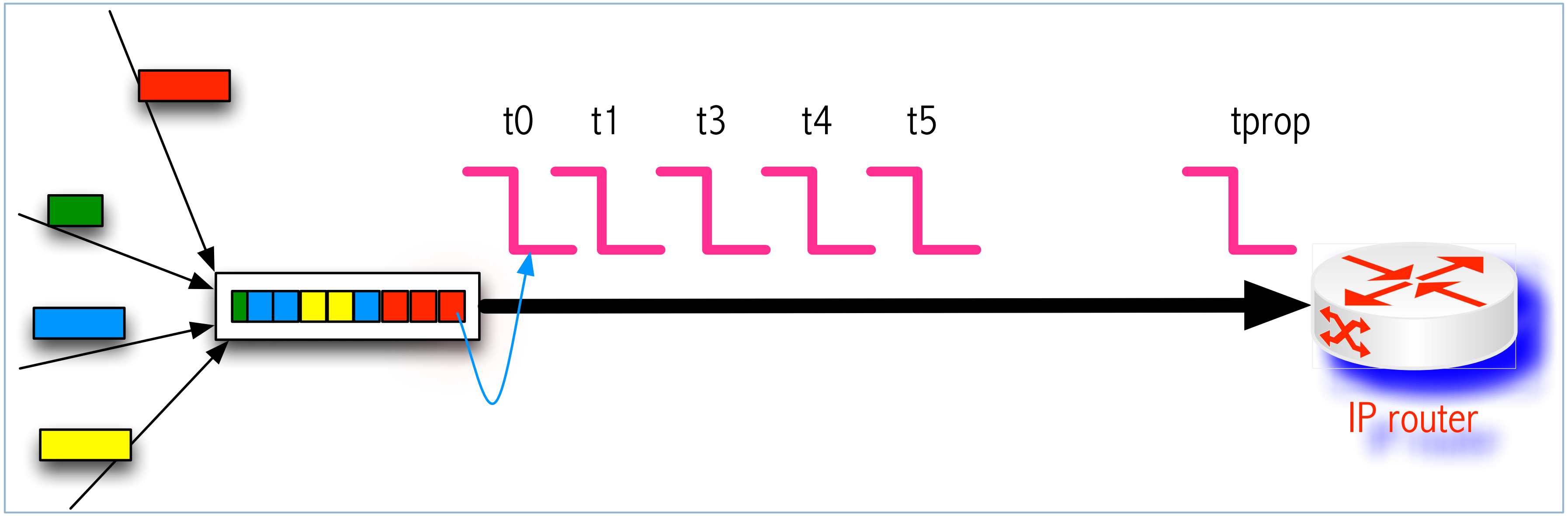
Transmission time

- The packet on the first position in Queue is transmitted
  - Packet bits are turned into signals (Electrical, Electromagnetic, Optical)
  - 1 bit from packet -> 1 signal
  - At which speed (bps = bits/s) can this be done?
  - High Bw => High speed


The FIRST packet is transmitted bit by bit
Output Queue
Link with Bandwidth B
Max transmission time of 1/B bps

Propagation time (t_{prop})

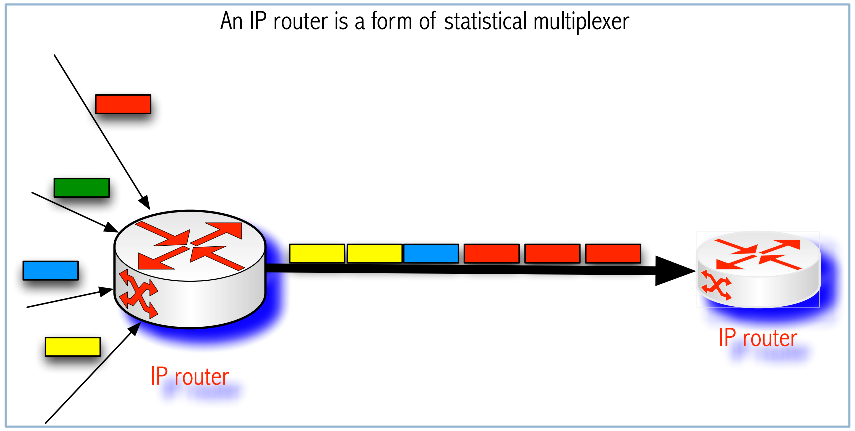
- Now, each signal must propagate over the wire until it reaches the receiver
  - Speed of light in empty space c = 3 x 10^8 m/s
  - Speed of light in other media
    - Copper: 2,3 x 10^8 m/s
    - Optical fibers: 2,0 x 10^8 m/s



Latency = total time to transfer one packet

- □ Latency = Propagation + transmit + queue
- □ Propagation = distance/speed of light
- □ Transmit = size/bandwidth

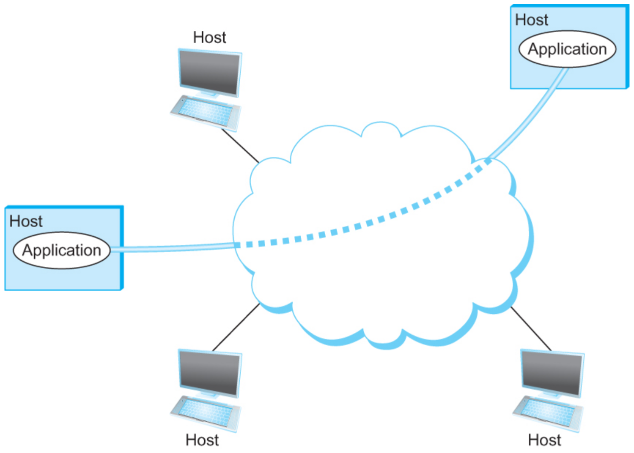
- □ If only one bit is transmitted => propagation is important
  - Or a small amount of bits
- □ If the amount of bits transmitted is large => bandwidth is important



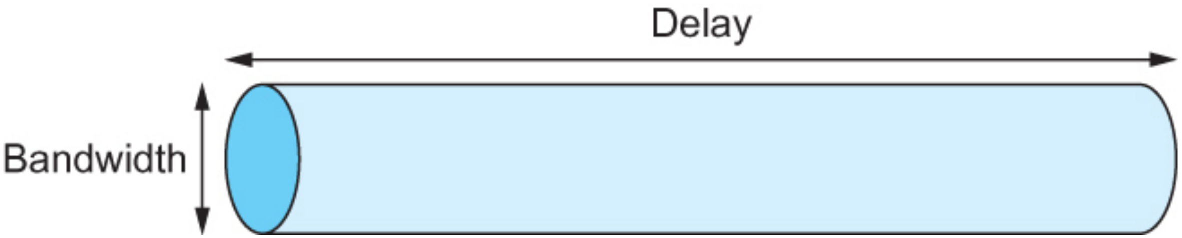
Delay x Bandwidth

- □ We think of the channel between a pair of processes as a hollow pipe

- □ Latency (delay) length of the pipe and bandwidth



- □ Delay of 50 ms and bandwidth of 45 Mbps
  - ⇒ 50 x 10^{-3} seconds x 45 x 10^{6} bits/second
  - ⇒ 2.25 x 10^{6} bits = 280 KB data.



Network as a pipe

Delay X Bandwidth

- Relative importance of bandwidth and latency depends on application
  - For large file transfer, bandwidth is critical
  - For small messages (HTTP, NFS, etc.), latency is critical
  - Variance in latency (jitter) can also affect some applications (*e.g.*, audio/video conferencing)

Delay X Bandwidth

- How many bits the sender must transmit before the first bit arrives at the receiver if the sender keeps the pipe full

- Takes another one-way latency to receive a response from the receiver

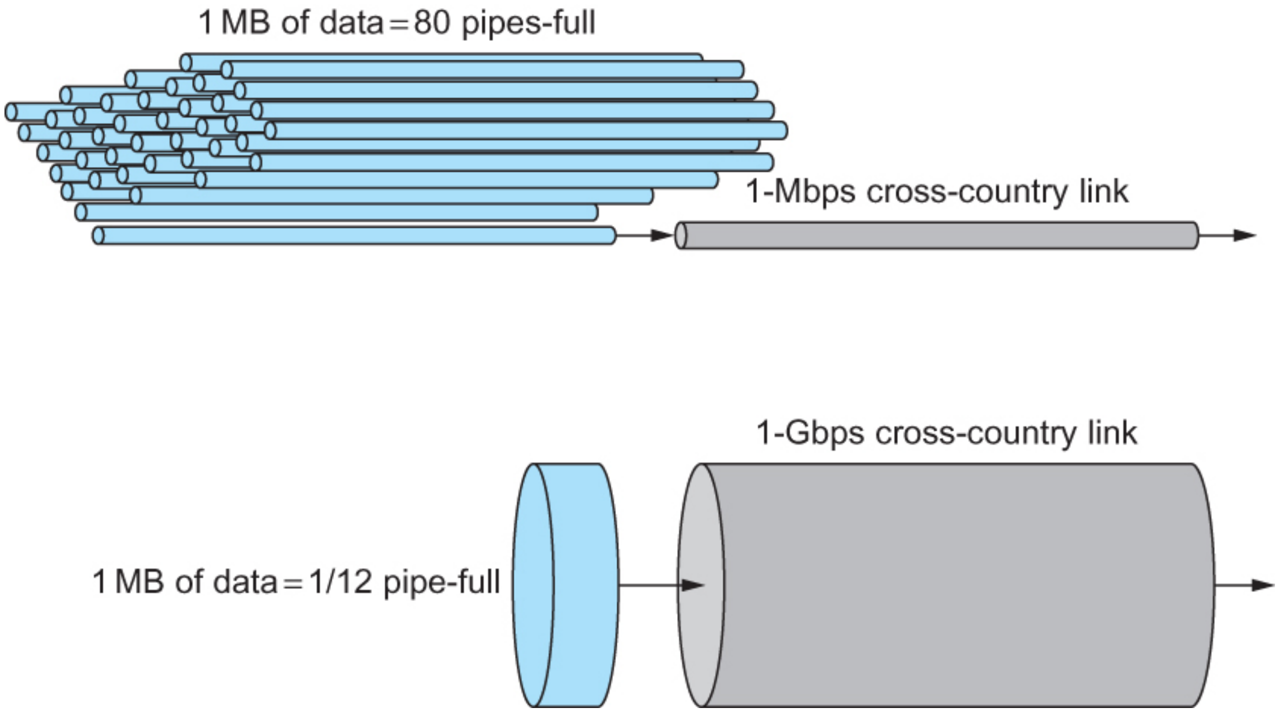
- If the sender does not fill the pipe—send a whole delay × bandwidth product's worth of data before it stops to wait for a signal—the sender will not fully utilize the network

Delay X Bandwidth

- Infinite bandwidth
  - RTT dominates
  - Throughput = TransferSize / TransferTime
  - TransferTime = RTT + 1/Bandwidth x TransferSize

- It's all relative
  - 1-MB file to 1-Gbps link looks like a 1-KB packet to 1-Mbps link

Relationship between bandwidth and latency


1 MB of data = 80 pipes-full
1-Mbps cross-country link
1-Gbps cross-country link
1 MB of data = 1/12 pipe-full

A 1-MB file would fill the 1-Mbps link 80 times, but only fill the 1-Gbps link 1/12 of one time

Summary

- We have identified what we expect from a computer network
- We have defined a layered architecture for computer network that will serve as a blueprint for our design
- We have discussed the socket interface which will be used by applications for invoking the services of the network subsystem
- We have discussed two performance metrics using which we can analyze the performance of computer networks

Exercises

What are the essential network performance metrics: bandwidth and latency