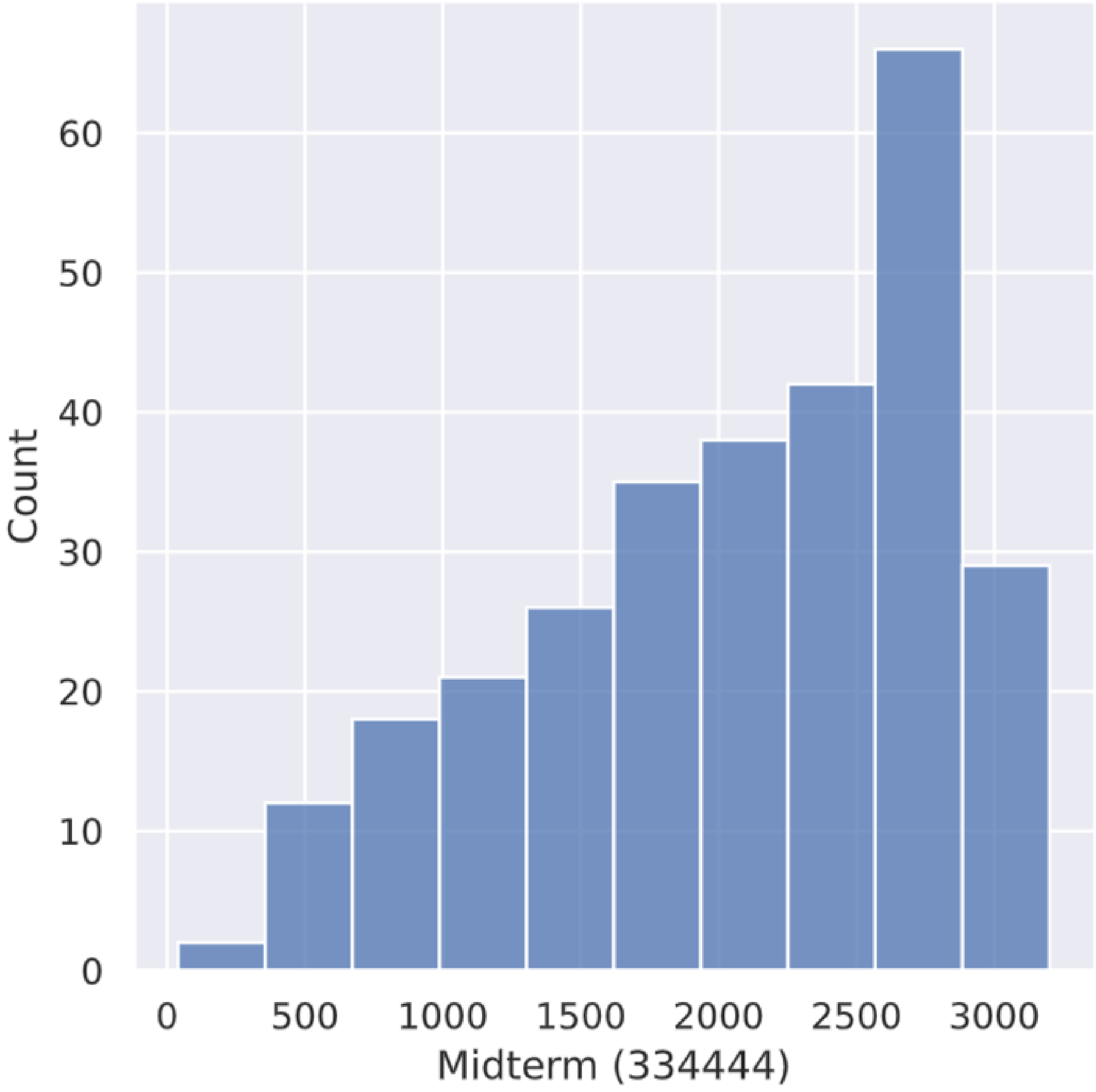
Computer Networks X_400487

Lecture 8

Chapter 5: The Network Layer—Part 2

Lecturer: Jesse Donkervliet

Well done at the Midterm!



ALL YOUR NETWORKS ARE BELONG TO IP!

MINECRAFT

FORTNITE

STADIA

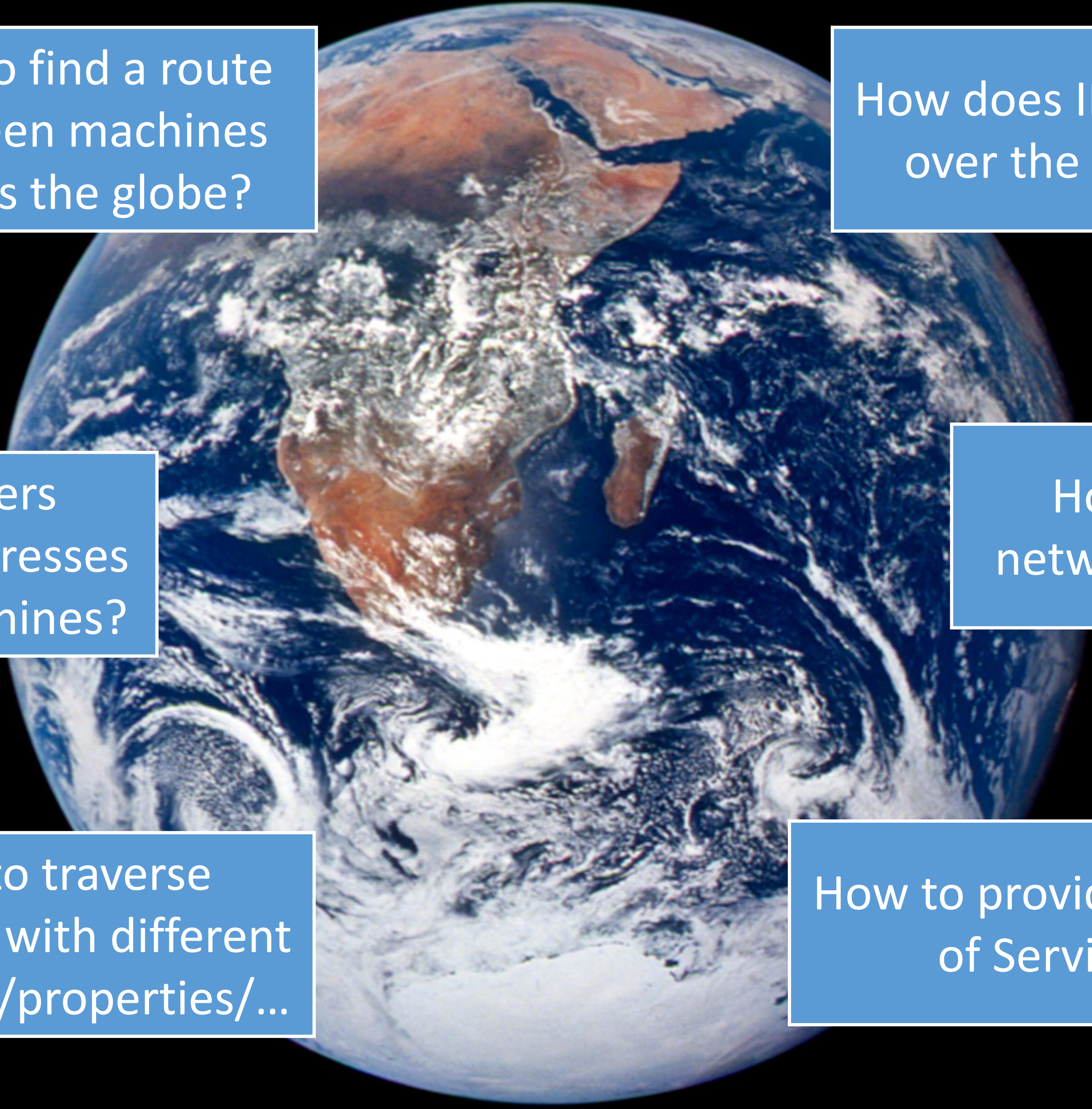
skype™

NVIDIA
GEFORCE NOW™

twitch

CANVAS
BY INSTRUCTURE

LEAGUE of LEGENDS



How to find a route between machines across the globe?

How does IP carry data over the Internet?

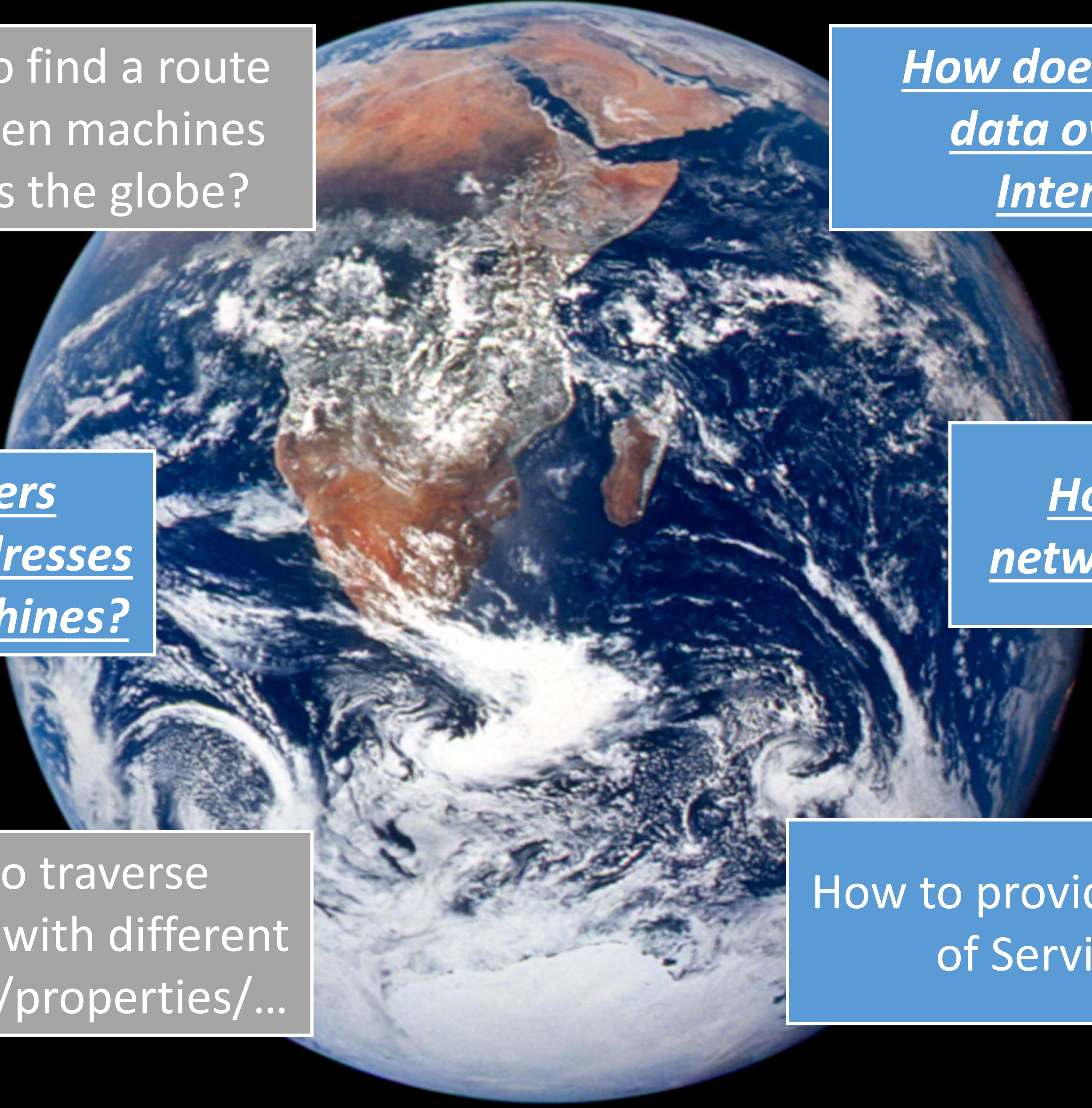
How do routers manage the addresses of all these machines?

How to prevent network congestion?

How to traverse networks with different protocols/properties/…

How to provide quality of Service?

Image source: NASA



How to find a route between machines across the globe?

__How does IP carry data over the Internet?__

__How do routers manage the addresses of all these machines?__

__How to prevent network congestion?__

How to traverse networks with different protocols/properties/…

How to provide quality of Service?

Image source: NASA

Quick Links for Today

1. IPv4
2. NAT
3. Subnets
4. Token Bucket

The Internet Protocol (IP)

Network Layer Protocol

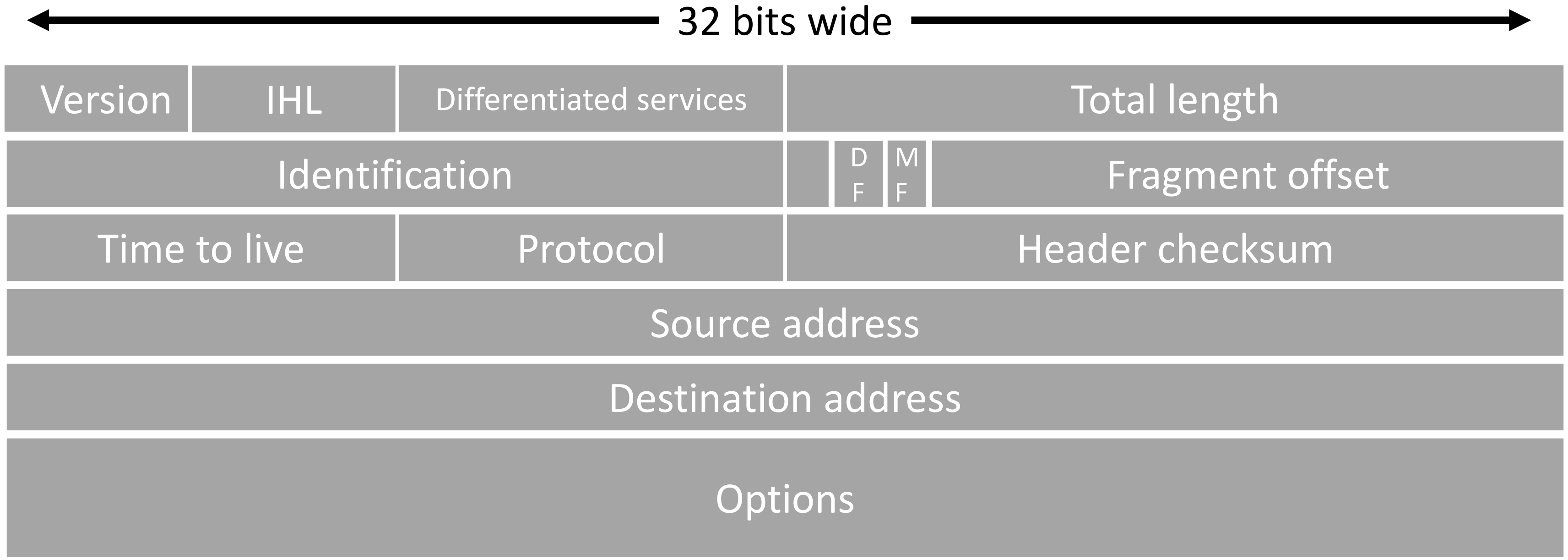
https://unsplash.com/photos/fN603qcEA7g

Challenges Addressed by IPv4 Protocol Design

1. Error detection/correction
2. Preventing permanently looping packets
3. Globally identifying computers
4. Carrying packets over links with different size requirements

IP version 4

Frame header



32 bits wide

Version | IHL | Differentiated services | Total length

Identification | | DF | MF | Fragment offset

Time to live | Protocol | Header checksum

Source address

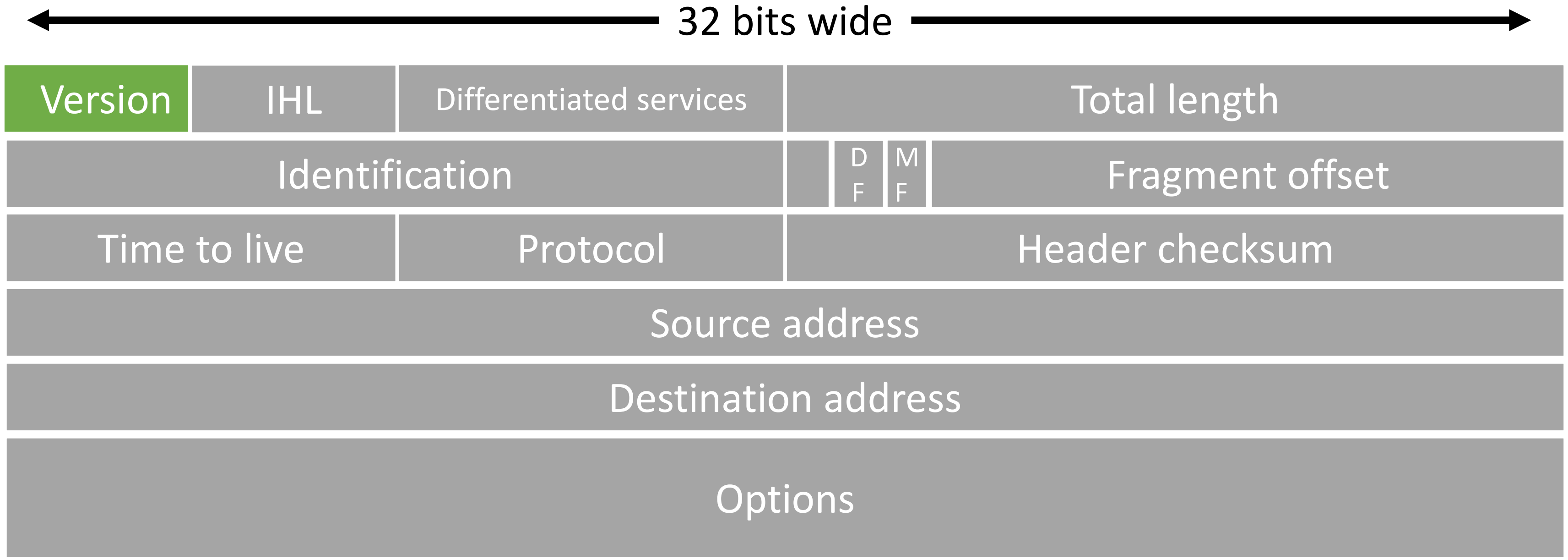
Destination address

Options

Check the book for the detailed view!

IP version 4

Frame header



32 bits wide

Version | IHL | Differentiated services | Total length

Identification | DF | MF | Fragment offset

Time to live | Protocol | Header checksum

Source address

Destination address

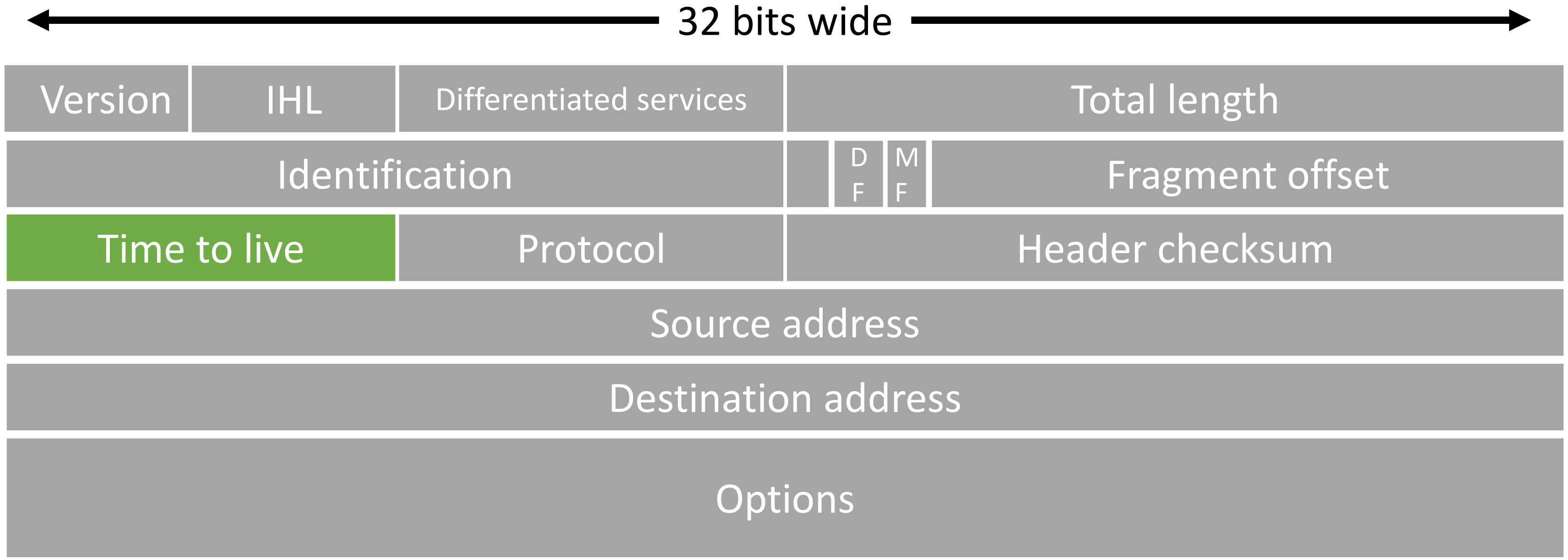
Options

Q: What is the value of this field?

Copyright Jesse Donkervliet 2024

IP version 4

Frame header



32 bits wide

Version | IHL | Differentiated services | Total length

Identification | DF | MF | Fragment offset

Time to live | Protocol | Header checksum

Source address

Destination address

Options

Q: Why have this field?

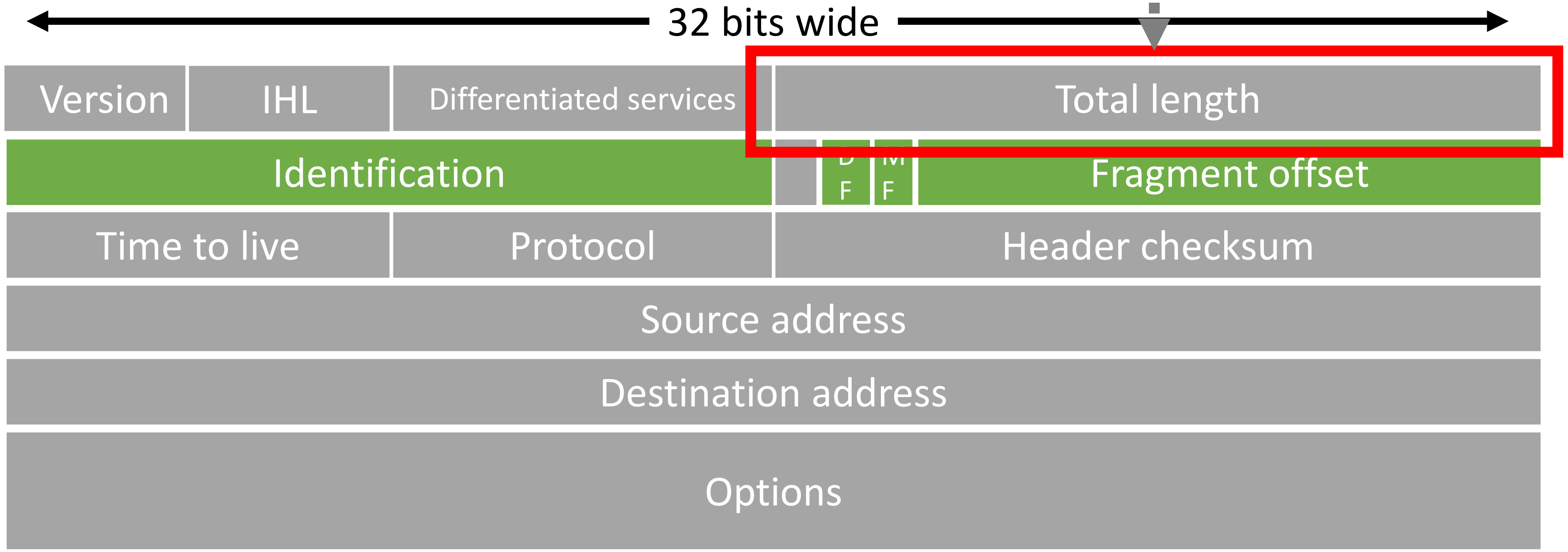
Challenges Addressed by IPv4 Protocol Design

1. Error detection/correction
2. ~~Preventing permanently looping packets~~
3. Globally identifying computers
4. Carrying packets over links with different size requirements

IP version 4

16 bit length field → packet size can reach 64KiB

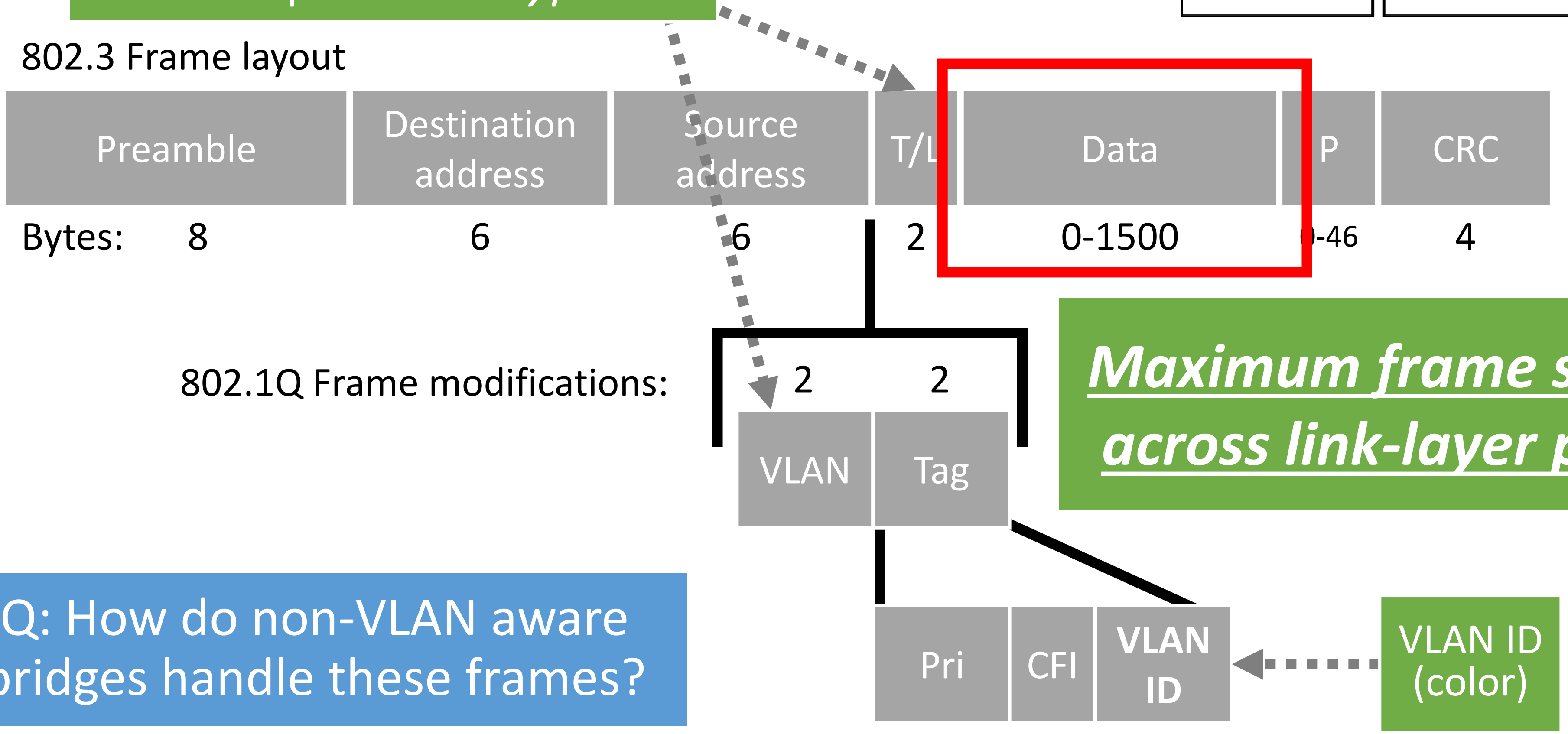
Frame header



Q: Why have this field?

802.3 → 802.1Q

VLAN type > 1500, interpreted as *type*



802.3 Frame layout

Preamble	Destination address	Source address	T/L	Data	P	CRC
Bytes: 8	6	6	2	0-1500	0-46	4

802.1Q Frame modifications:

VLAN	Tag
2	2

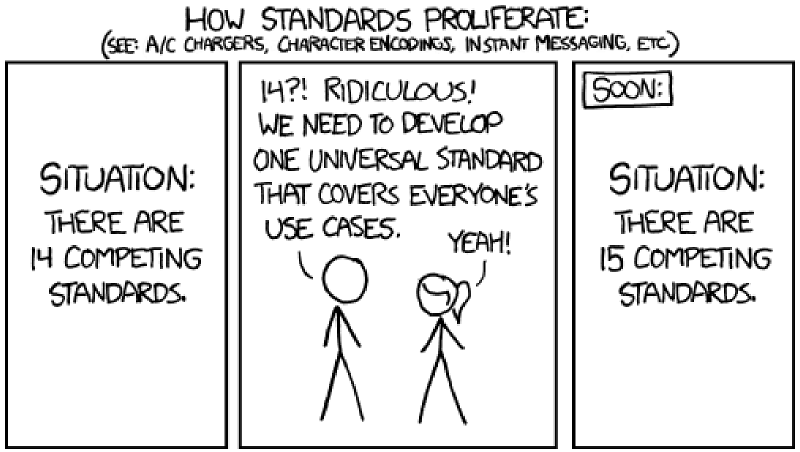
Tag:

Pri	CFI	**VLAN ID**

VLAN ID (color)

Maximum frame size differs across link-layer protocols!

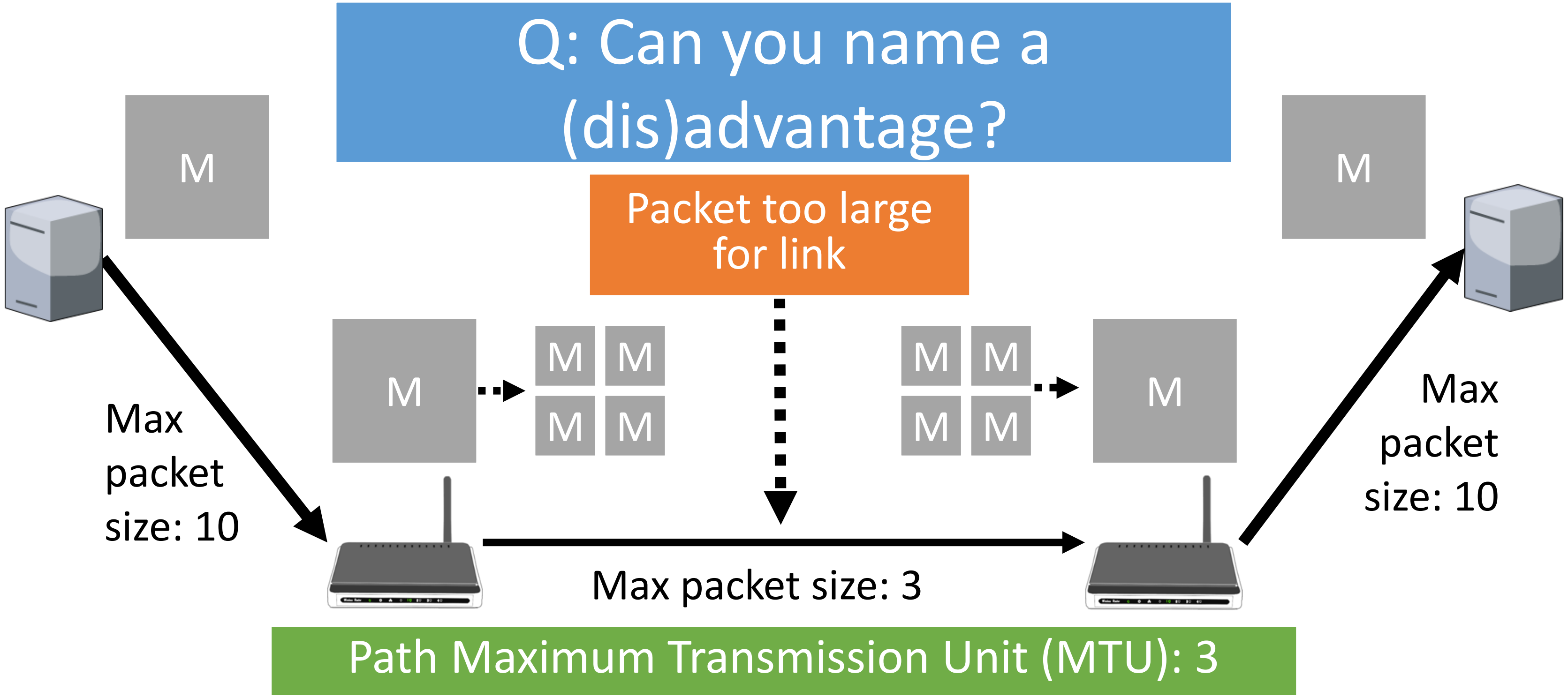
Q: How do non-VLAN aware bridges handle these frames?



Packet fragmentation
Transparent fragmentation

Q: What can cause packet size limits?

Packet size can be limited by hardware, software, protocols, law, etc.



Q: Can you name a (dis)advantage?

Packet too large for link

Max packet size: 10

Max packet size: 3

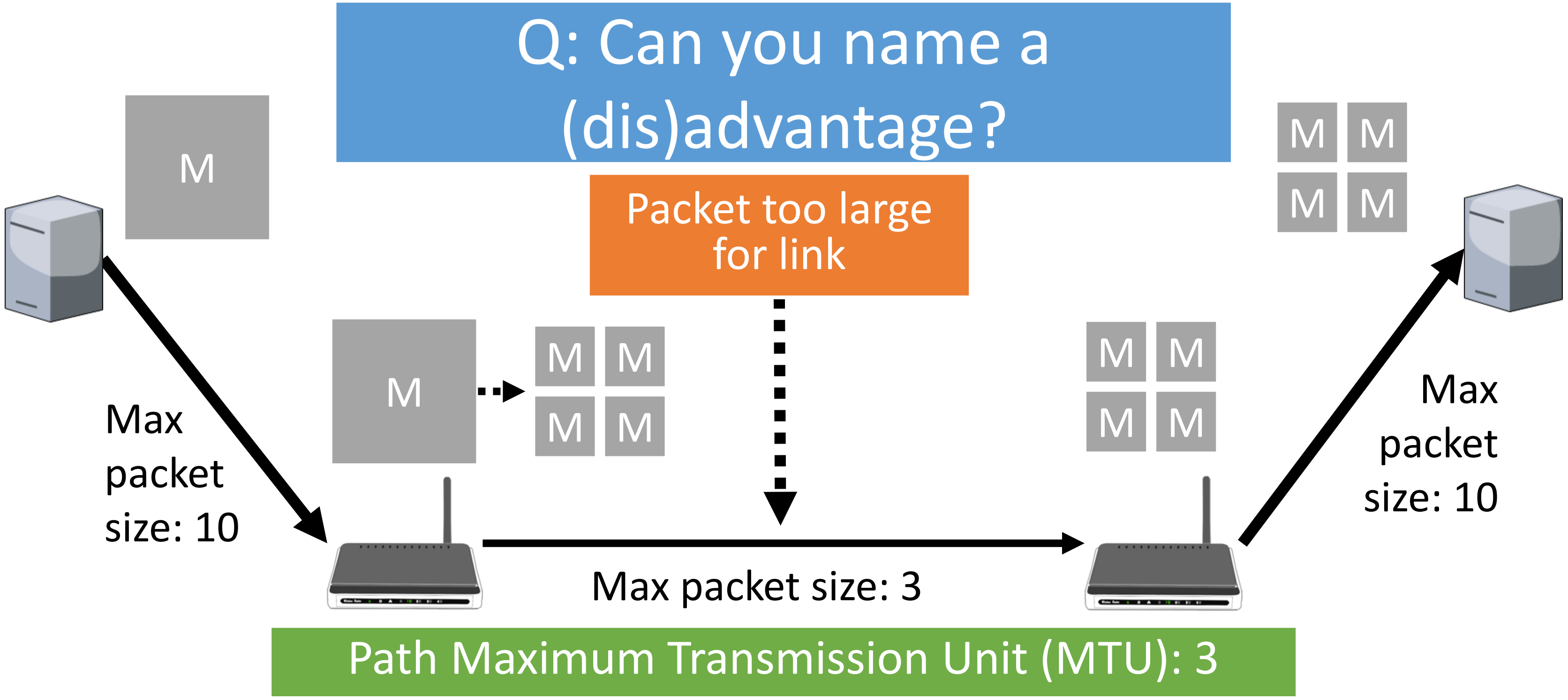
Max packet size: 10

Path Maximum Transmission Unit (MTU): 3

Used in IP!

Packet fragmentation
Nontransparent fragmentation

Packet size can be limited by hardware, software, protocols, law, etc.

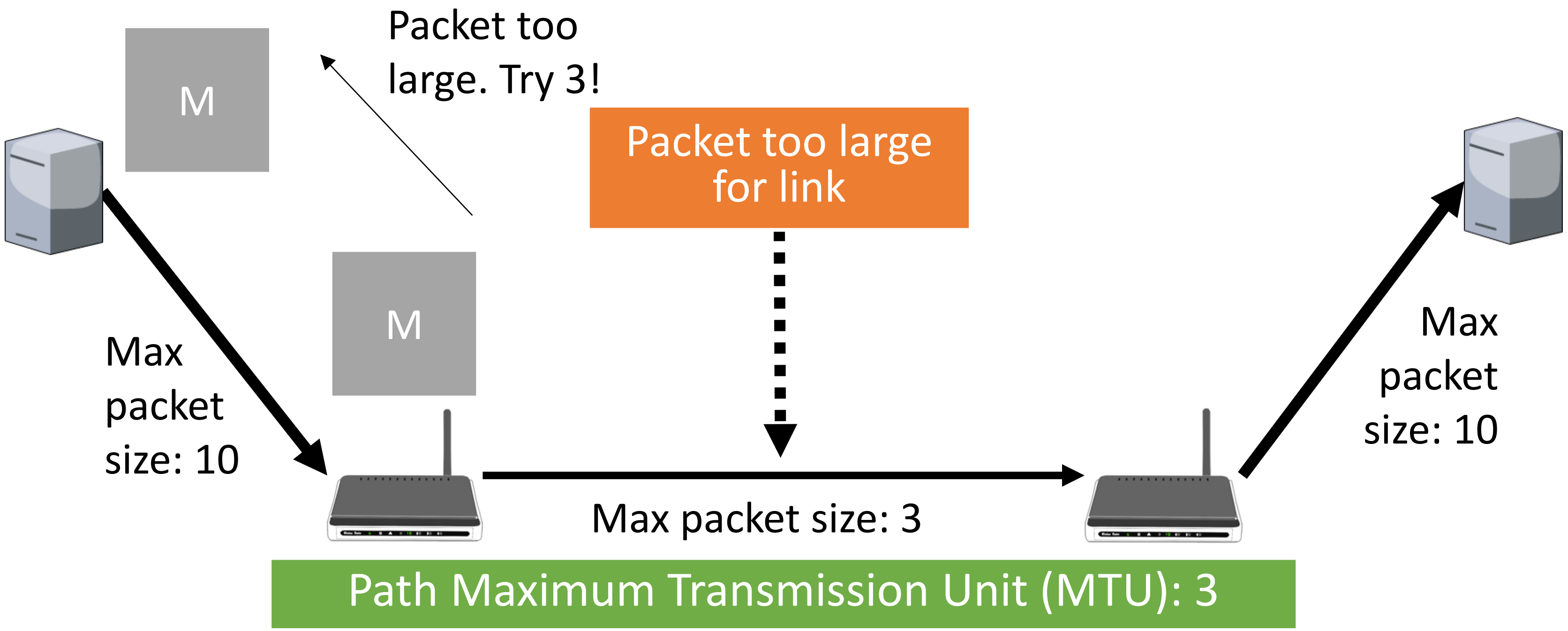

Q: Can you name a (dis)advantage?

Packet too large for link

Path Maximum Transmission Unit (MTU): 3

Avoiding packet fragmentation
MTU discovery

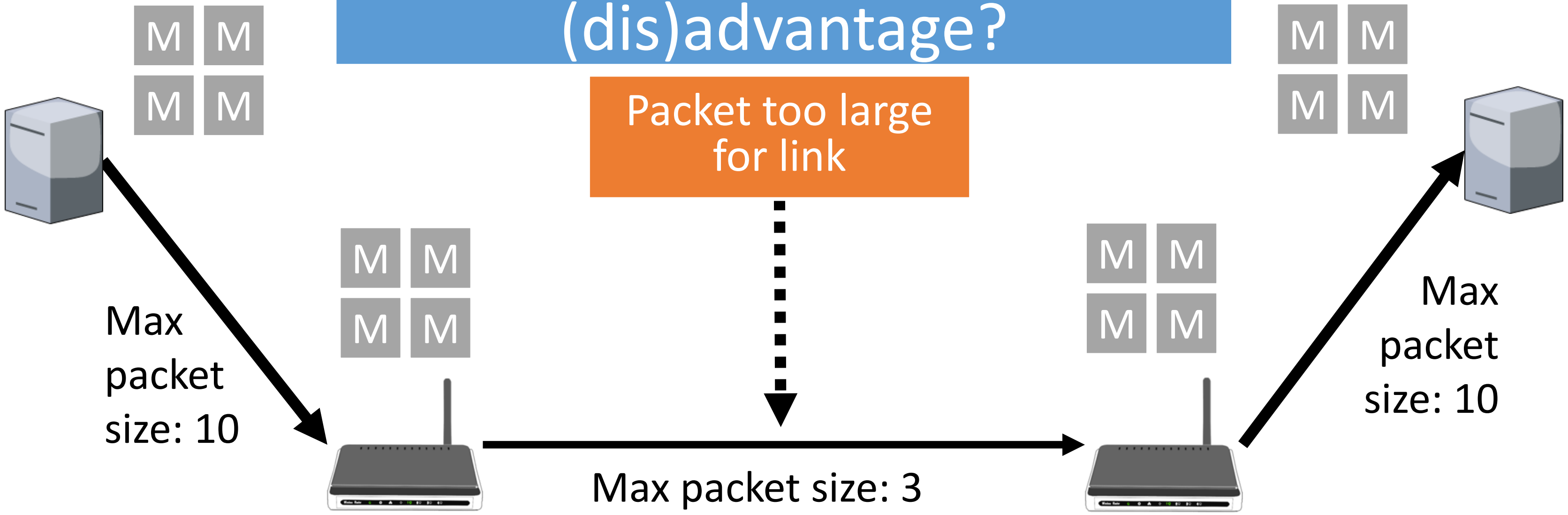
Packet size can be limited by hardware, software, protocols, law, etc.



Used in IP!

Avoiding packet fragmentation
MTU discovery

Packet size can be limited by hardware, software, protocols, law, etc.

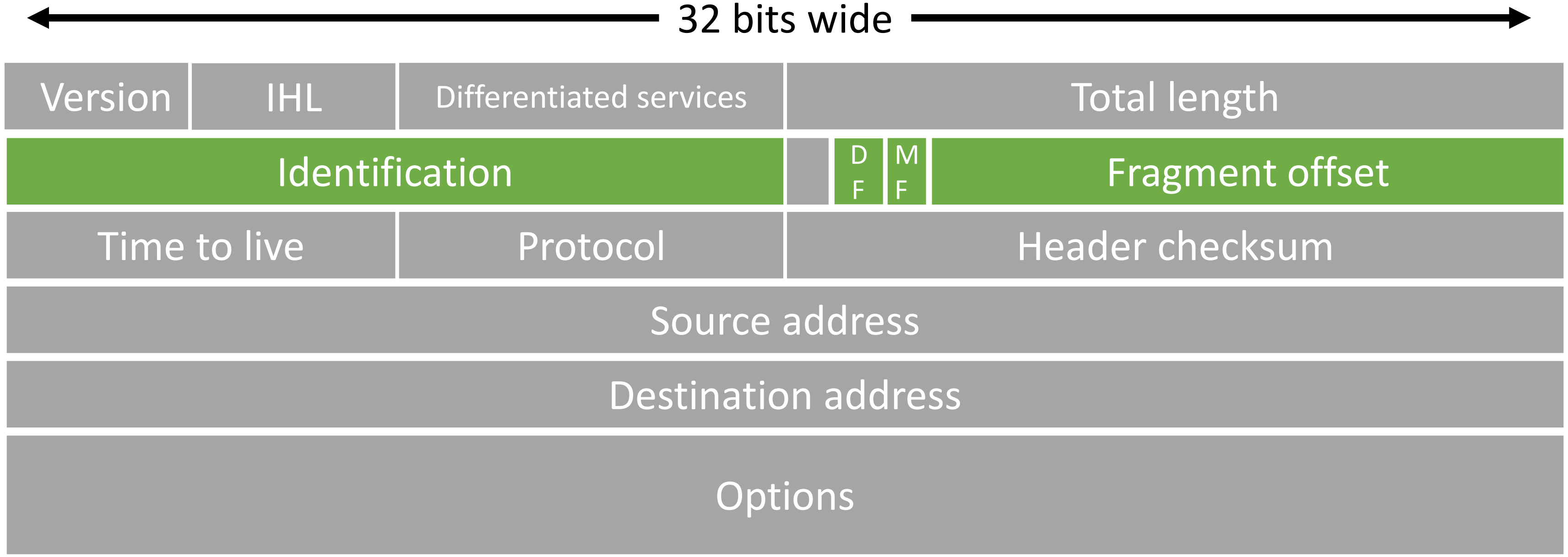

Q: Can you name a (dis)advantage?

Packet too large for link

Path Maximum Transmission Unit (MTU): 3

IP version 4

Frame header



32 bits wide

Version

IHL

Differentiated services

Total length

Identification

DF

MF

Fragment offset

Time to live

Protocol

Header checksum

Source address

Destination address

Options

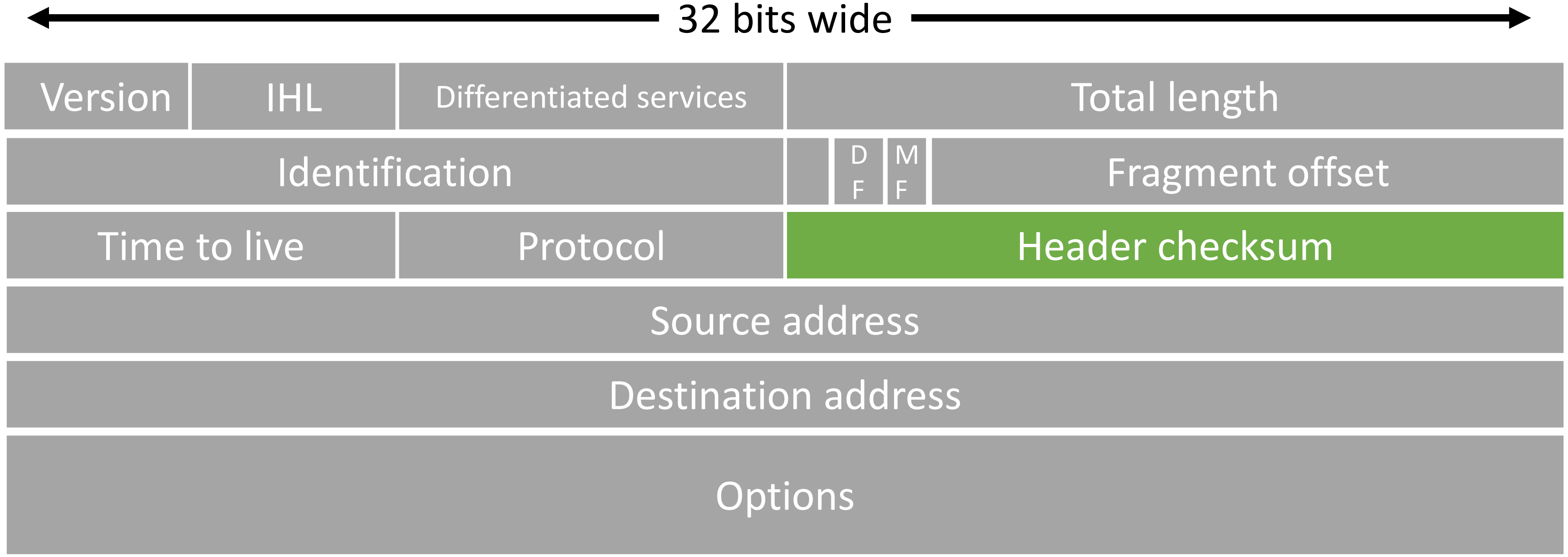
Q: Why have this field?

Challenges Addressed by IPv4 Protocol Design

1. Error detection/correction
2. ~~Preventing permanently looping packets~~
3. Globally identifying computers
4. ~~Carrying packets over links with different size requirements~~

IP version 4

Frame header



32 bits wide

Version | IHL | Differentiated services | Total length

Identification | DF | MF | Fragment offset

Time to live | Protocol | Header checksum

Source address

Destination address

Options

IPv4 does not use a CRC but a checksum.

Computed by adding all 16-bit half-words *in the header*

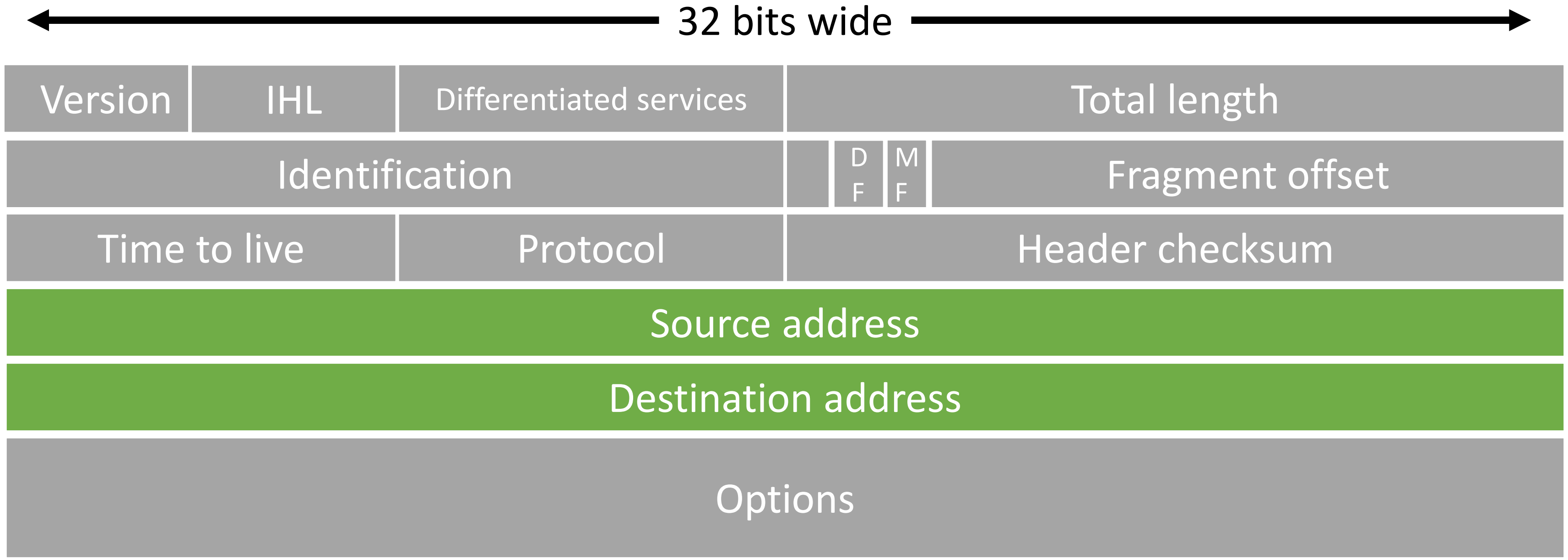
Challenges Addressed by IPv4 Protocol Design

1. ~~Error detection/correction~~
2. ~~Preventing permanently looping packets~~
3. Globally identifying computers
4. ~~Carrying packets over links with different size requirements~~

IP version 4

Q: What service does IP ***not*** provide?

Frame header



32 bits wide

Version | IHL | Differentiated services | Total length

Identification | D F | M F | Fragment offset

Time to live | Protocol | Header checksum

Source address

Destination address

Options

Q: Why have this field?

IPv4 addresses

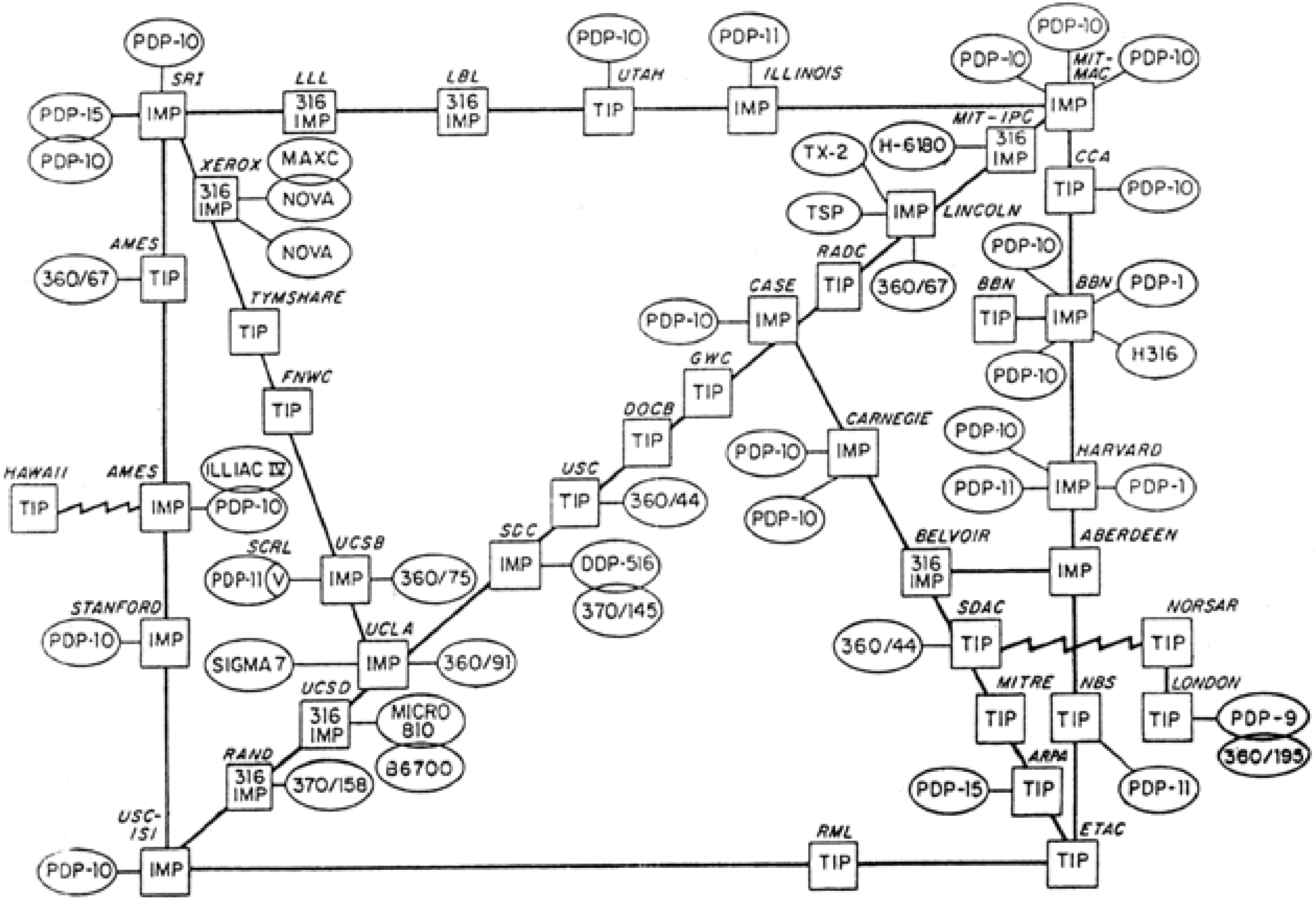
IPv4 uses 32-bit addresses.
Written in ***dotted decimal notation***.
Address 0x80D00297 is written as 128.208.2.151.

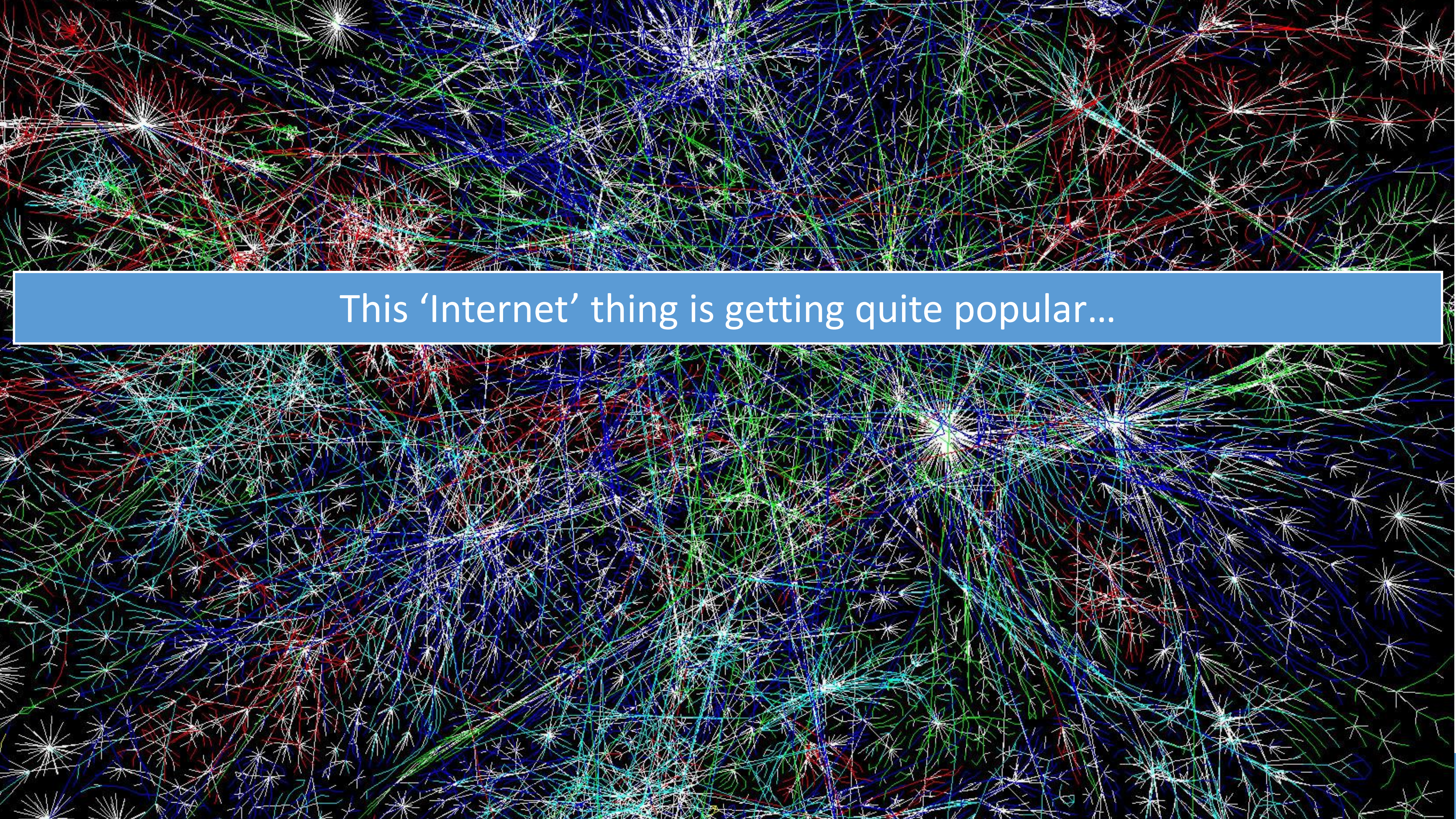
32-bit address gives $2^{32} > 4$ billion addresses.

Q: How to route packets to these addresses with latencies in the order of milliseconds?

Reduce routing table sizes using ***hierarchical routing***!

ARPA NETWORK, LOGICAL MAP, SEPTEMBER 1973



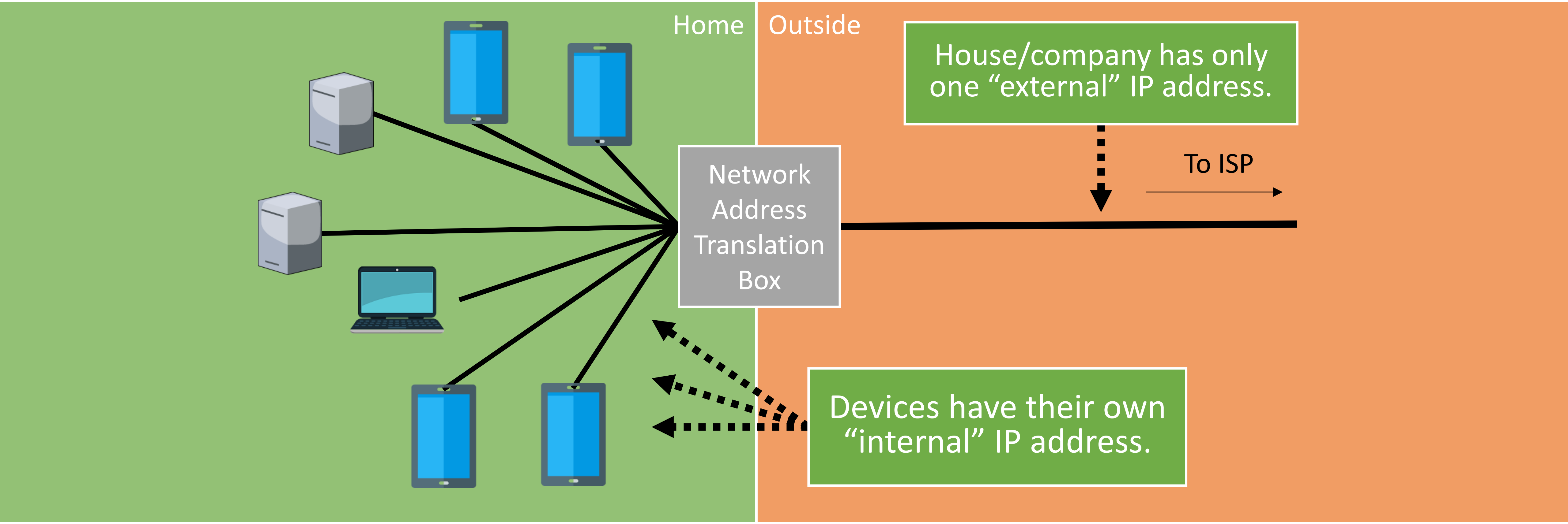


This ‘Internet’ thing is getting quite popular…

Network Address Translation (NAT)

Q: No headers left in IP header. How to implement this?

How to let everybody go online with only 2^{32} addresses?



Network Address Translation Box

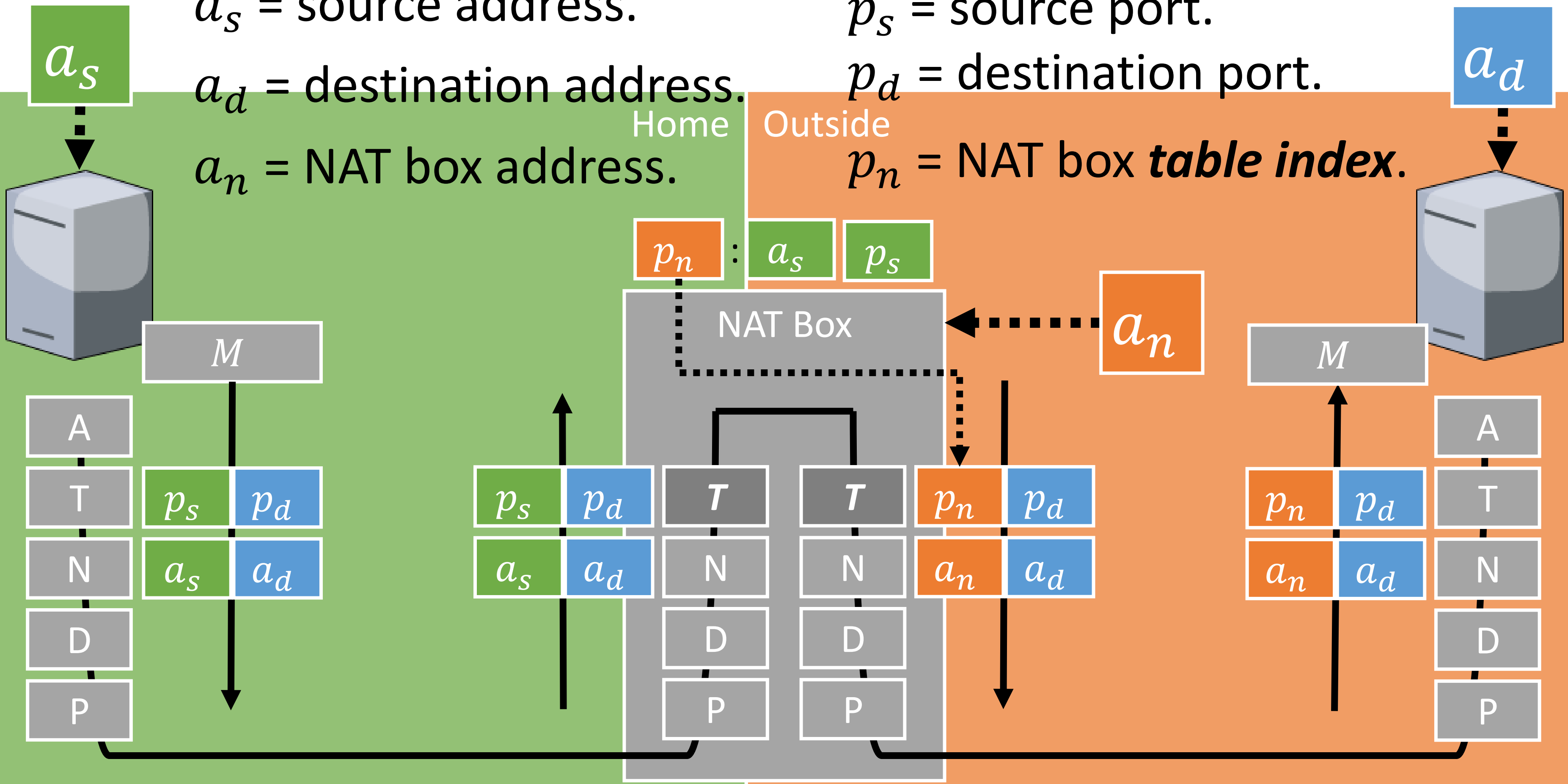
House/company has only one “external” IP address.

Devices have their own “internal” IP address.

Copyright Jesse Donkervliet 2024

Q: How to send something back to a_s?

Network Address Translation (NAT)



a_s = source address.

a_d = destination address.

a_n = NAT box address.

p_s = source port.

p_d = destination port.

p_n = NAT box ***table index***.

Q: Objections to this approach?

Challenges Addressed by IPv4 Protocol Design

1. ~~Error detection/correction~~
2. ~~Preventing permanently looping packets~~
3. ~~Globally identifying computers~~
4. ~~Carrying packets over links with different size requirements~~

IP version 6

Multiple improvements over IPv4.

1. ***Many*** more addresses!
2. Simplified header – improves bandwidth/latency.
3. Easier to add ***options*** in the header.
4. Improved security support. ◄····· Backported to IPv4

IP version 6

IP version 4

Address size:
32 bits.

Dotted decimal notation:
192.31.20.46

Number of addresses:

$$2^{32} = 4,294,967,296$$

IP version 6

Address size:
128 bits.

Hexadecimal notation:
8000::123:4567:89AB:CDEF

Number of addresses:

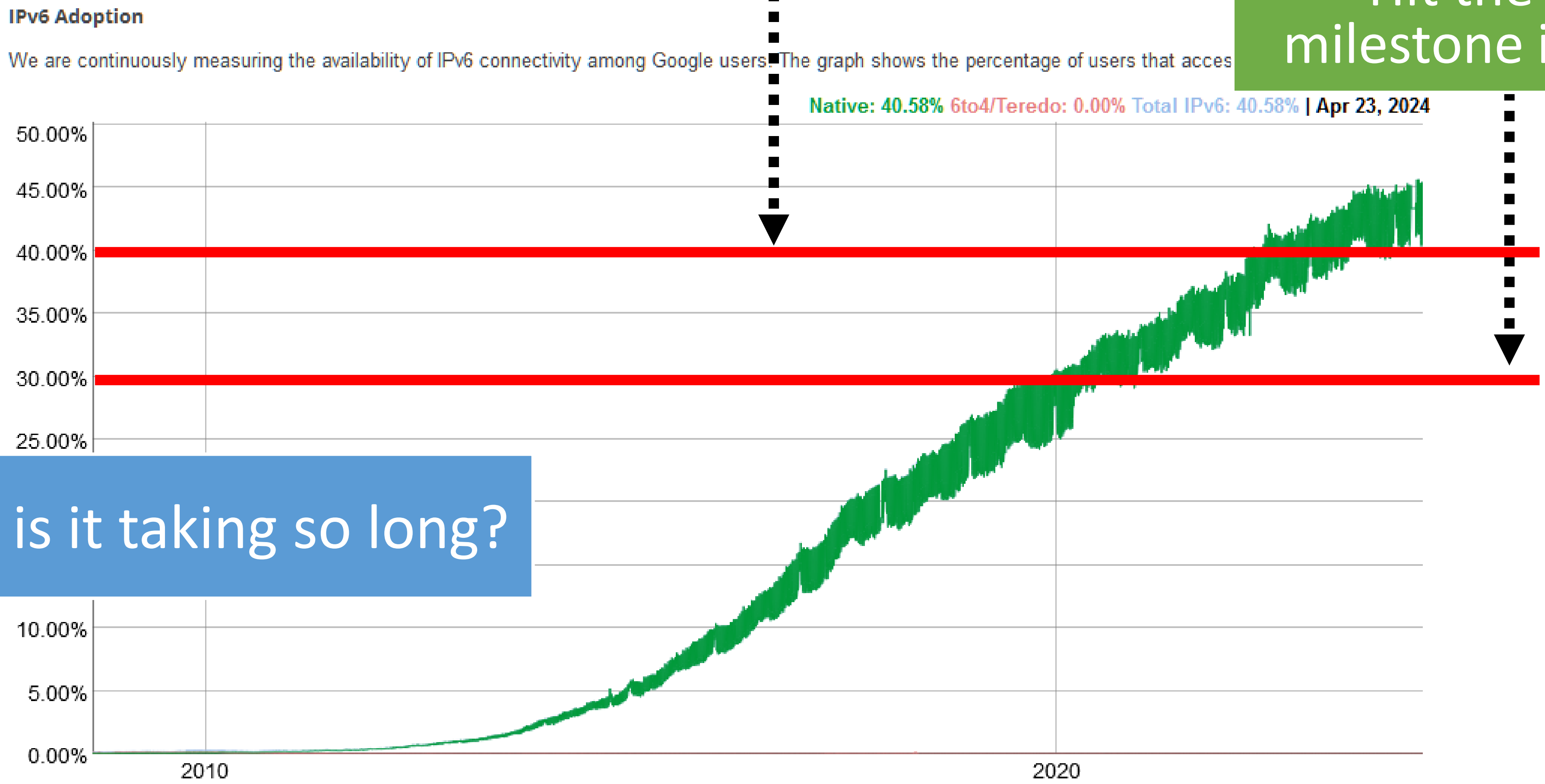
$$2^{128} =$$

$$340,282,366,920,938,463,463,374,607,431,768,211,456$$

That’s a lot!

Hit the 40% milestone in 2022!

Hit the 30% milestone in 2020!



IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that acces

Q: Why is it taking so long?

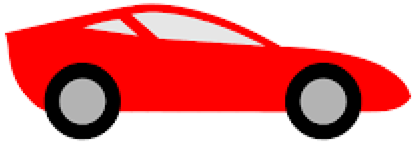
Connecting Networks with Different Protocols

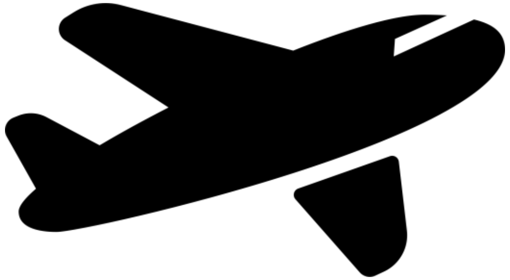
If source and destination networks use different protocols, they cannot communicate.

Network A:
Uses ‘cars’ protocol.

Network B:
Uses ‘boats’ protocol.

Network C:
Uses ‘planes’ protocol.



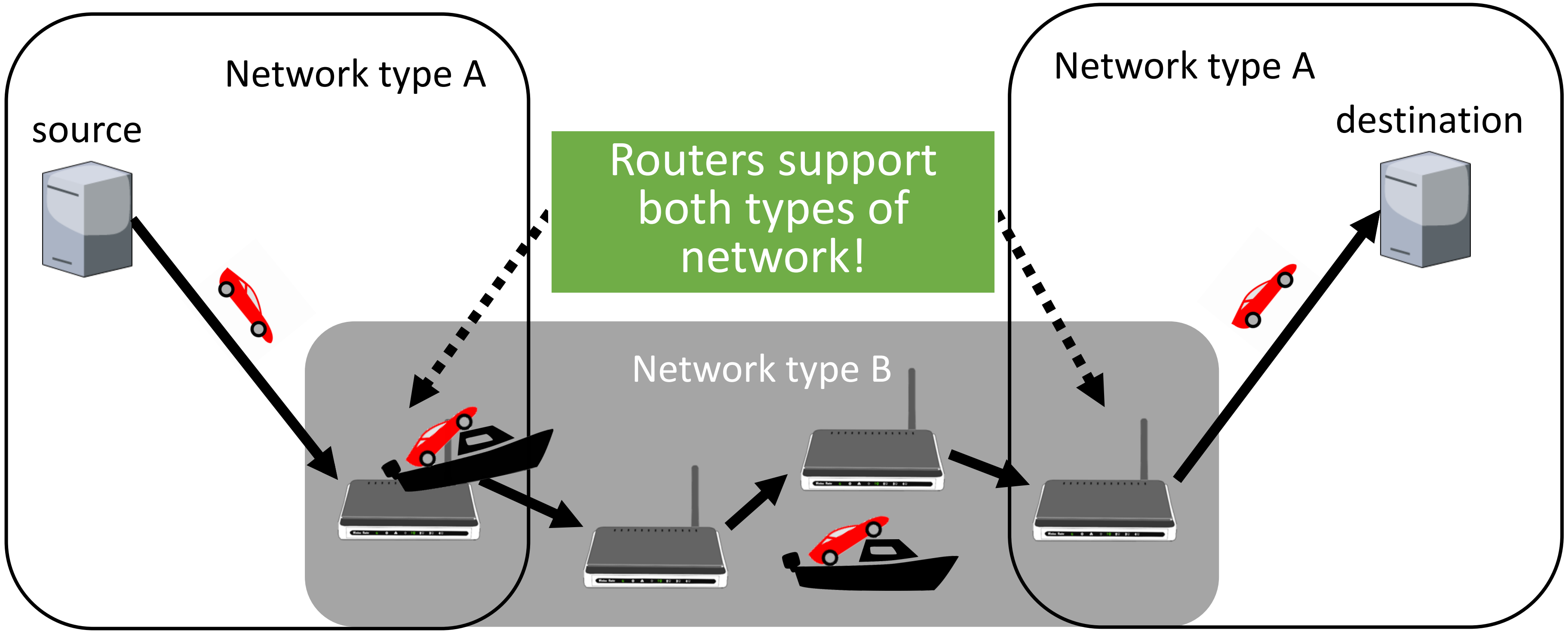


Tunneling

Used to route IPv6 packets over IPv4 networks

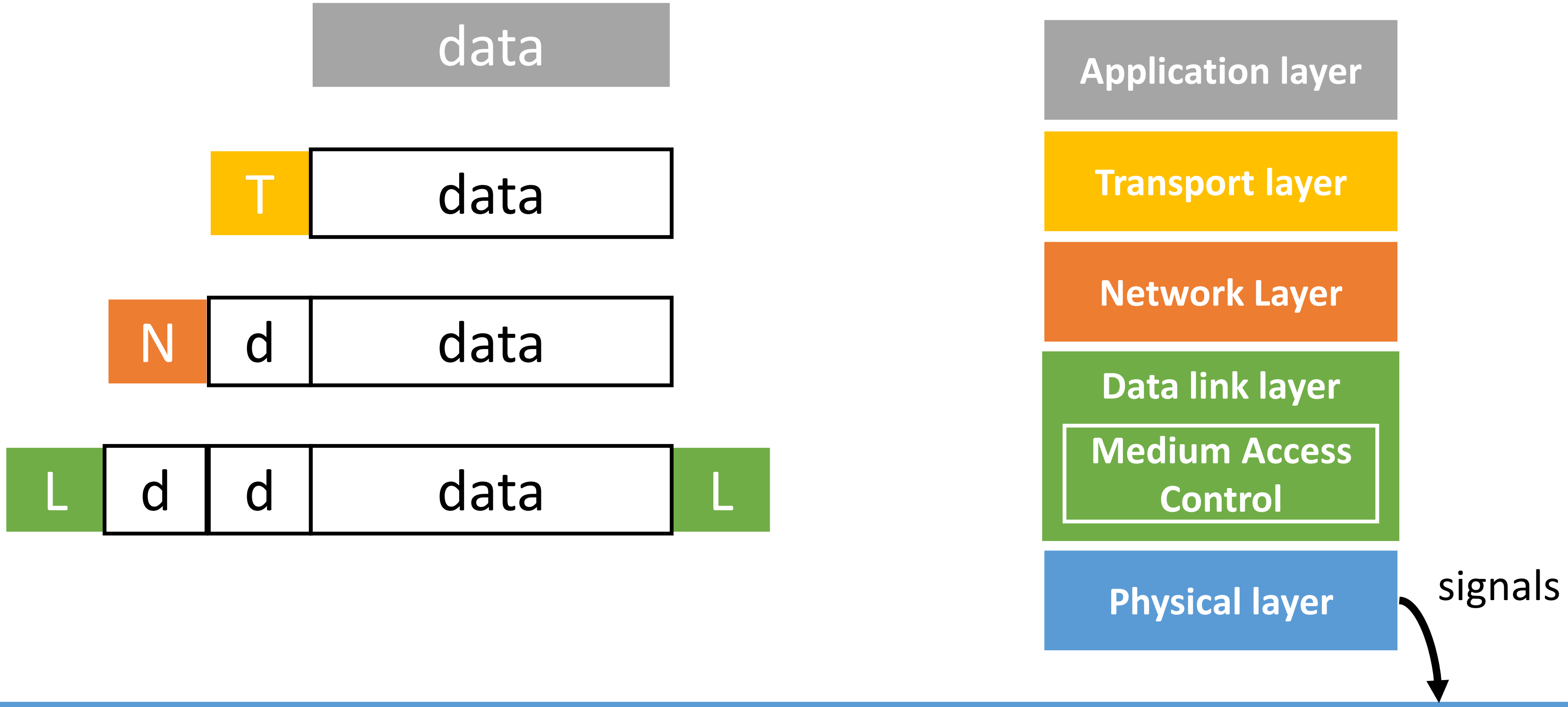
Q: Can you name a (dis)advantage?

If an intermediate network uses different protocols, they can communicate by tunneling.



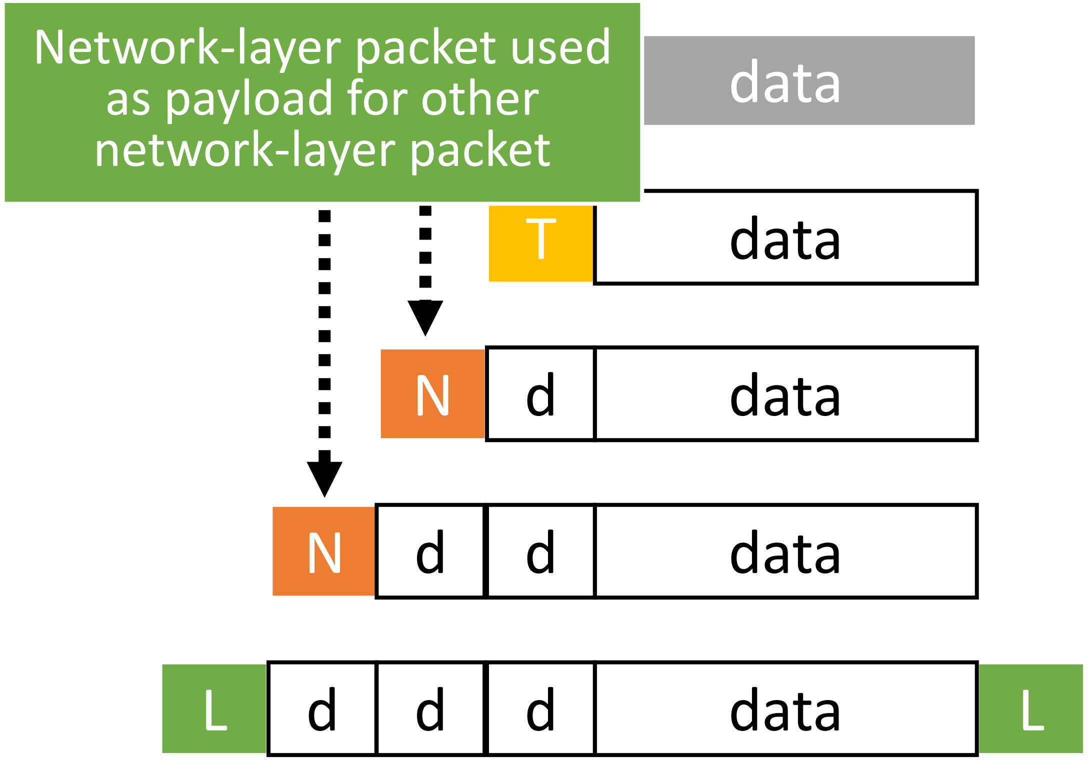
Business as usual
Packets in packets in packets in …

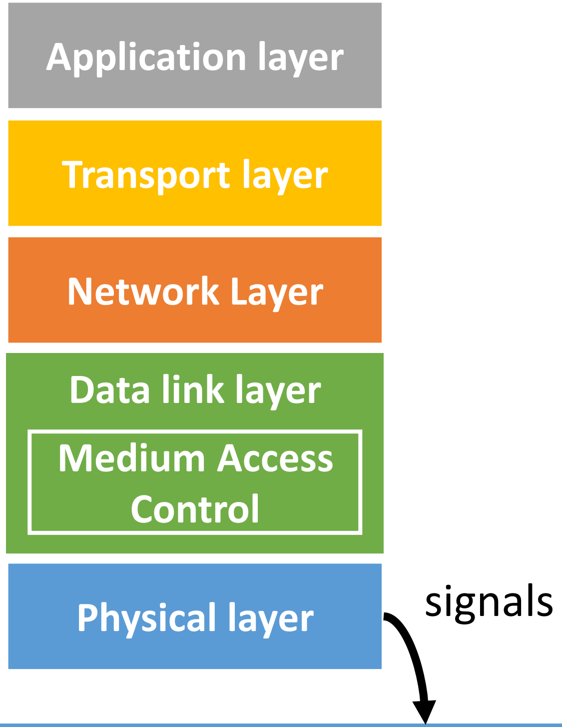
Data wrapped in headers from multiple networking layers.



Tunneling
Packets in packets in packets in …

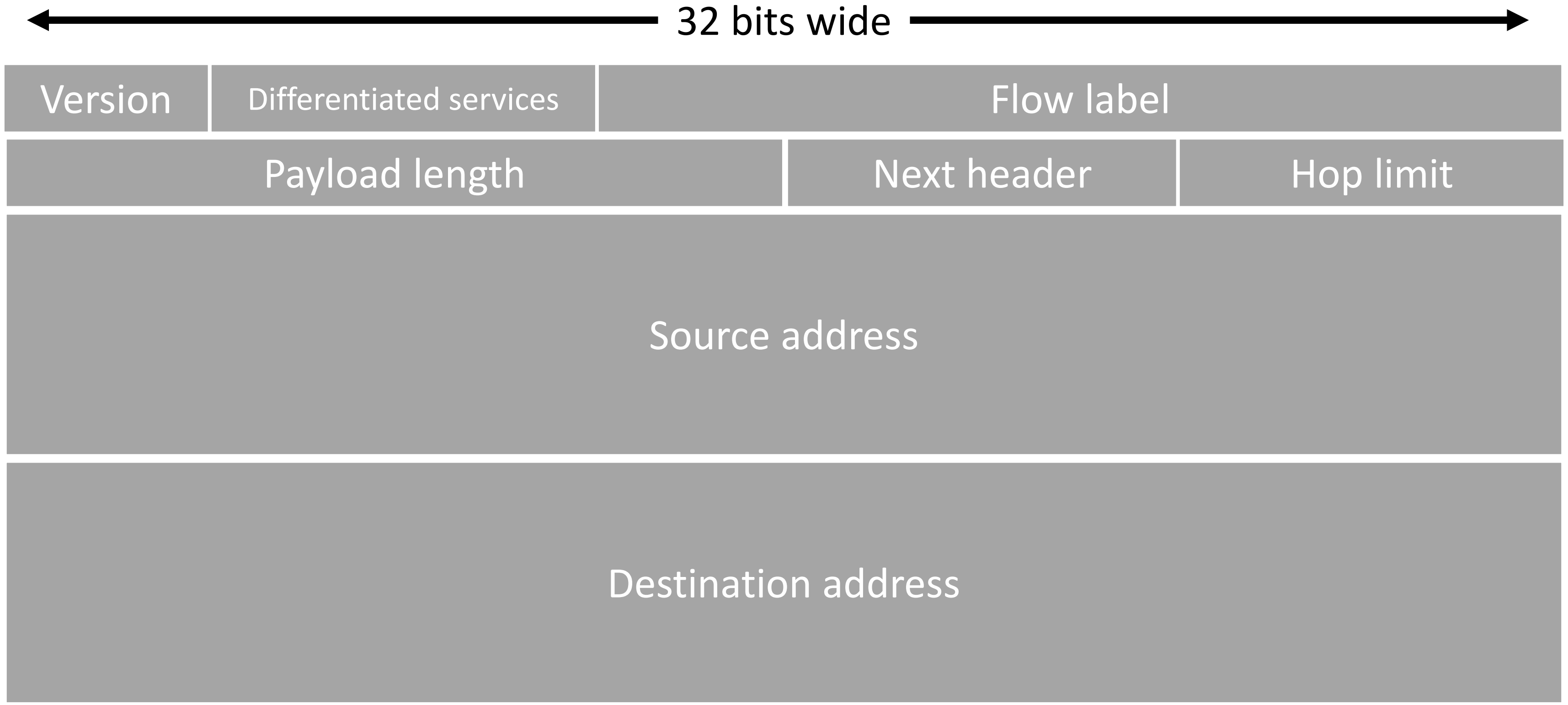
Data wrapped in headers from multiple networking layers.





IP version 6

Frame header



32 bits wide

Version | Differentiated services | Flow label

Payload length | Next header | Hop limit

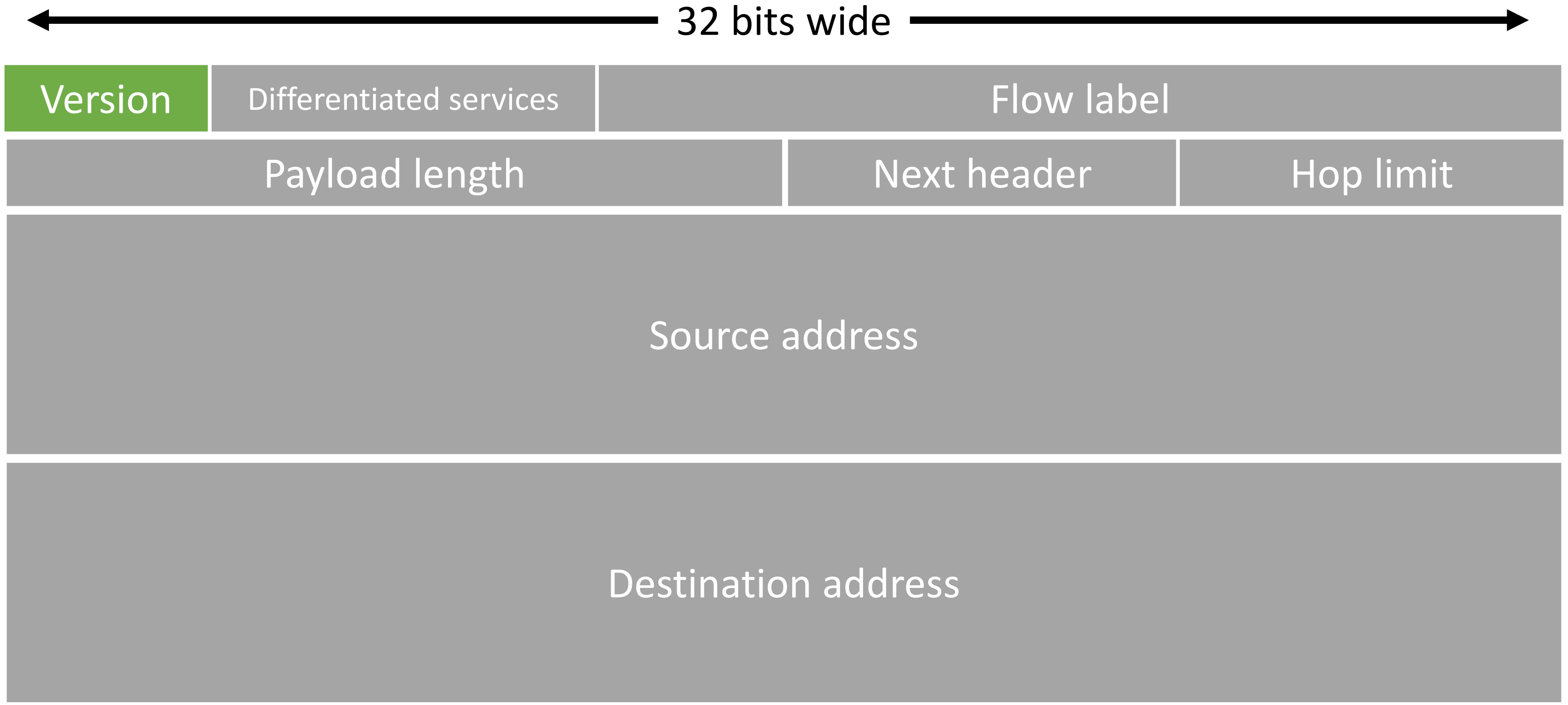
Source address

Destination address

IP version 6

Value 0x06 to indicate IP version 6

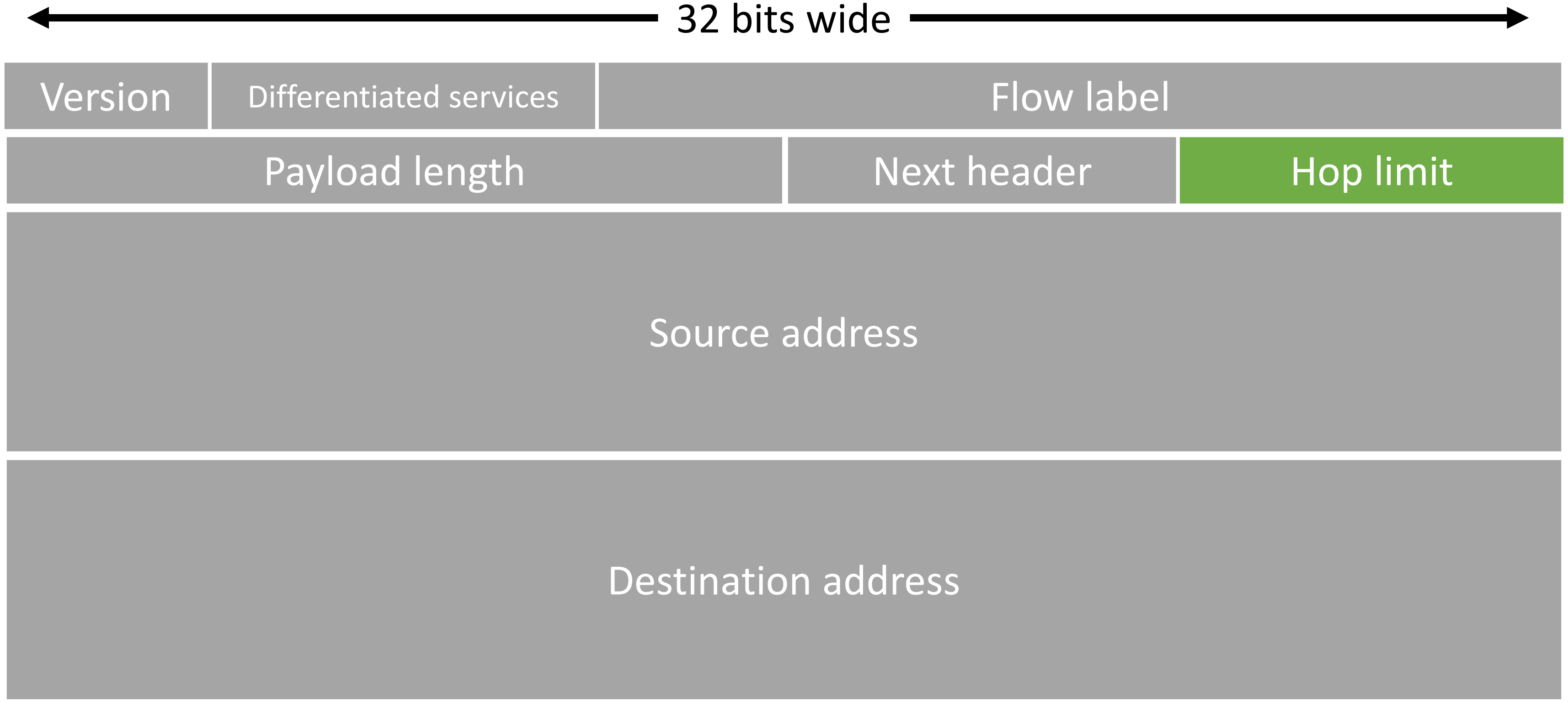
Frame header



IP version 6

“Time to live” renamed to “Hop limit”

Frame header



32 bits wide

Version | Differentiated services | Flow label

Payload length | Next header | Hop limit

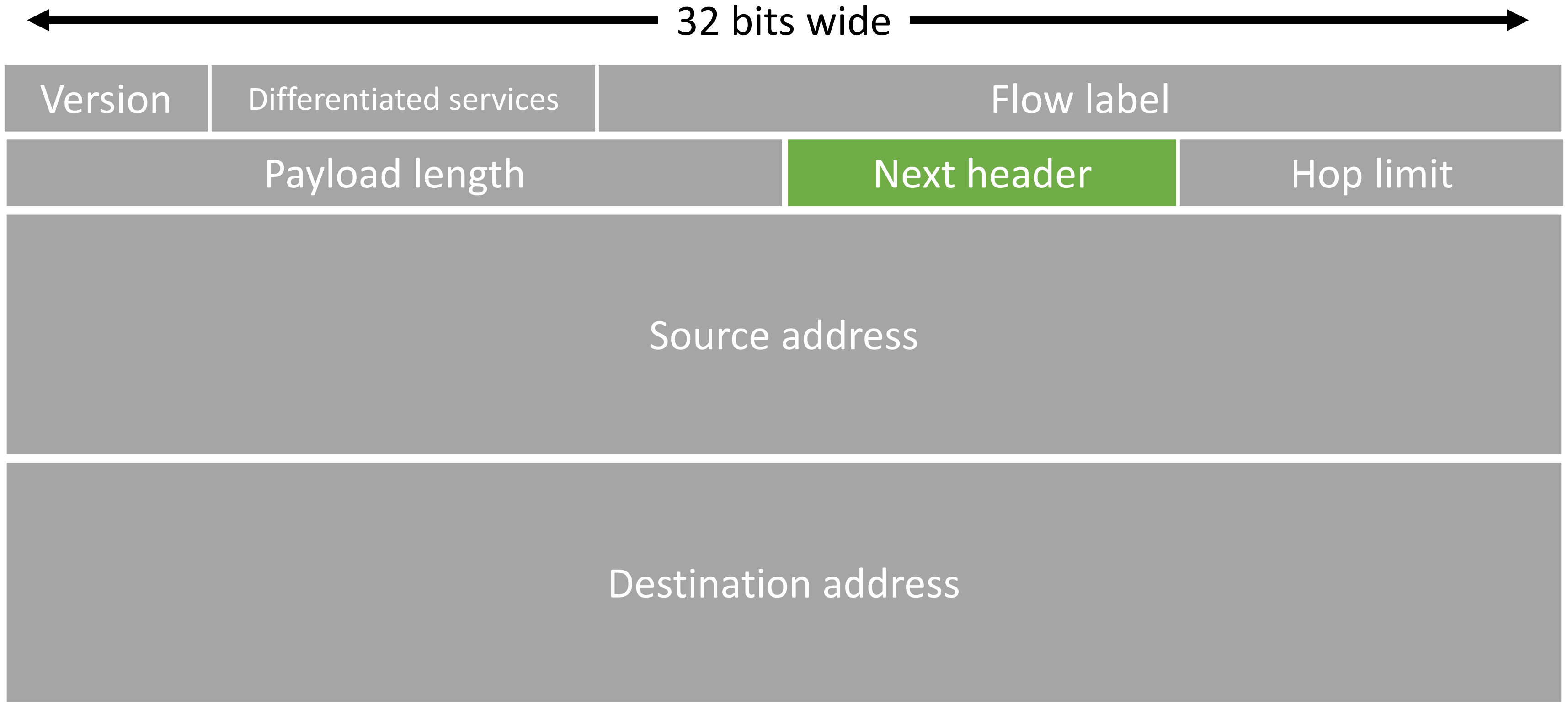
Source address

Destination address

IP version 6

Specifies transport layer protocol or extension header

Frame header



Addressing the Problem of Too Many Addresses to Route

Managing the size of routing tables

Routing algorithms can calculate routes to prefixes, instead of to every individual address

Internet Protocol Prefixes and Subnets

Vrije Universiteit given a *prefix*. E.g., all IP addresses that match ***37.60*.x.y**.

Address starts with 37.60?
If yes, route to VU.

Example address: 37.60.194.64.

00100101.00111100.11000010.01000000

Network

Host

Prefixes handed out by single organization: ICANN

Organizations can further subdivide their prefix to create *subnets*

16 bits used by network

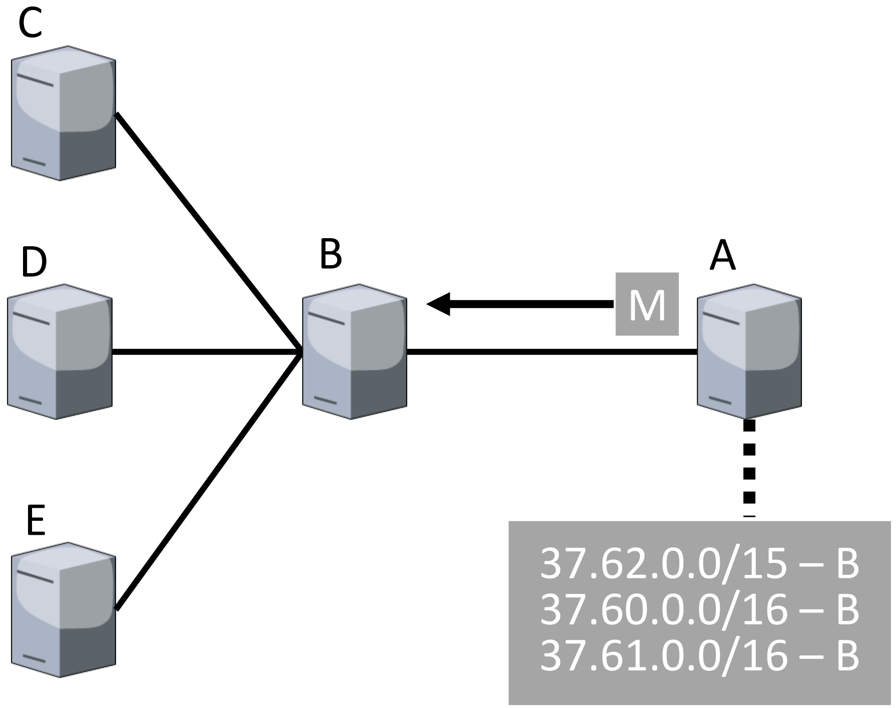
Prefix: 37.60.0.0/16
Subnet mask: 11111111.11111111.00000000.00000000

Internet Protocol - CIDR
Classless InterDomain Routing

TU Delft
37.62.0.0/15
00100101.00111110.00000000.00000000

VU
37.60.0.0/16
00100101.00111100.00000000.00000000

UvA
37.61.0.0/16
00100101.00111101.00000000.00000000



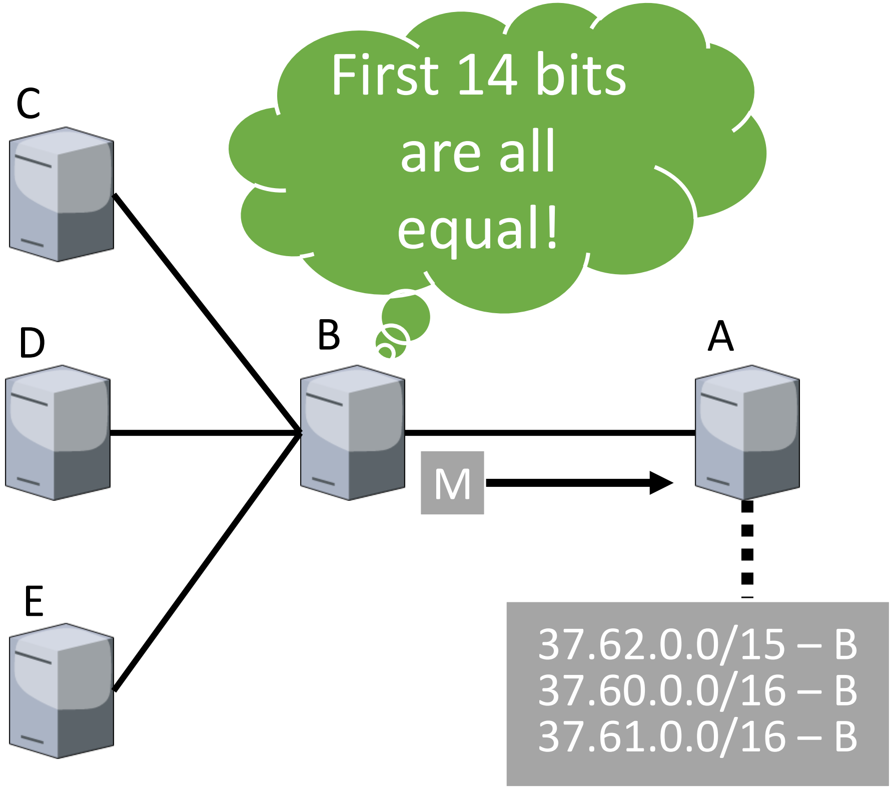
37.62.0.0/15 – B
37.60.0.0/16 – B
37.61.0.0/16 – B

Internet Protocol - CIDR
Classless InterDomain Routing

TU Delft
37.62.0.0/15
00100101.00111110.00000000.00000000

VU
37.60.0.0/16
00100101.00111100.00000000.00000000

UvA
37.61.0.0/16
00100101.00111101.00000000.00000000



First 14 bits are all equal!

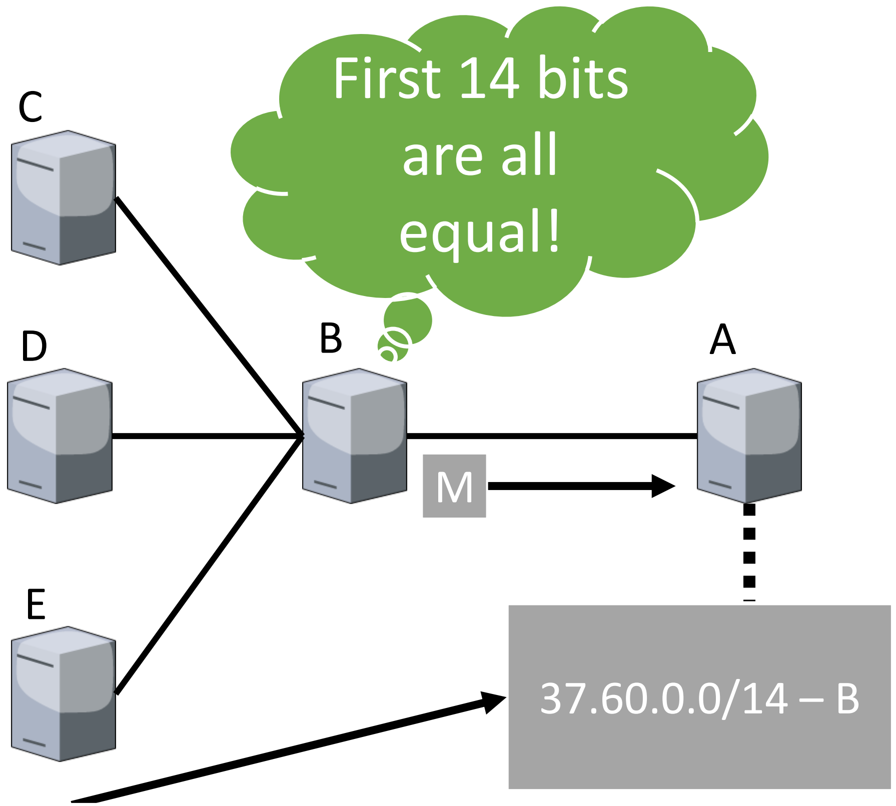
37.62.0.0/15 – B
37.60.0.0/16 – B
37.61.0.0/16 – B

Internet Protocol - CIDR
Classless InterDomain Routing

TU Delft
37.62.0.0/15
`00100101.00111110.00000000.00000000`

VU
37.60.0.0/16
`00100101.00111100.00000000.00000000`

UvA
37.61.0.0/16
`00100101.00111101.00000000.00000000`



Reducing routing table size through ***route aggregation***

Longest Matching Prefix

Consider the following routing table:

Prefix	Port	Binary
✖ 137.70.32.192/26	A	10001001.01000110.00100000.11000000
✓ 137.70.32.0/20	B	10001001.01000110.00100000.00000000
137.64.0.0/10	C	10001001.01000000.00000000.00000000
0.0.0.0/0	D	00000000.00000000.00000000.00000000
137.70.32.128		10001001.01000110.00100000.10000000

An incoming IP packet carries the destination address 137.70.32.128. On which port is this packet forwarded? Assume that the router uses the *longest matching prefix*.

Internet Control Message Protocol (ICMP)

Network Layer Protocol

Internet Control Message Protocol (ICMP)

If something goes wrong, ***routers*** send these messages to ***senders***.

Some examples:

1. Destination unreachable
2. Time exceeded
3. “Echo” and “echo reply”
4. Router advertisement/solicitation
5. Packet needs fragmentation / packet too big

Used by the program `traceroute`

Used by the program `ping`

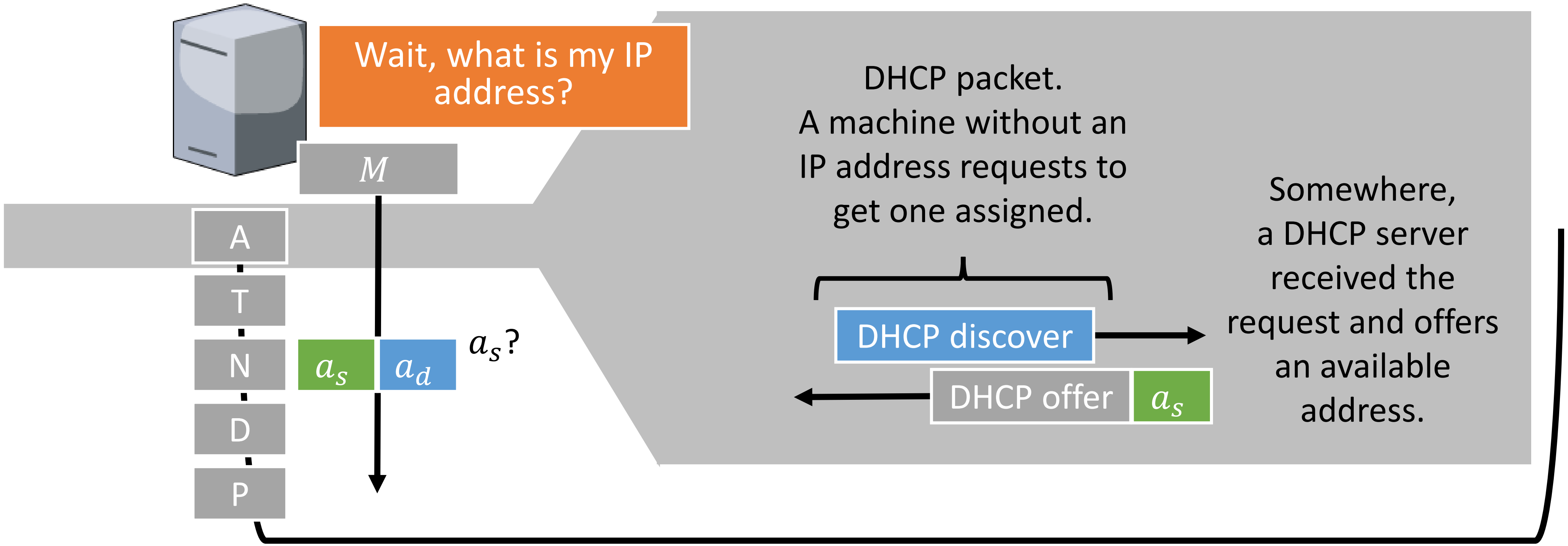
Dynamic Host Configuration (DHCP)

Network Layer Protocol

Dynamic Host Configuration Protocol (DHCP)

MAC addresses are built into NICs. But network addresses are not.

Used to configure other settings such as: network mask, addresses of default gateway, DNS, time servers, etc.



Wait, what is my IP address?

DHCP packet. A machine without an IP address requests to get one assigned.

Somewhere, a DHCP server received the request and offers an available address.

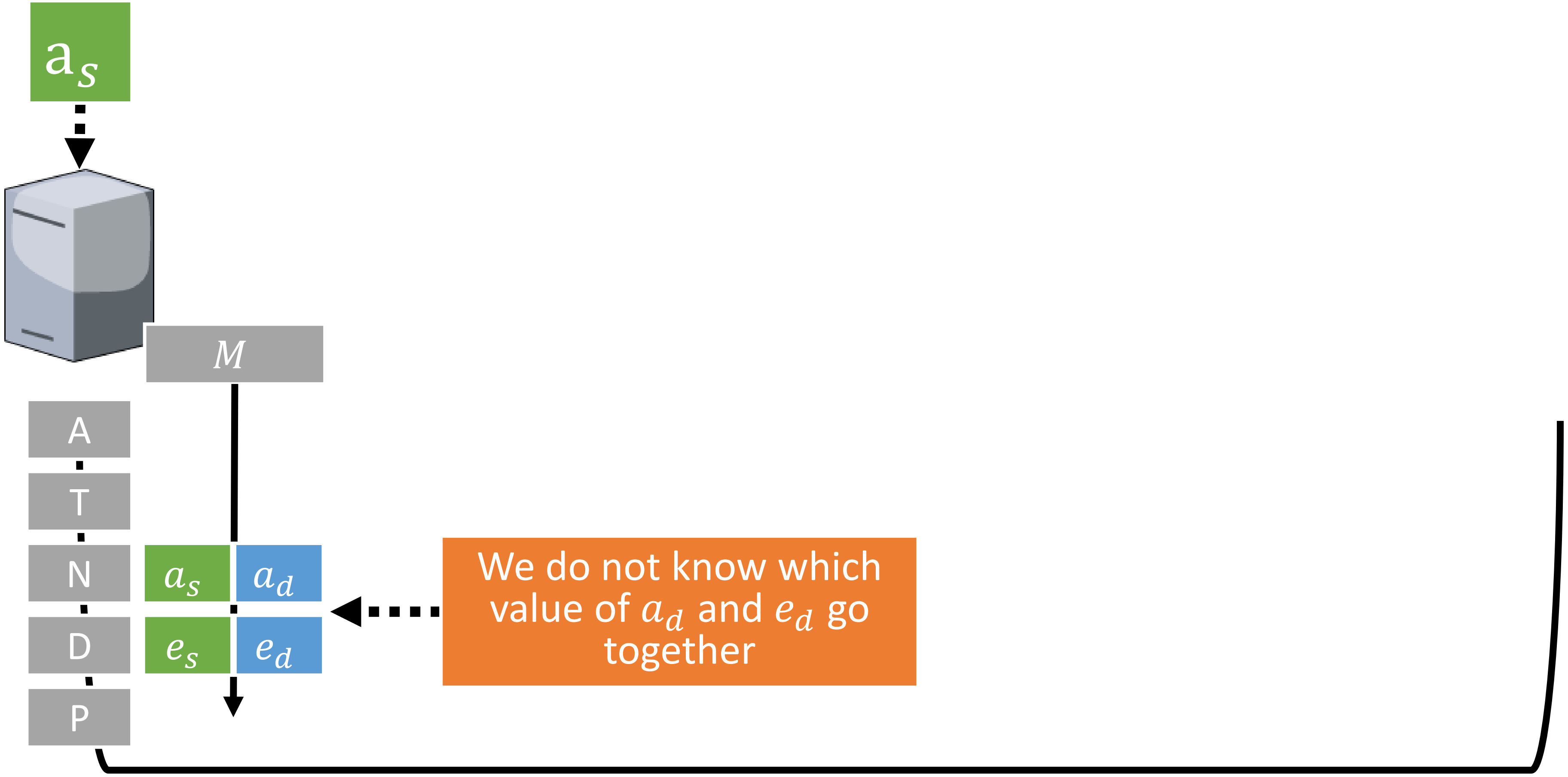
Q: How to send DHCP offer back to machine without an address?

Address Resolution Protocol (ARP)

Network Layer Protocol

Address Resolution Protocol (ARP)

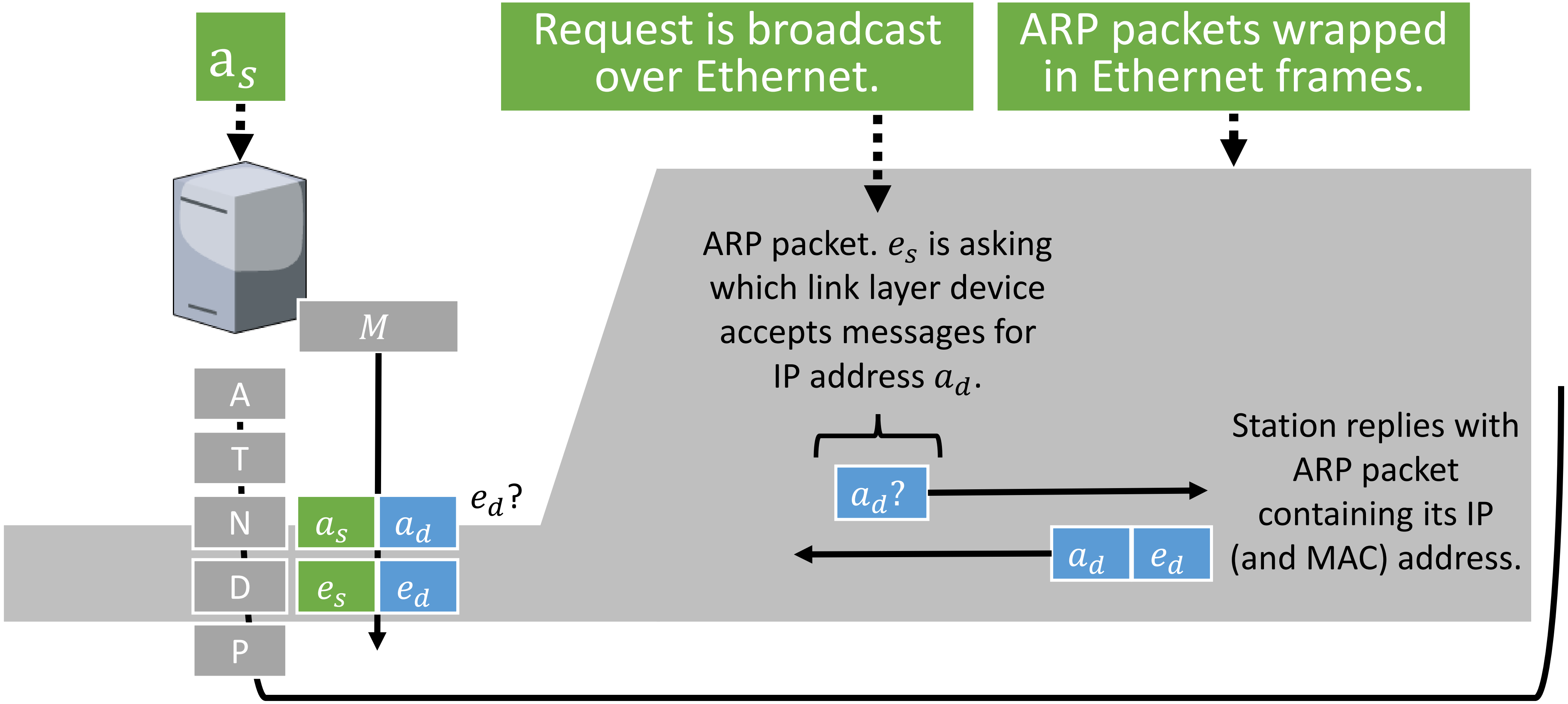
Q: Problems with this approach?



We do not know which value of a_d and e_d go together

Address Resolution Protocol (ARP)

Q: Problems with this approach?



Request is broadcast over Ethernet.

ARP packets wrapped in Ethernet frames.

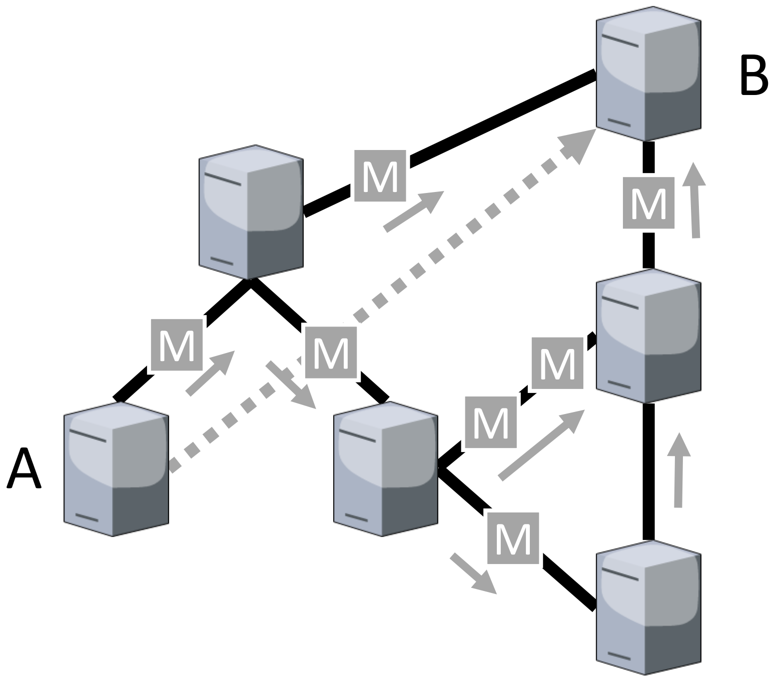
ARP packet. e_s is asking which link layer device accepts messages for IP address a_d.

Station replies with ARP packet containing its IP (and MAC) address.

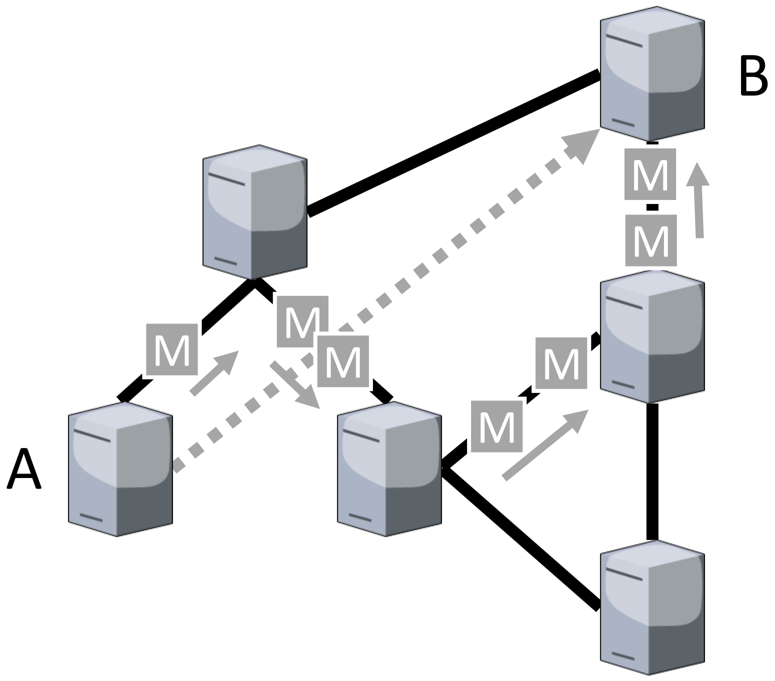
Network-Layer Resource Allocation

Two Main Service Types

Connectionless service:



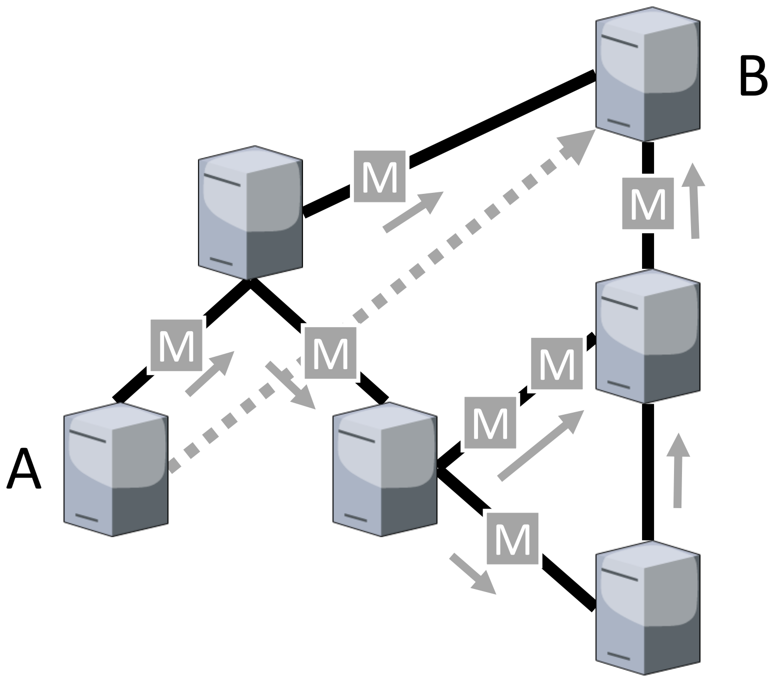
Connection-oriented service:



Connectionless Service
Datagrams

Routers use routing algorithms to decide where to send each packet *individually*.

Used by the Internet Protocol (IP).

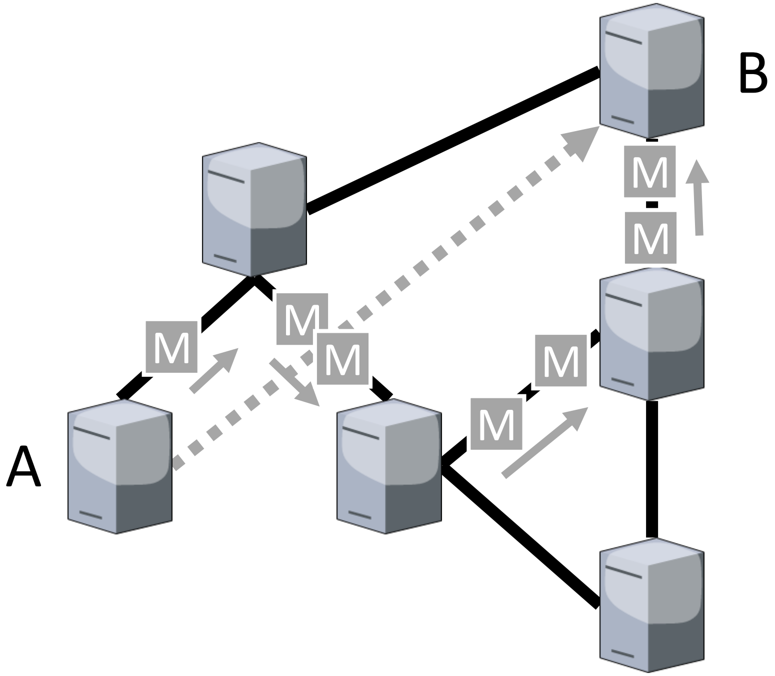


Connection-Oriented Service
Virtual Circuits

Decide fixed route during connection setup.

All packets part of the connection follow this route.

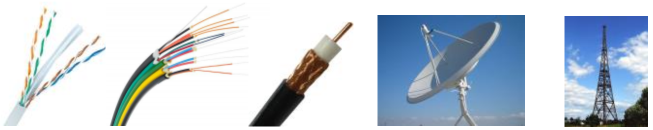
ISPs can use this on top of IP.



Q: What kind of parameters are negotiated during connection setup?

The “best” service depends on your use case

Service comparison



Datagrams

No setup required

Router failures have low impact

Packets contain full addressing information. Routers are stateless.

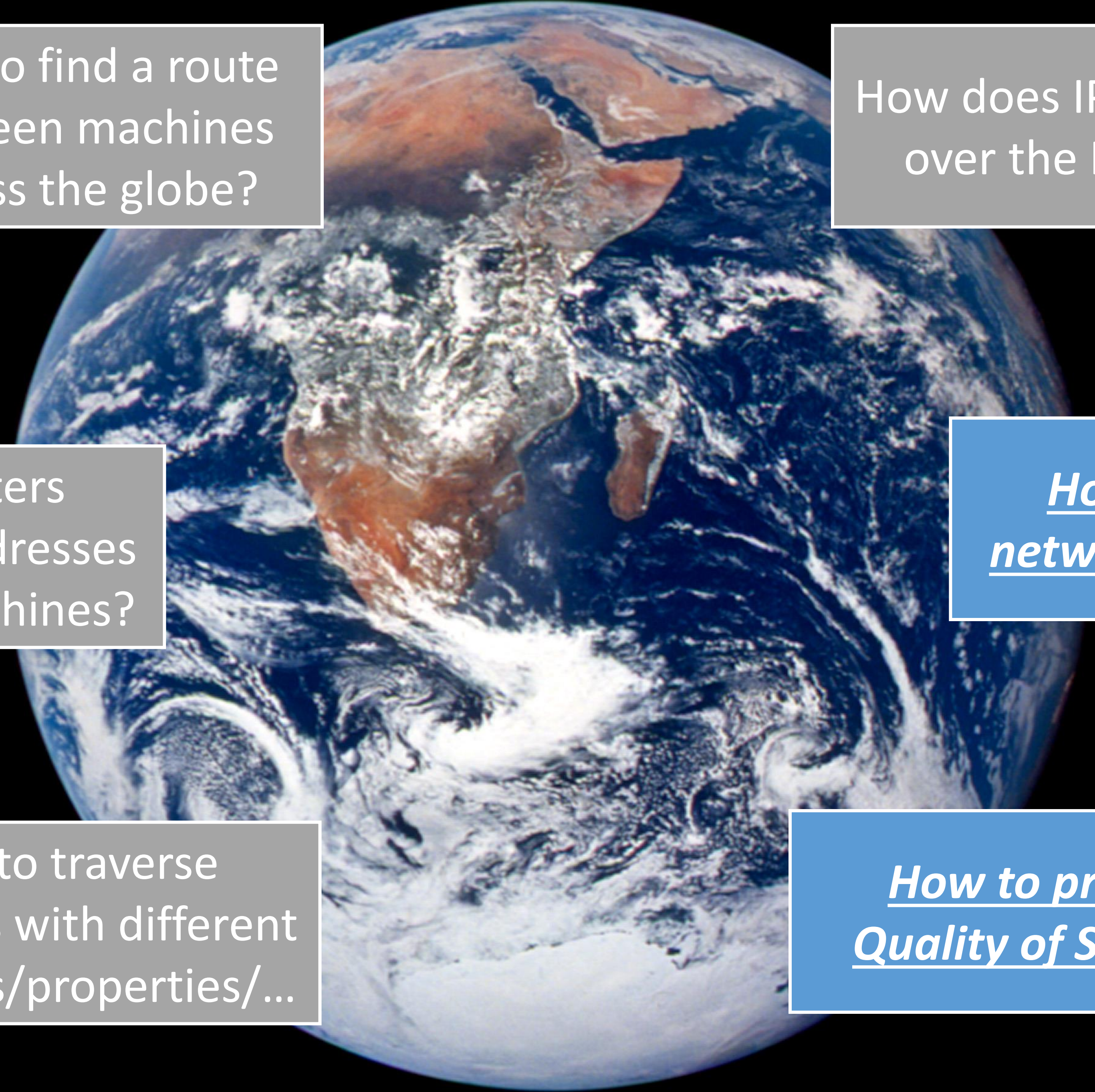
Virtual Circuits

Easy congestion control

Easy Quality of Service guarantees

Packets contain VC number. Routers keep track of active VCs.

Complexity moved. No free lunch



How to find a route between machines across the globe?

How does IP carry data over the Internet?

How do routers manage the addresses of all these machines?

How to prevent network congestion?

How to traverse networks with different protocols/properties/…

How to provide Quality of Service?

Image source: NASA

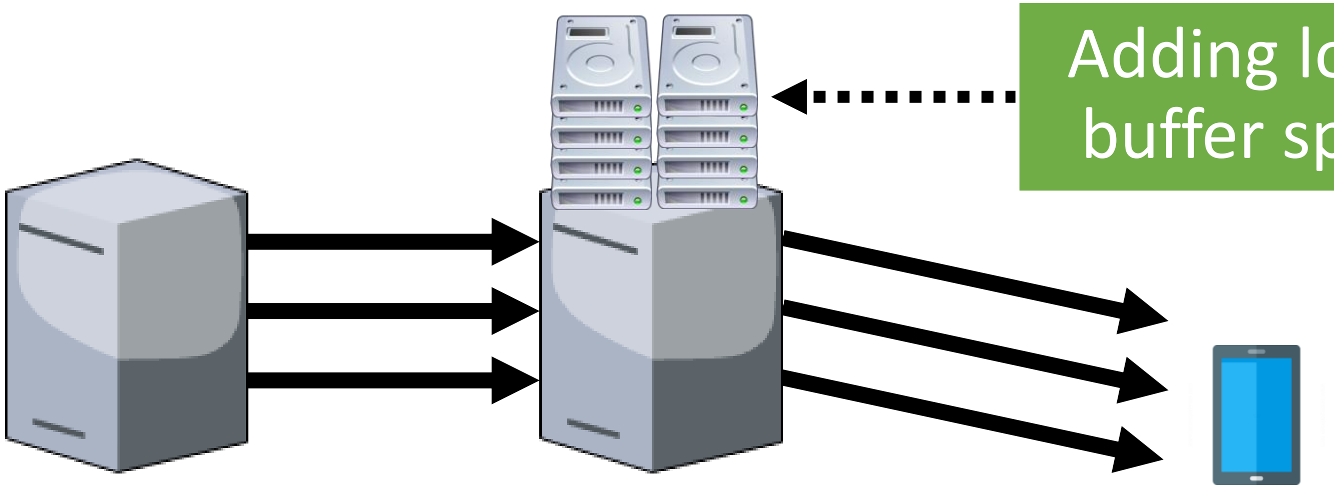
Congestion control

Preventing traffic jams

Looking back on flow control

Mechanism in data link layer.

Makes sure a sender does not send information faster than a receiver can accept.



Adding lots of buffer space.

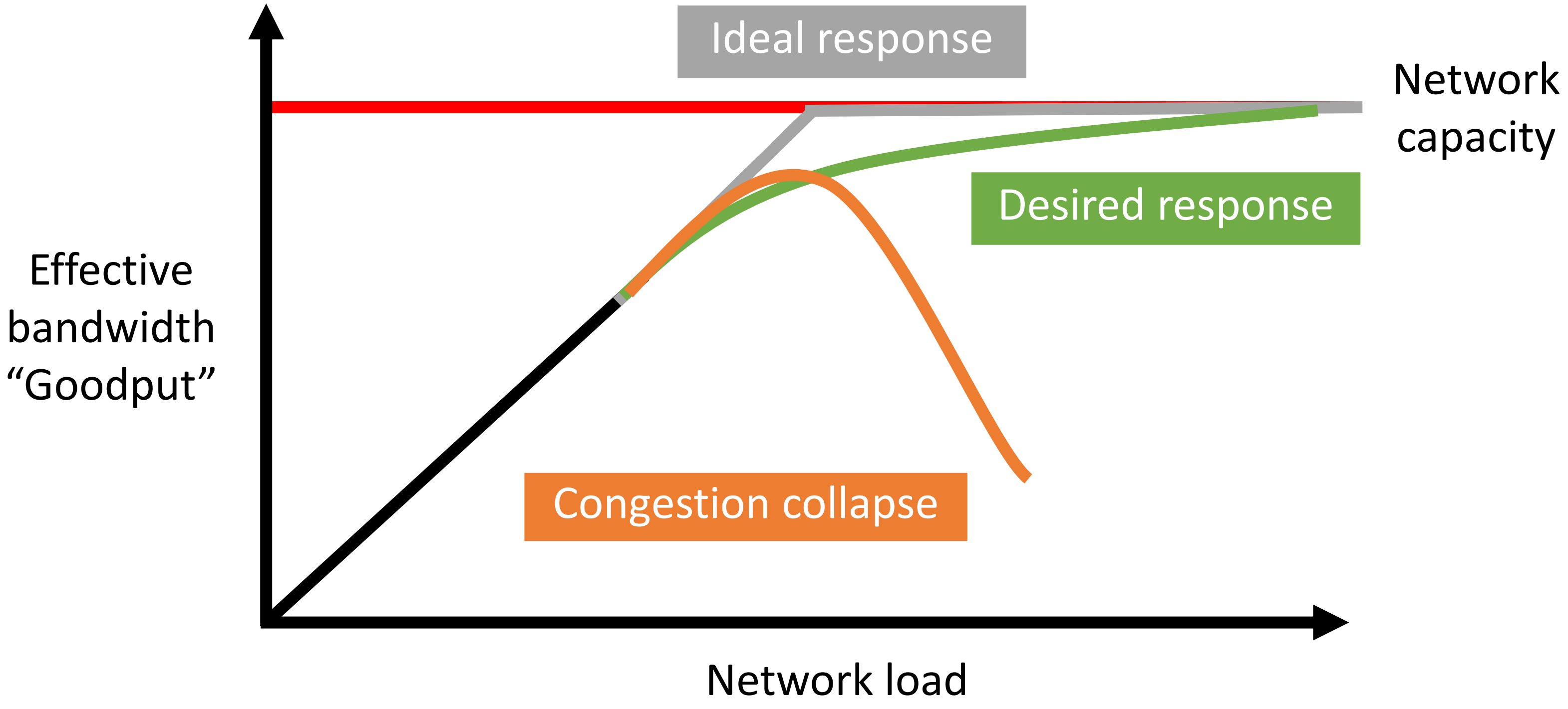
Q: What can go wrong?

Q: Did we fix the issue?

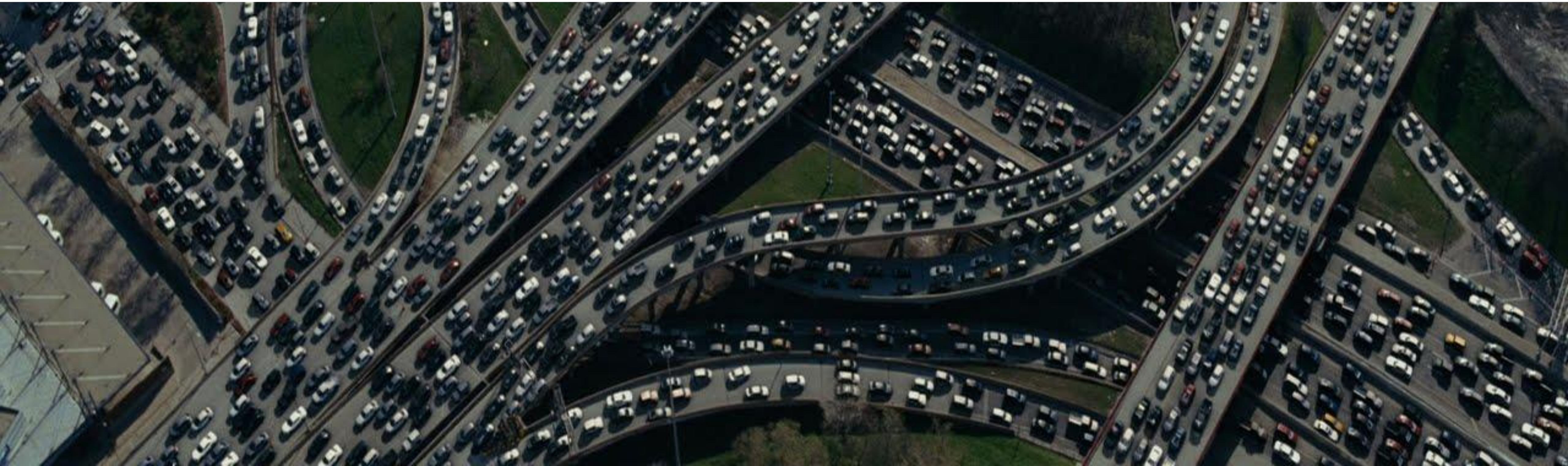
Congestion control

Goodput: rate of useful packets arriving at the receiver

Combined responsibility of the ***network*** and ***transport*** layers.



How can we fix this?



Approaches to congestion control

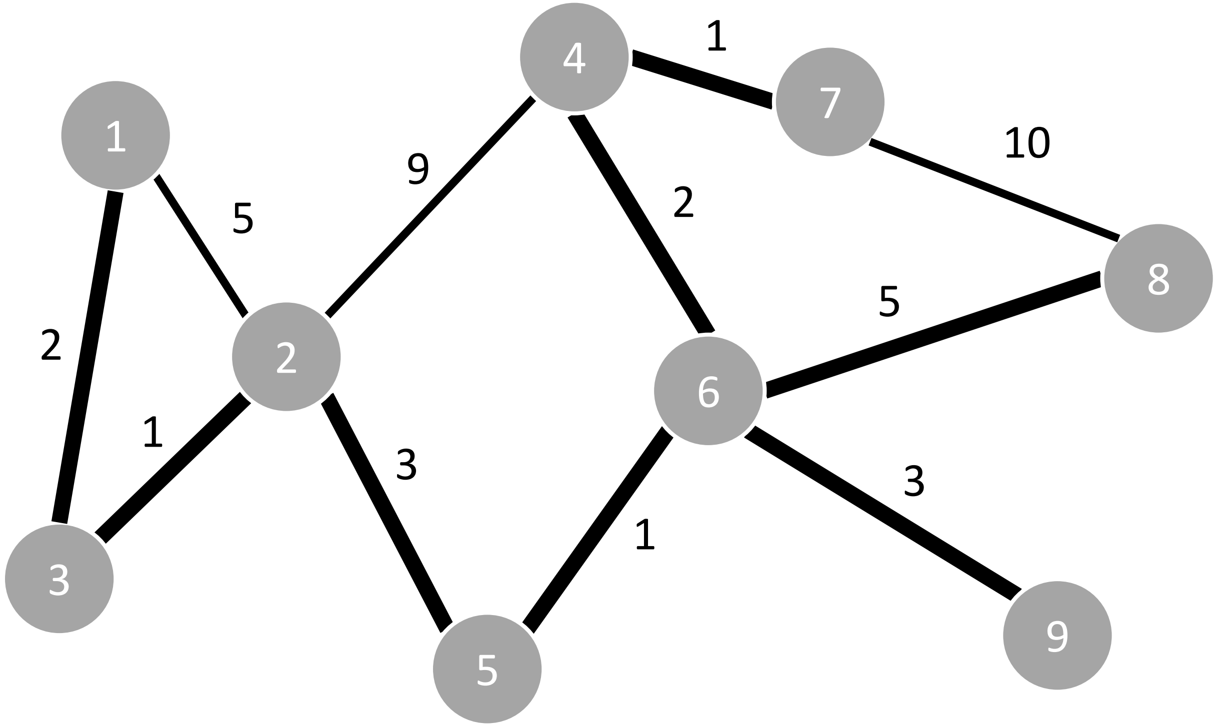
Can we do something smarter?

Simplest approach is *resource over-provisioning*.

Preventing congestion by installing more bandwidth.

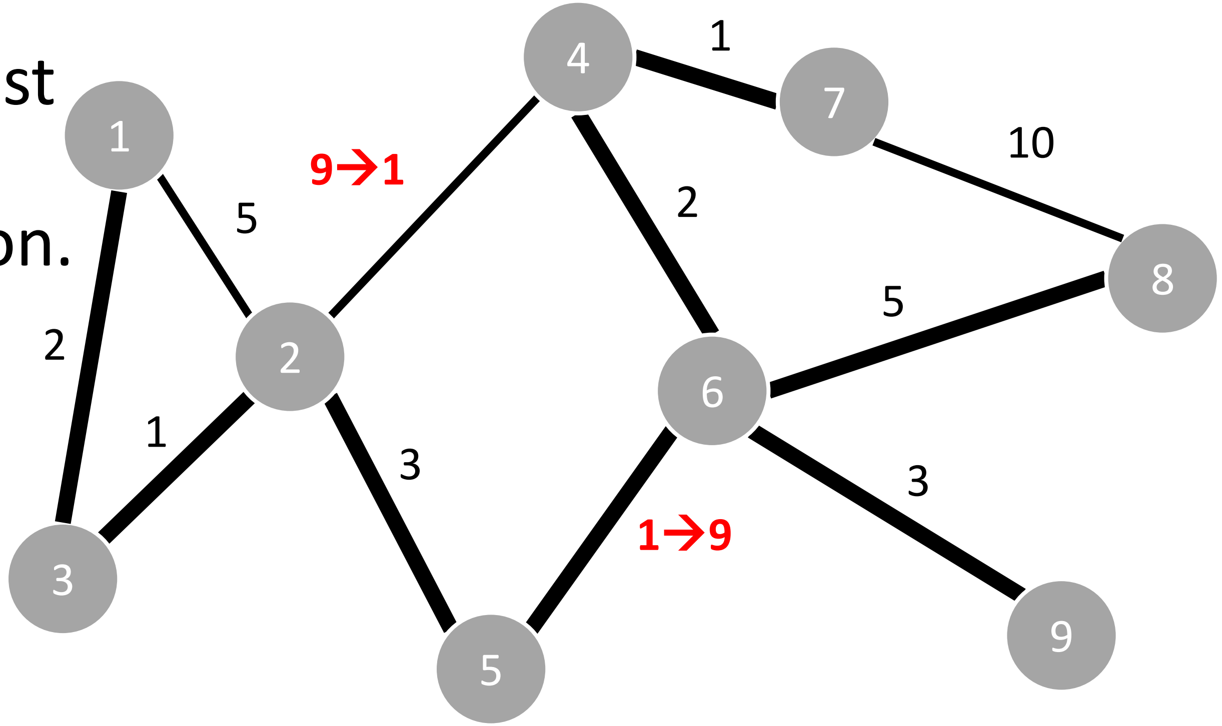
Traffic-aware routing

If link costs are static, all traffic is routed over lowest-cost links.



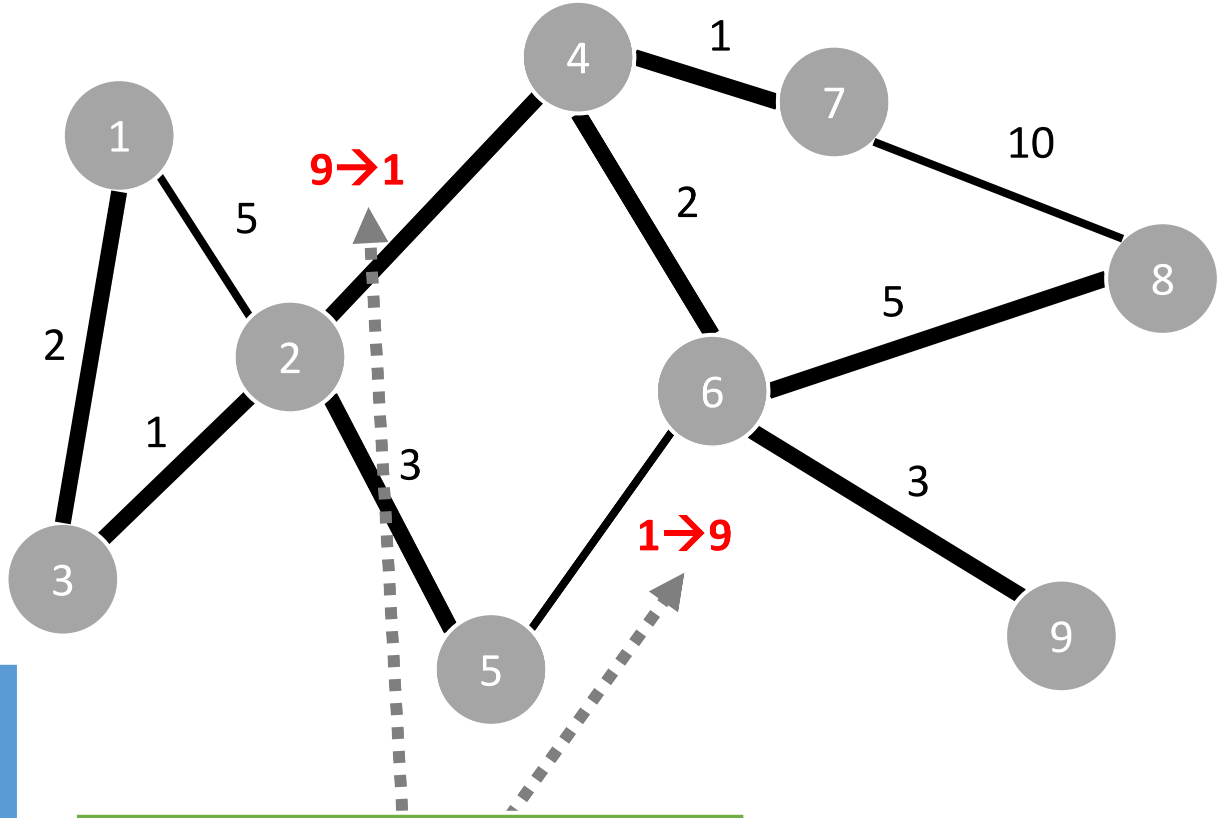
Traffic-aware routing

Using dynamic cost calculation can prevent congestion.



Traffic-aware routing

Using dynamic cost calculation can prevent congestion.

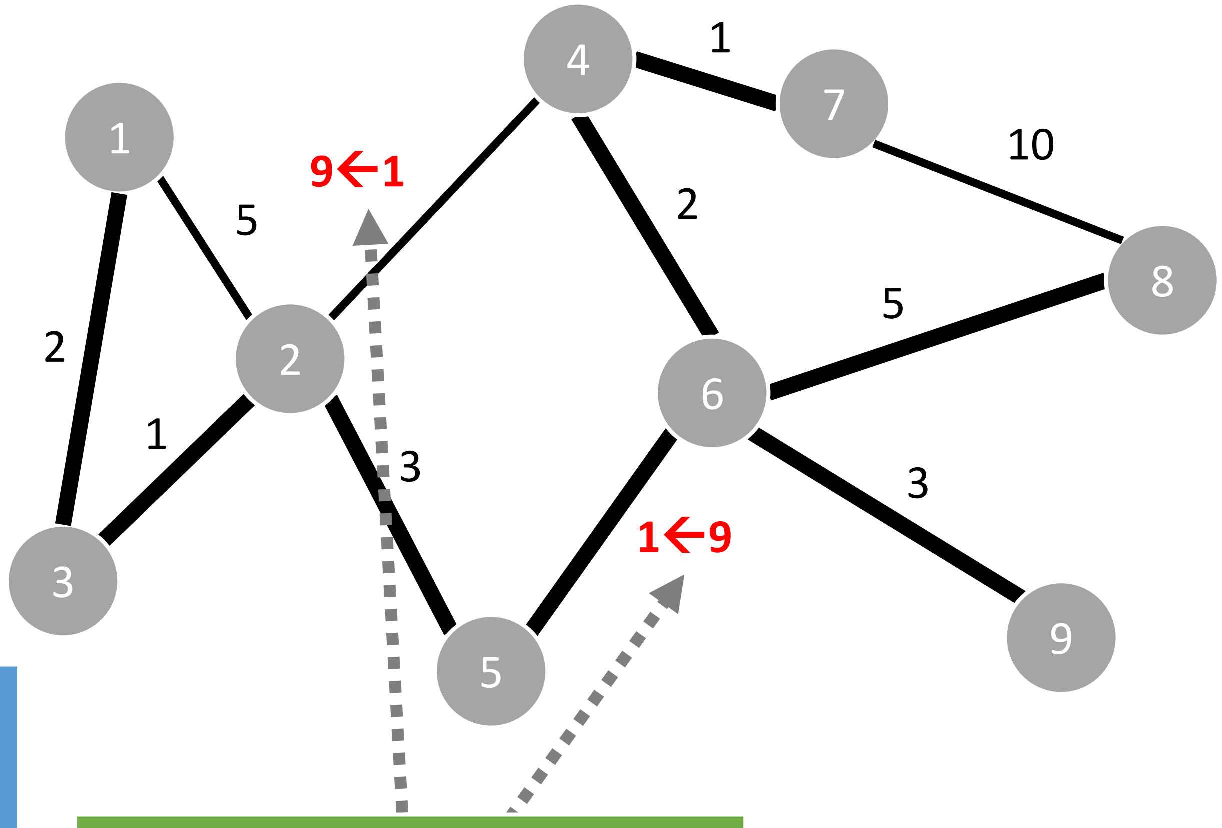


Calculate link cost as a function of current load

Can you think of a problem with this approach?

Traffic-aware routing

Using dynamic cost calculation can prevent congestion.



Calculate link cost as a function of current load

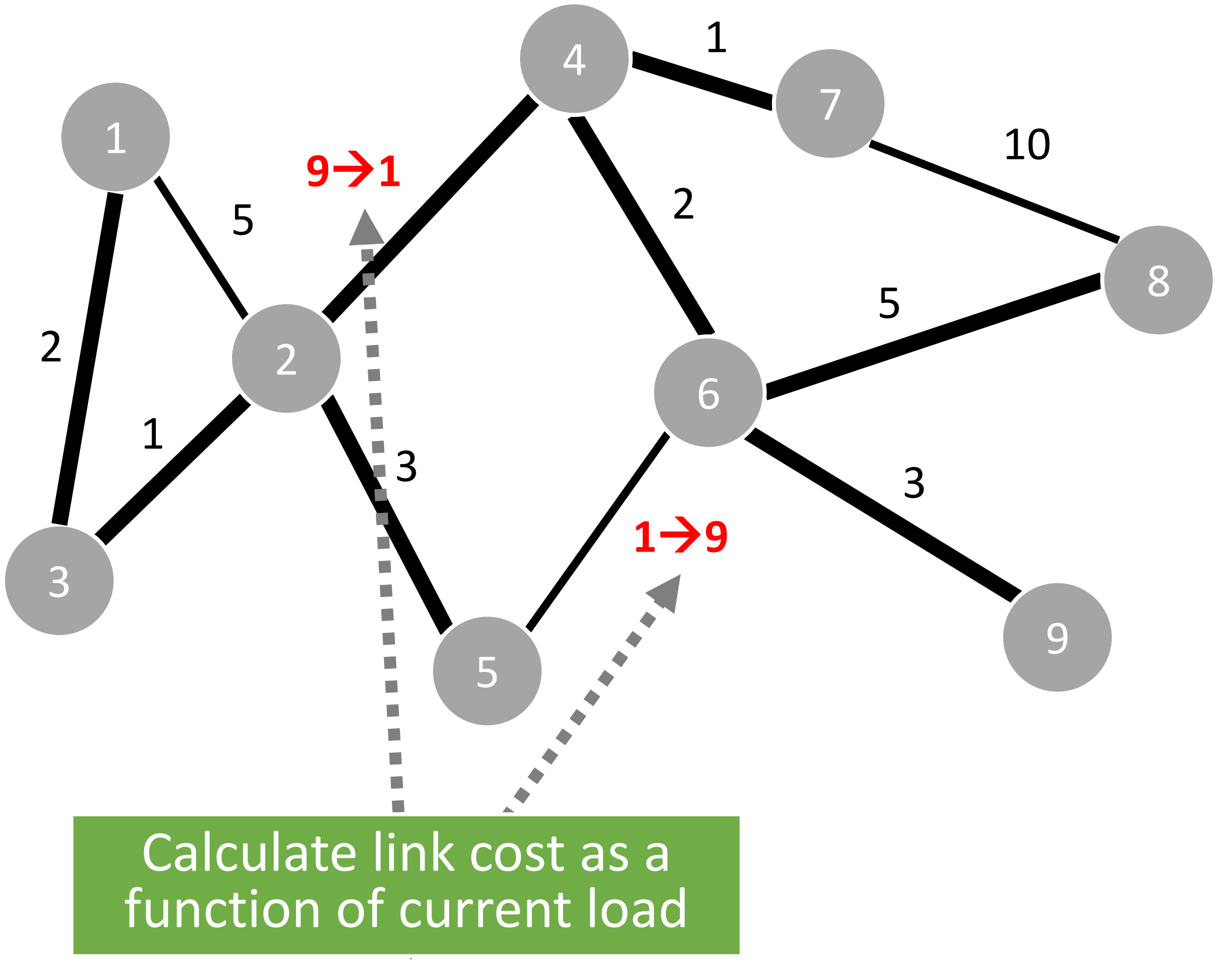
Can you think of a problem with this approach?

Traffic-aware routing

Using dynamic cost calculation can prevent congestion.

Need to prevent oscillations.

1. Small cost updates.
2. Multiple paths.



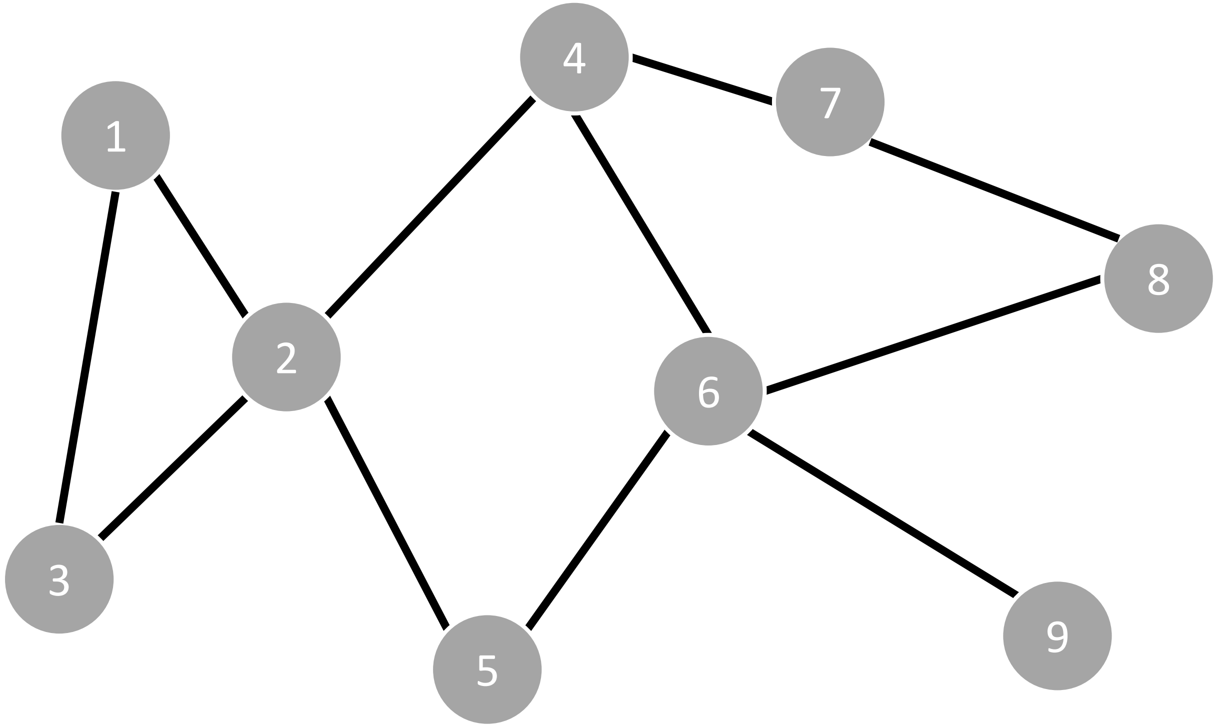
Calculate link cost as a function of current load

Admission Control

If there is congestion, new traffic has to wait!

Admission control

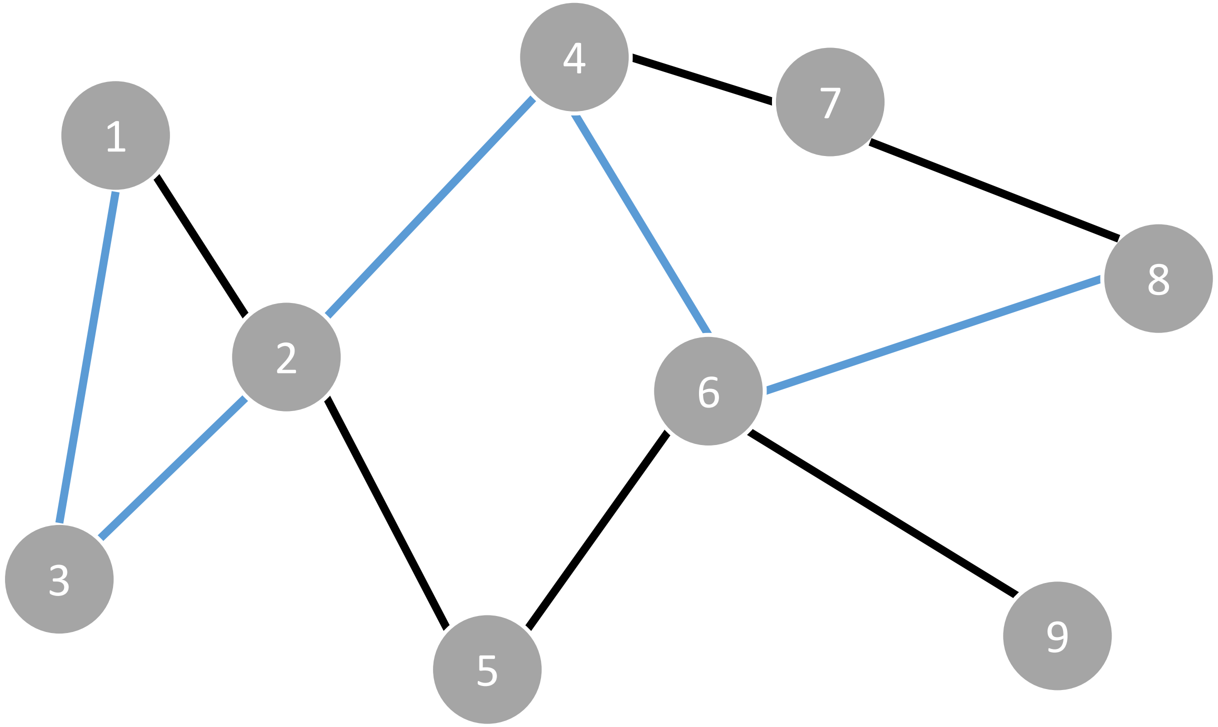
Admission control allows a new traffic load only if the network has sufficient capacity.



Admission control

Admission control allows a new traffic load only if the network has sufficient capacity.

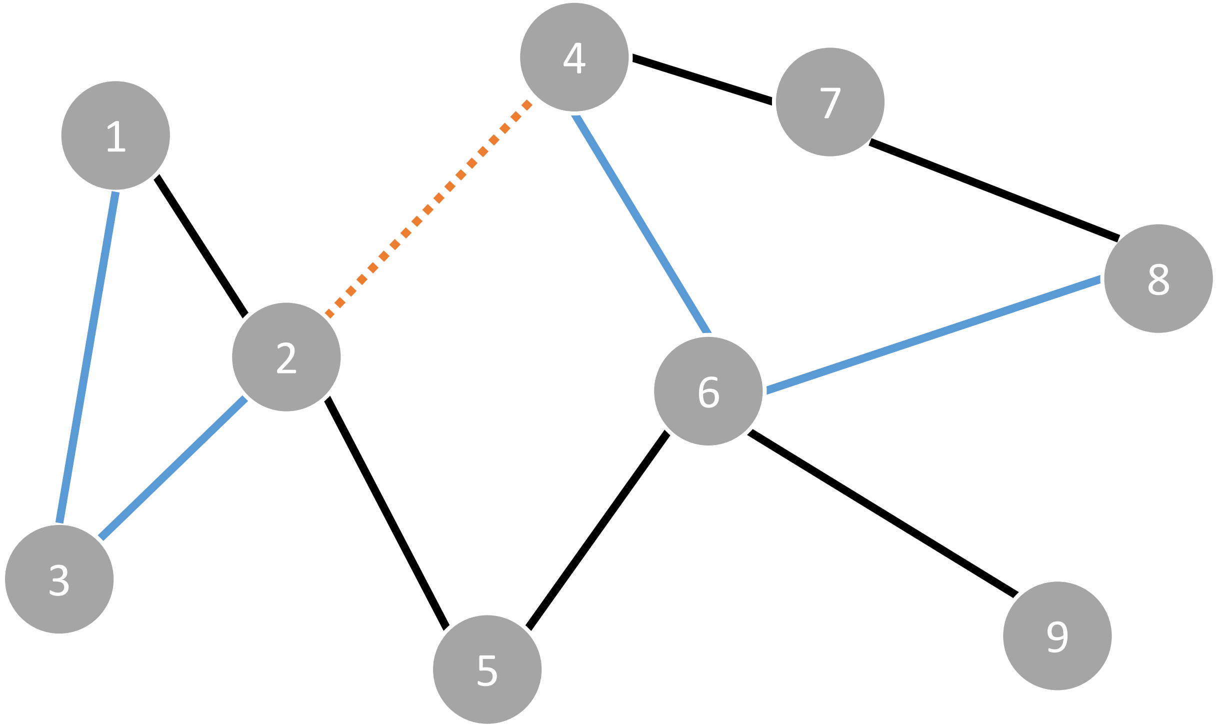
Can you find a path that does not result in congestion?



Admission control

Admission control allows a new traffic load only if the network has sufficient capacity.

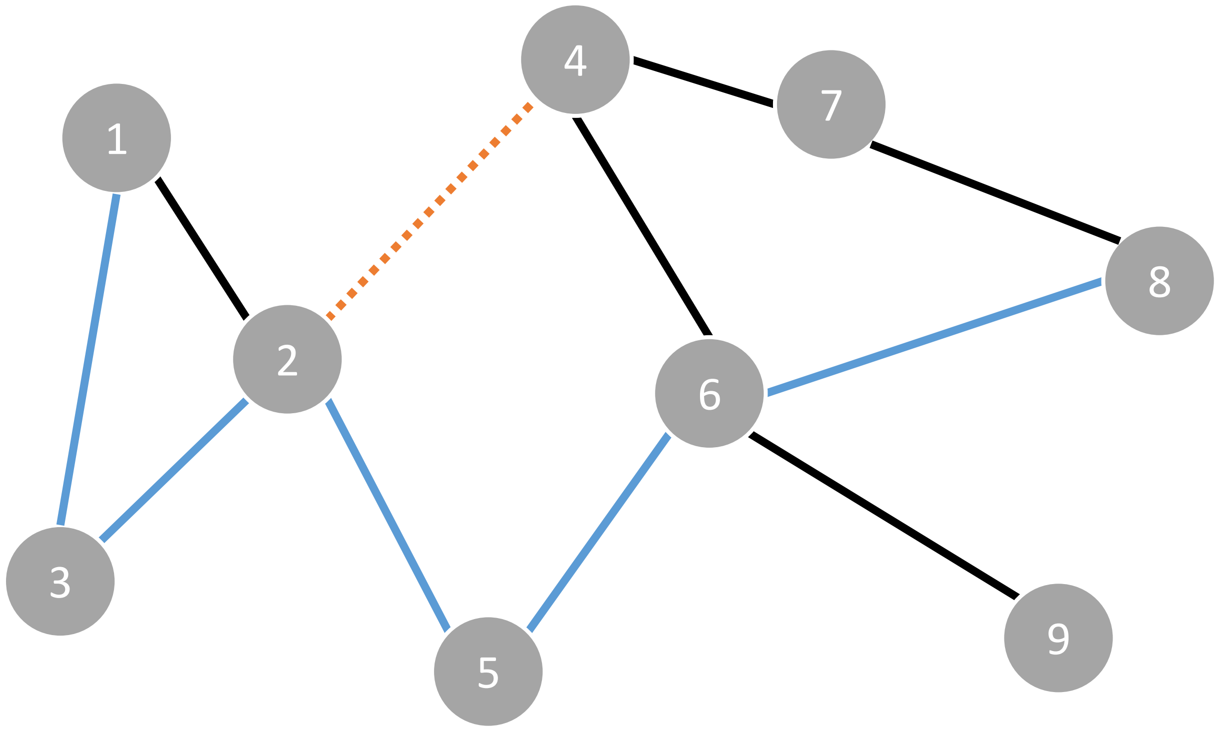
Can you find a path that does not result in congestion?



Admission control

Admission control allows a new traffic load only if the network has sufficient capacity.

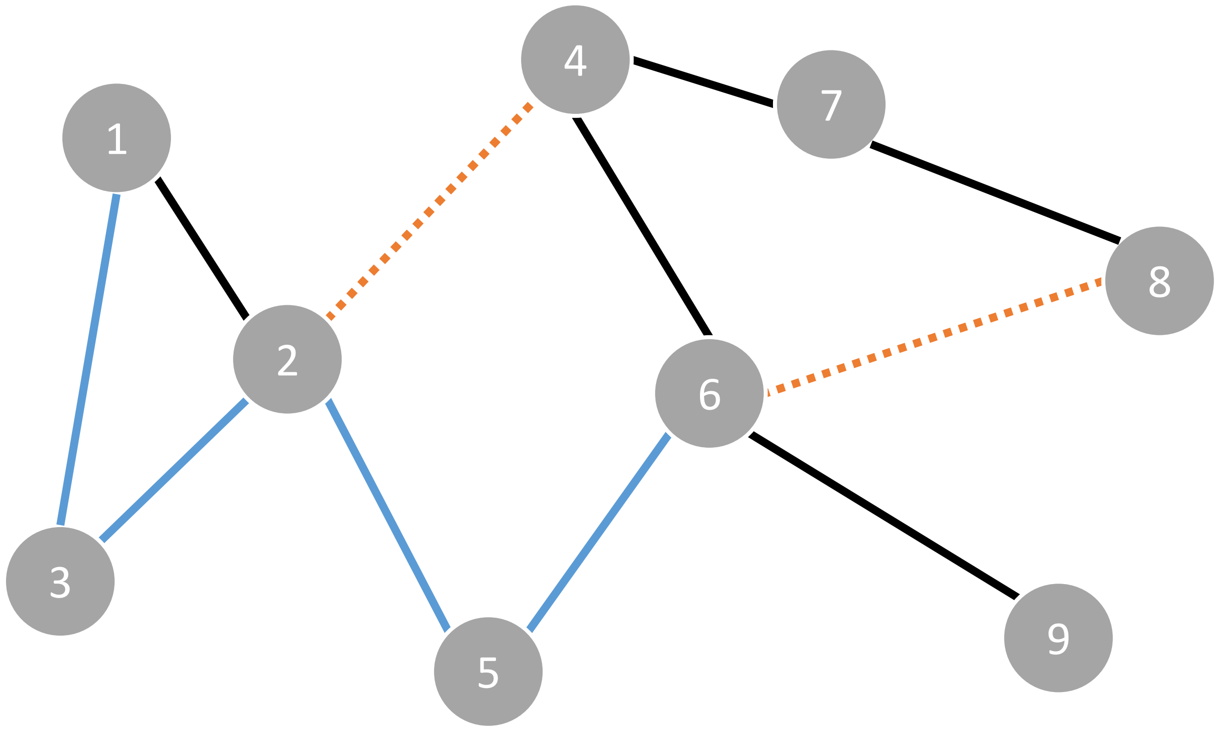
Can you find a path that does not result in congestion?



Admission control

Admission control allows a new traffic load only if the network has sufficient capacity.

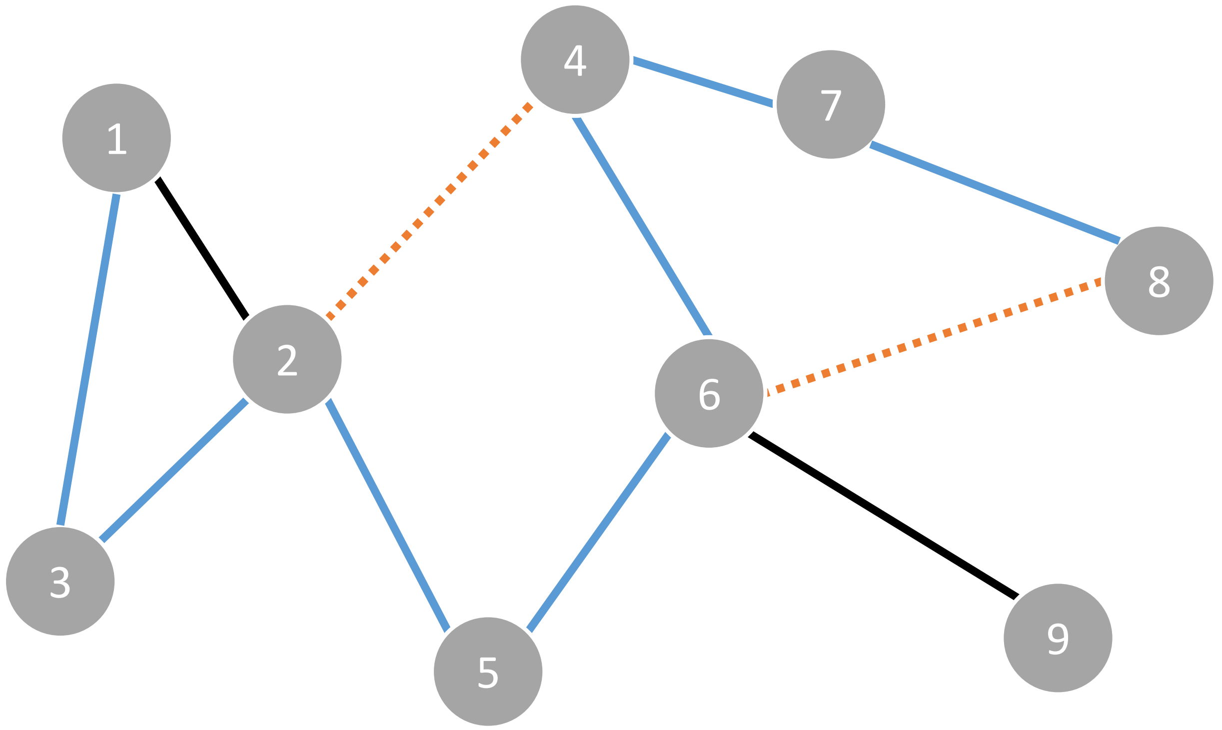
Can you find a path that does not result in congestion?



Admission control

Admission control allows a new traffic load only if the network has sufficient capacity.

Can you find a path that does not result in congestion?



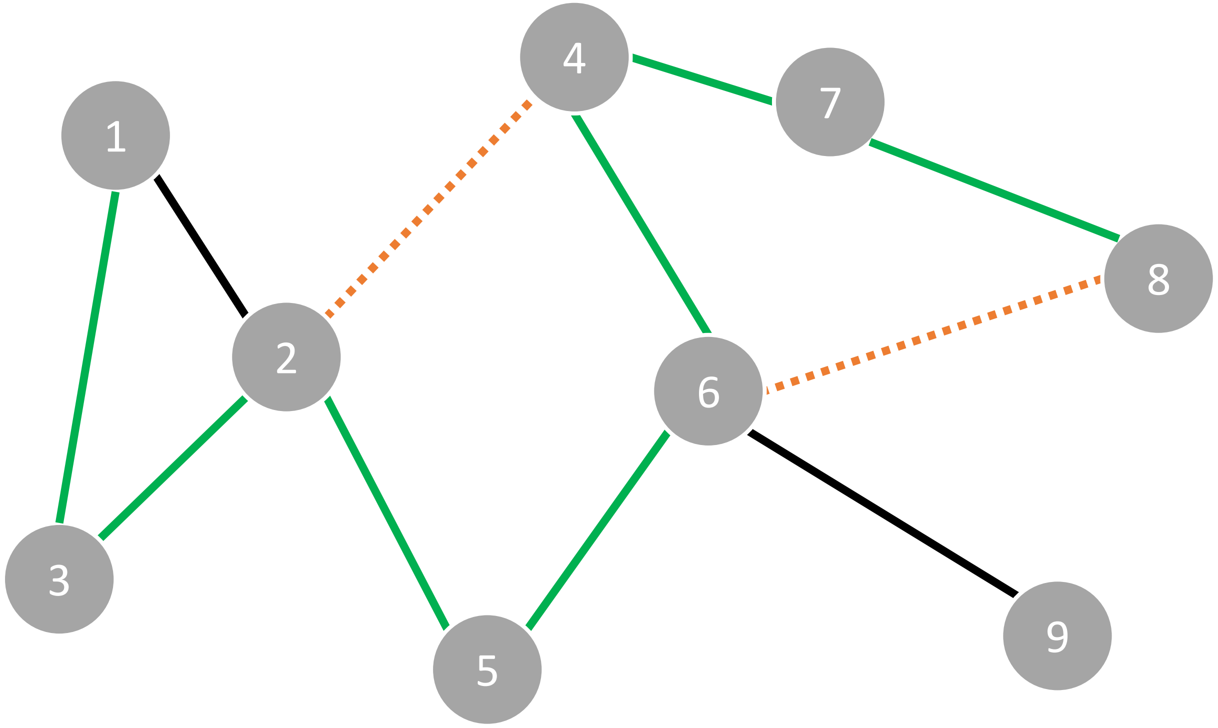
Admission control

Admission control allows a new traffic load only if the network has sufficient capacity.

Can you find a path that does not result in congestion?

Yes: allow traffic.

No:
traffic must wait



Traffic throttling

Send messages in the opposite direction to explicitly indicate network congestion.

Most common implementation:

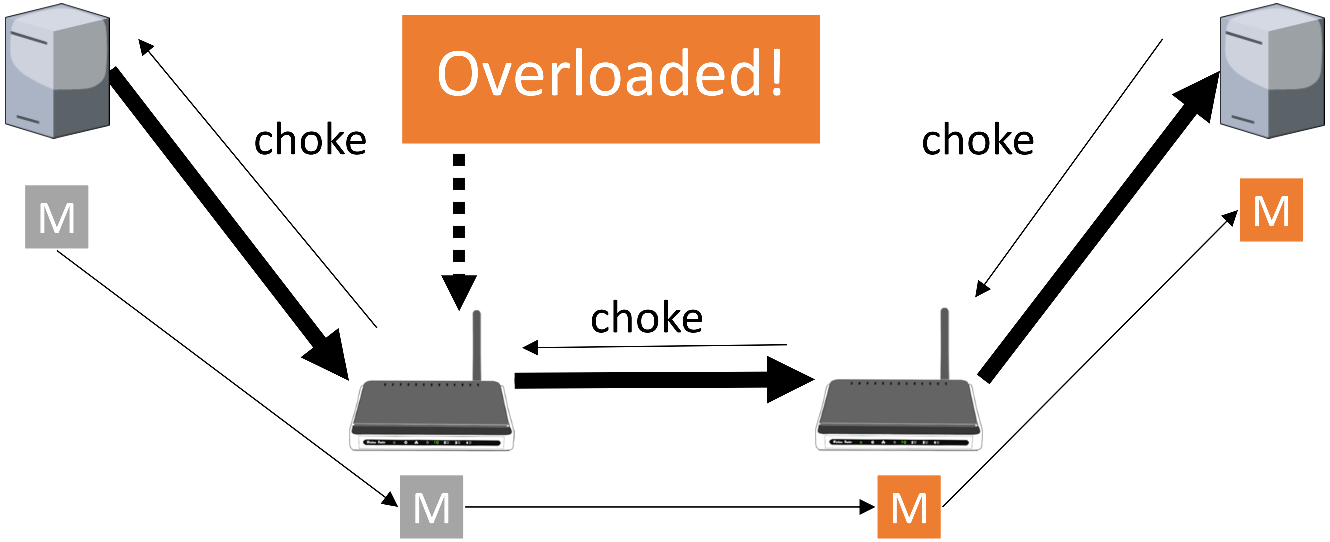
1. Set special bits in IP packet.
2. Inform sender of congestion through TCP.

Traffic throttling
End-to-end

Q: Can think of a (dis)advantage?

Send back a ‘choke’ signal. When ***the source*** receives this packet, it slows down transmission.

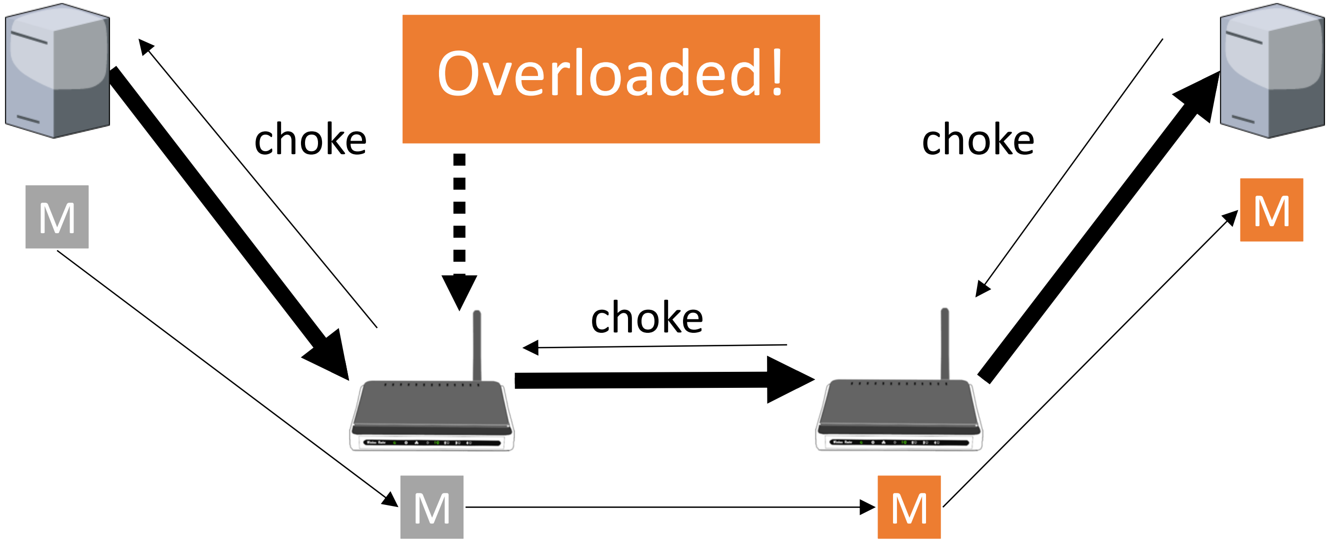
Used in TCP/IP via Explicit Congestion Notification (ECN).



Traffic throttling
Link-by-link

Q: Can think of a (dis)advantage?

Send back a ‘choke’ signal. ***Every router*** that receives this packet slows down transmission.


Overloaded!

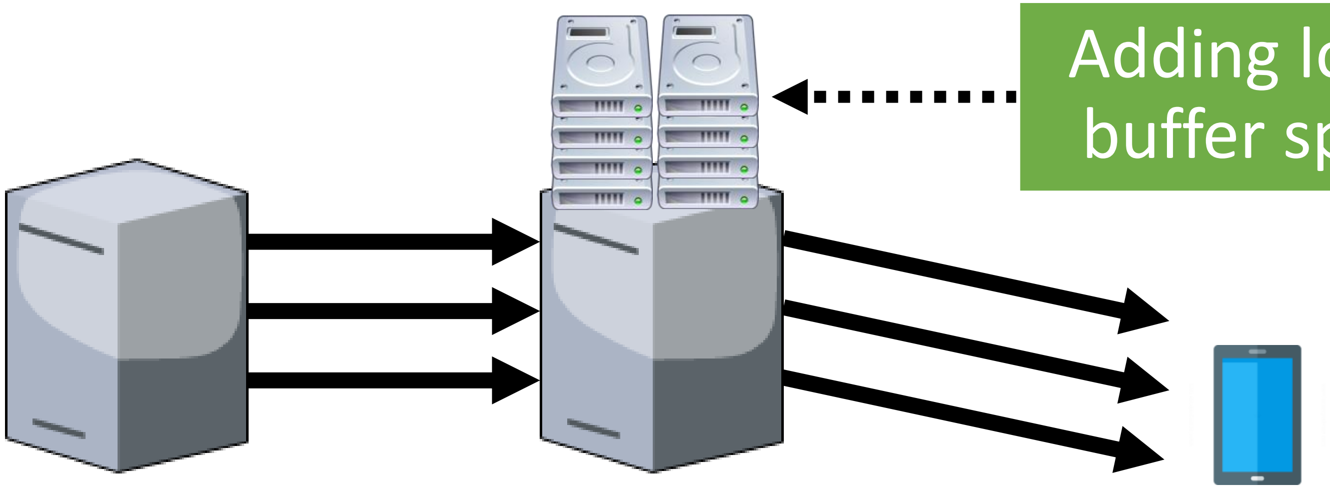
Traffic Shaping

Regulating Network Resource Usage

Looking back on flow control

Mechanism in data link layer.

Makes sure a sender does not send information faster than a receiver can accept.



Adding lots of buffer space.

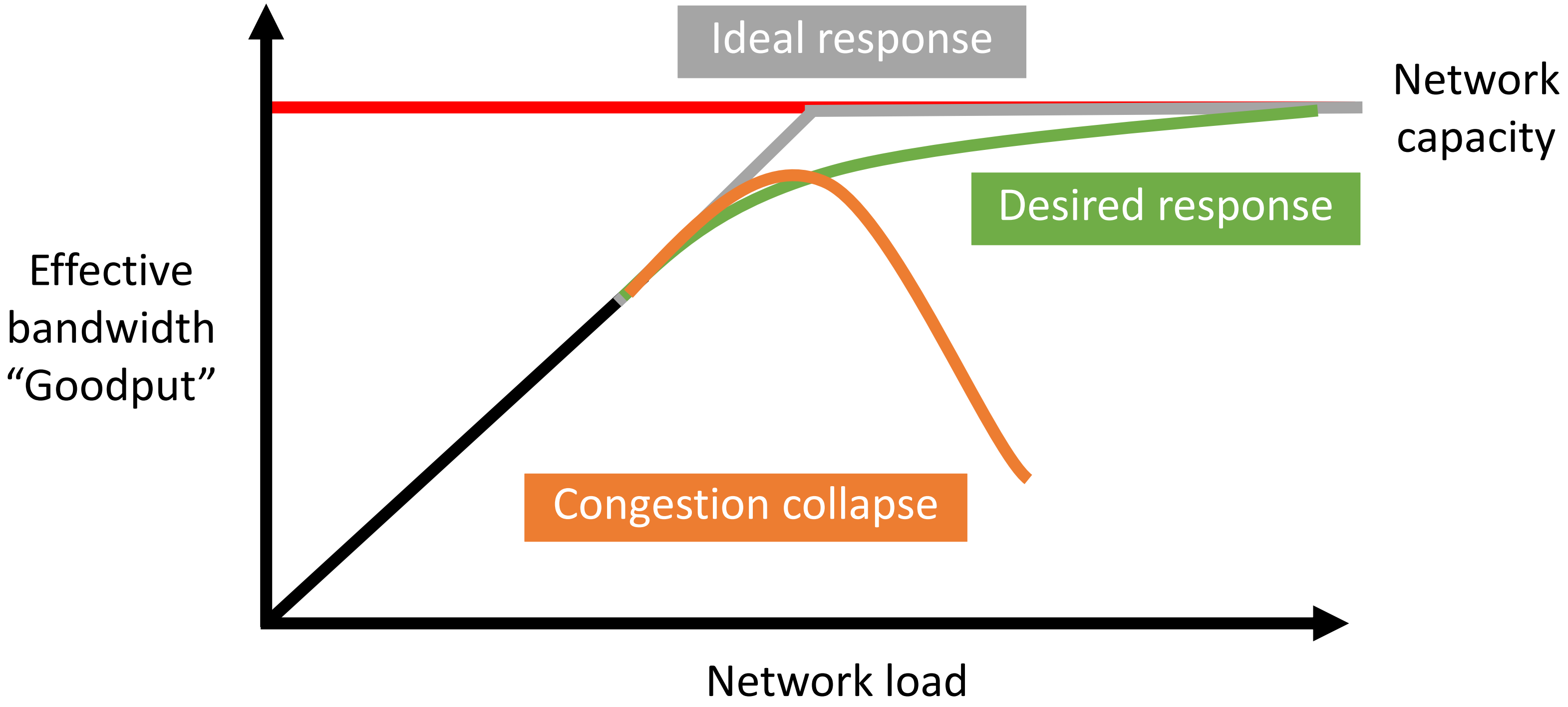
Q: What can go wrong?

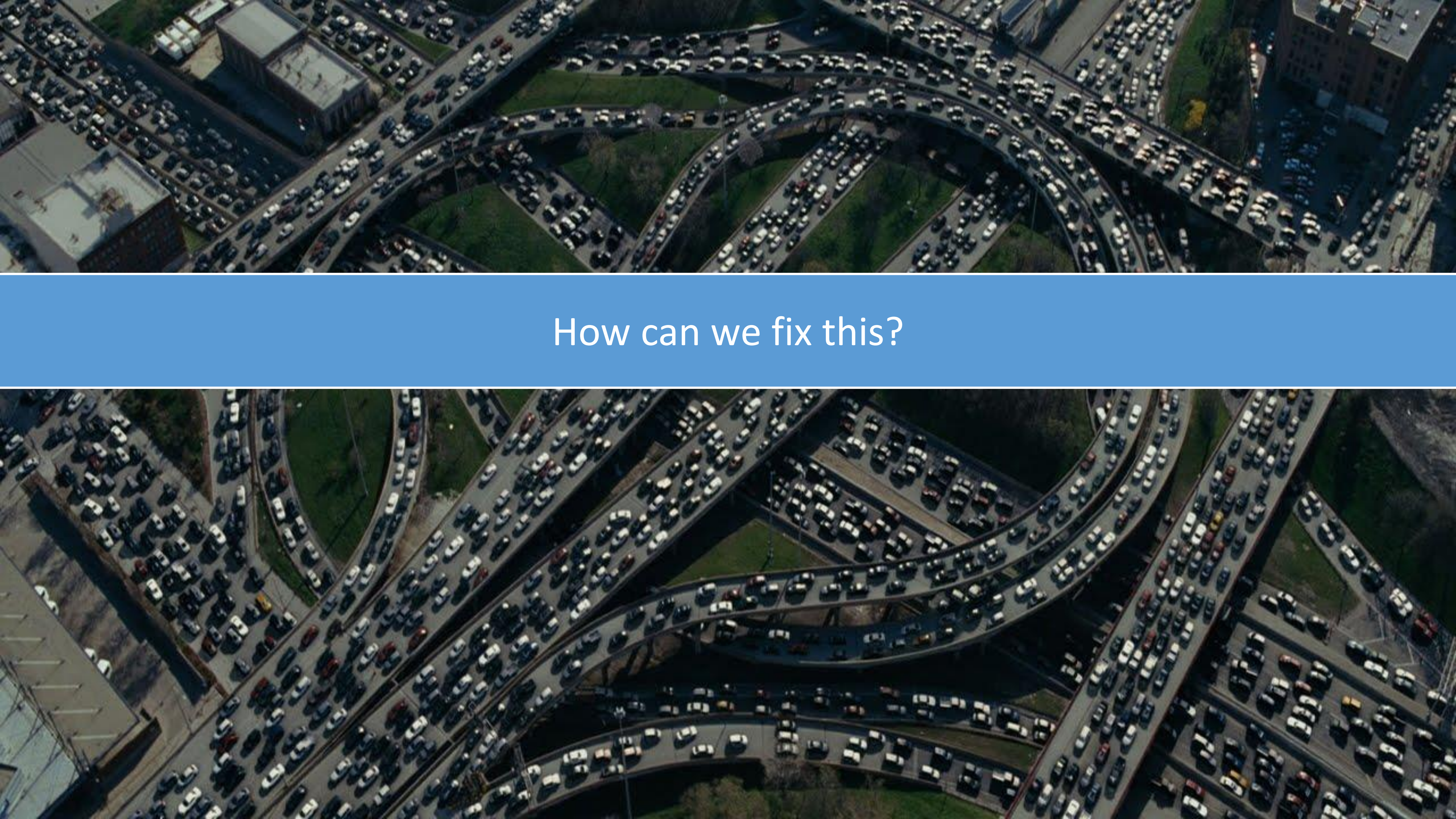
Q: Did we fix the issue?

Congestion control

Goodput: rate of useful packets arriving at the receiver

Combined responsibility of the ***network*** and ***transport*** layers.



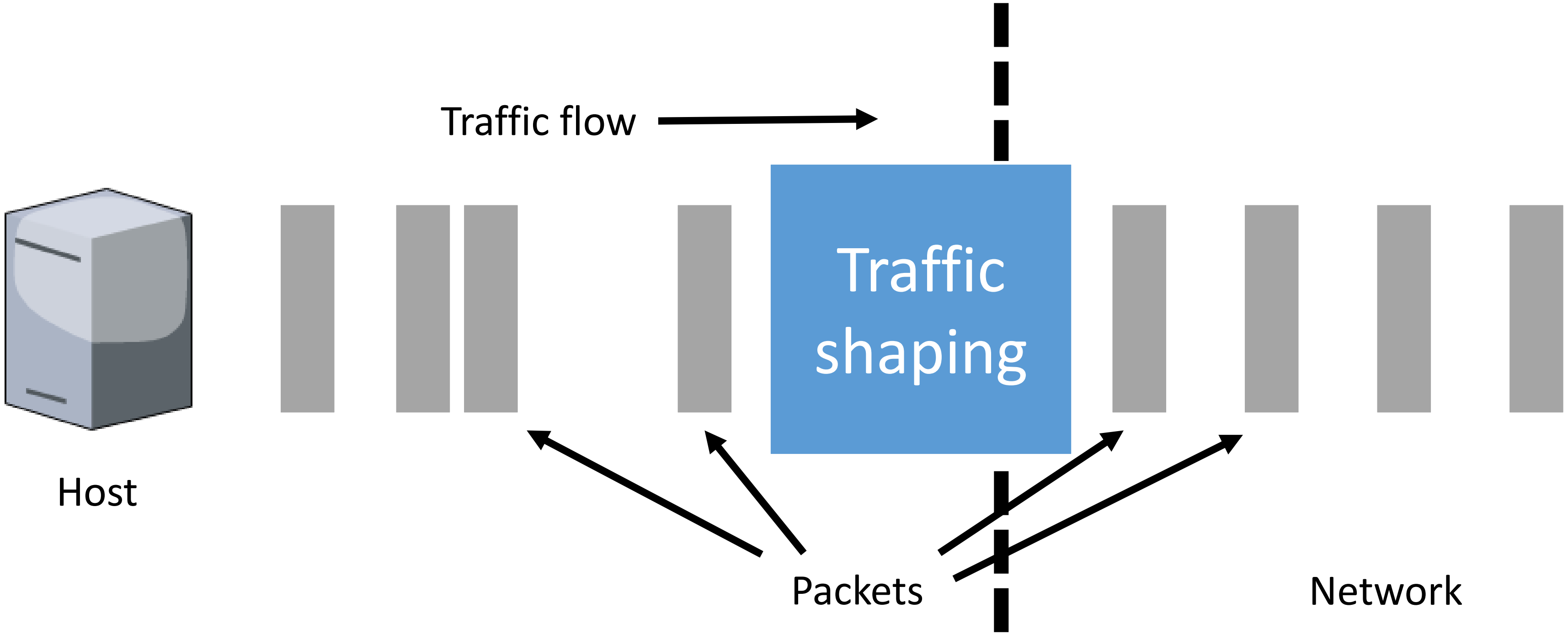


How can we fix this?

Traffic shaping

Challenge: limit available data data rate, but allow bursty traffic

Regulates **rate** and **burstiness** of data entering the network.

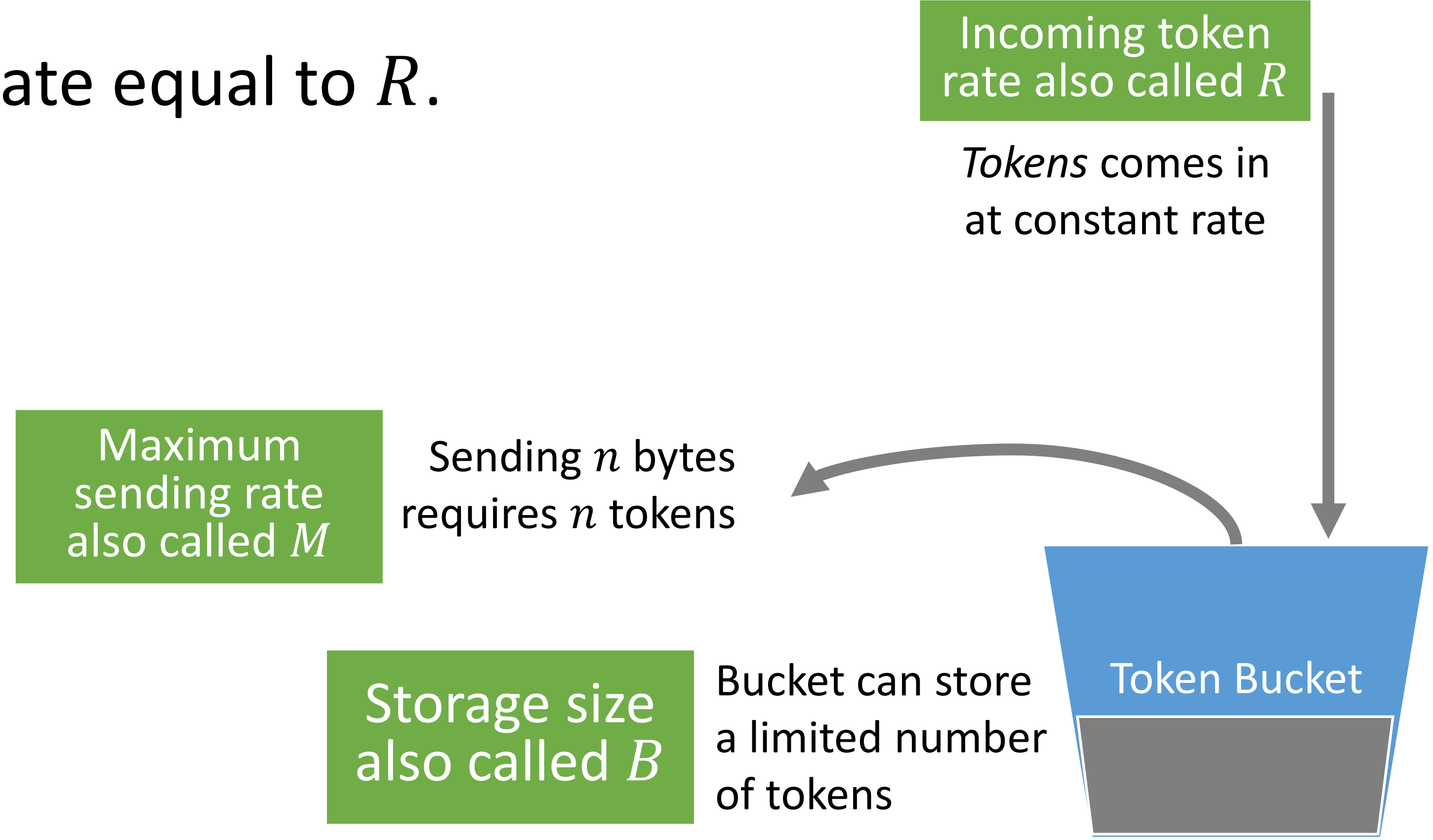


Traffic shaping
Token bucket

Maximum *burst duration* is $\frac{B}{M-R}$ seconds

Outgoing rate between 0 and B.

Average outgoing rate equal to R.



Incoming token rate also called R

Tokens comes in at constant rate

Maximum sending rate also called M

Sending n bytes requires n tokens

Storage size also called B

Bucket can store a limited number of tokens

Traffic shaping
Token bucket example

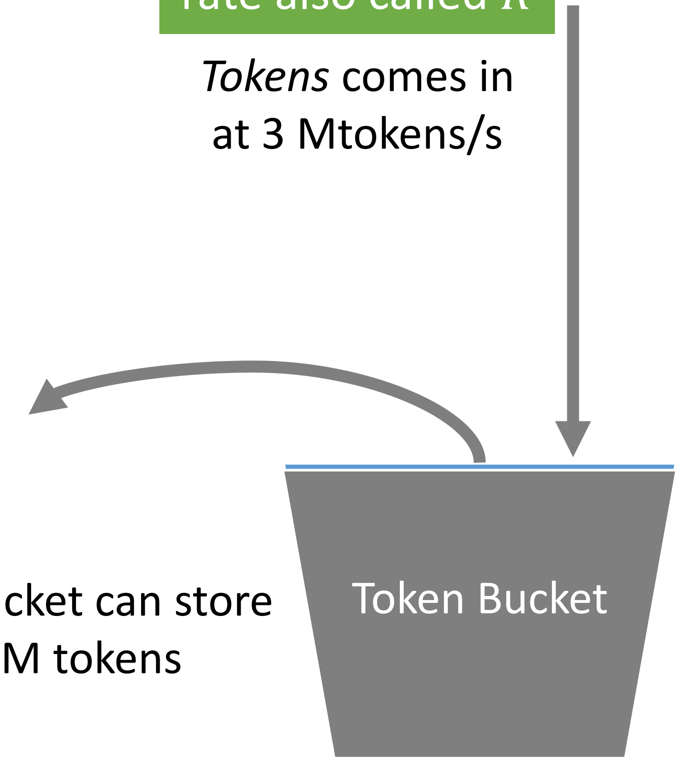
Maximum *burst duration* is $\frac{B}{M-R}$ seconds

Bucket loses 1Mtokens every second.

Full bucket contains 16M tokens.

Maximum burst duration is 16 seconds.

Or: $\frac{16}{4-3} = \frac{16}{1} = 16\text{s}.$



Incoming token rate also called R

Tokens comes in at 3 Mtokens/s

Maximum sending rate also called M

Machine wants to send 4 MB/s

Storage size also called B

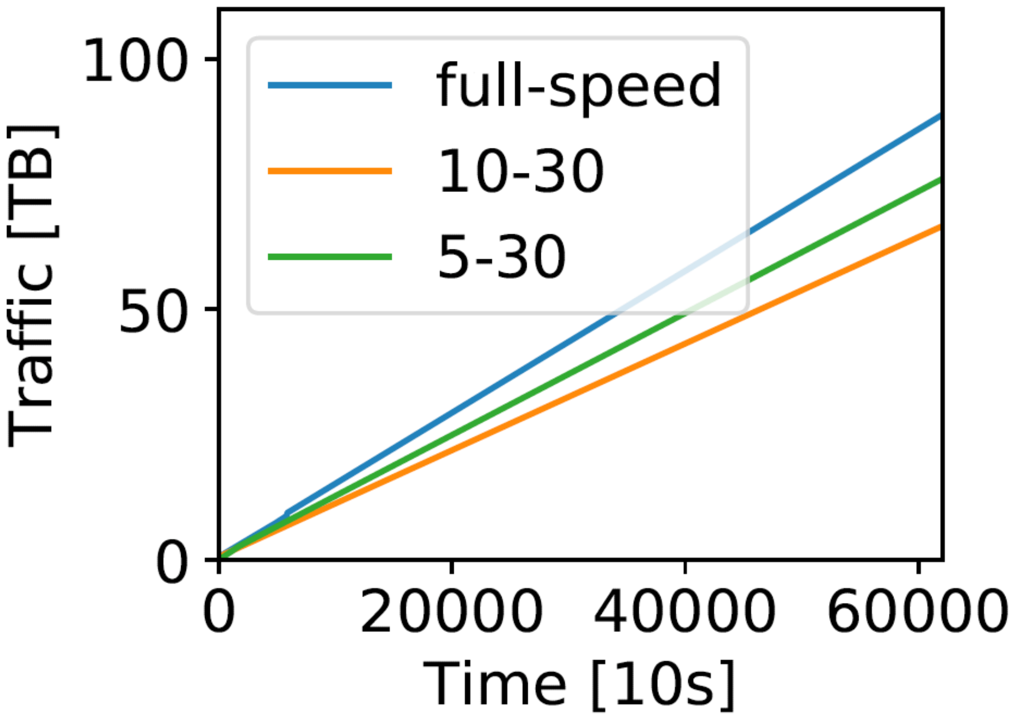
Bucket can store 16M tokens

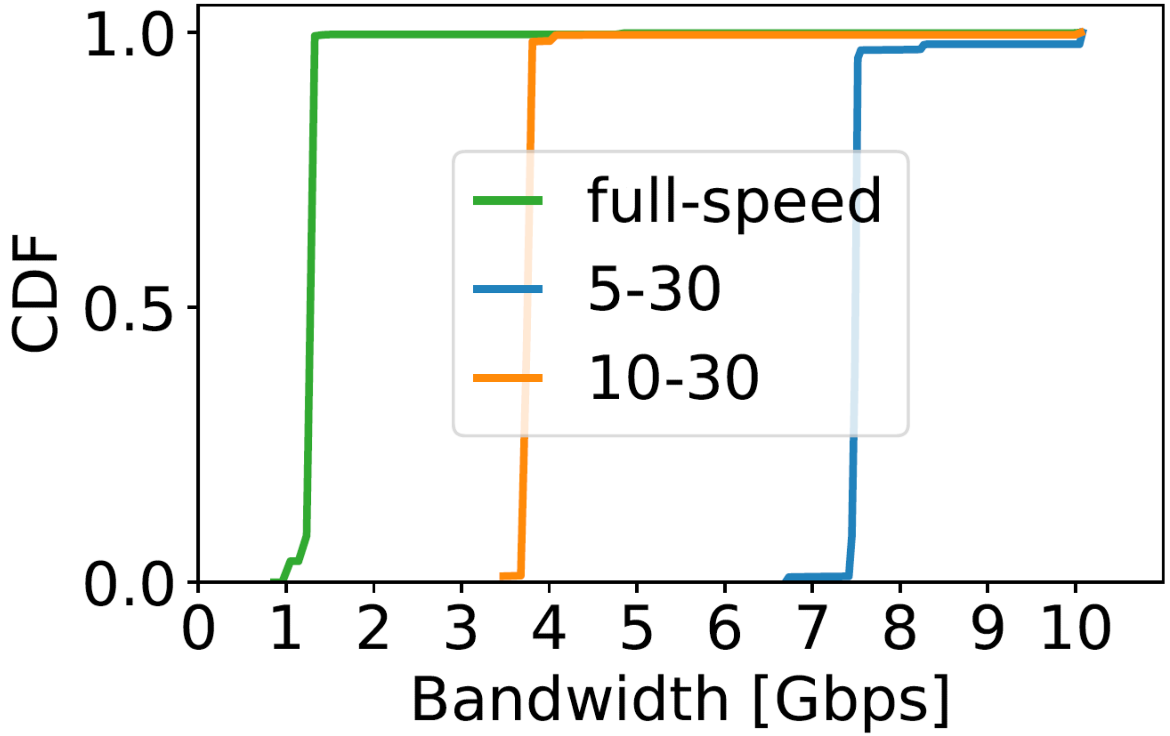
Q: What happens after 16s?

Copyright Jesse Donkervliet 2024

Traffic Shaping in Cloud Networks

Traffic shaping being used in practice

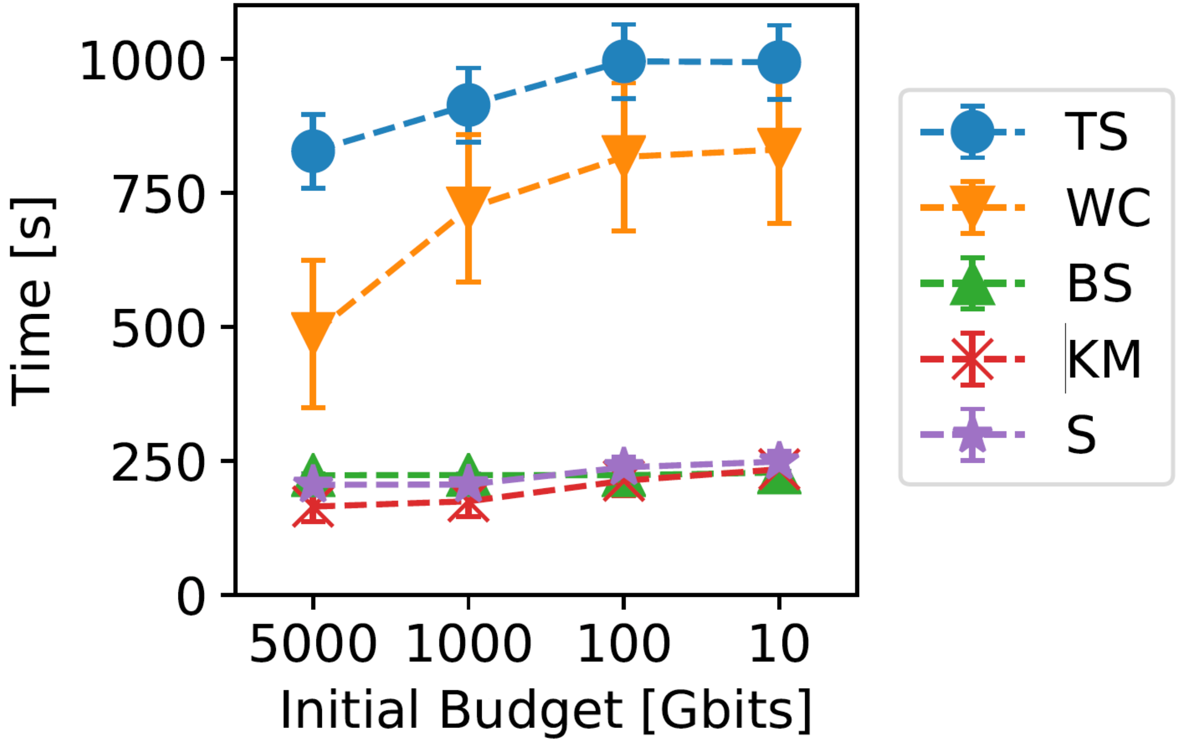




Alexandru Uta, Alexandru Custura, Dmitry Duplyakin, Ivo Jimenez, Jan Rellermeyer, Carlos Maltzhan, Robert Ricci, Alexandru Iosup (2020) Is Big Data Performance Reproducible In Modern Cloud Networks? USENIX Networked Systems Design and Implementation (NSDI)

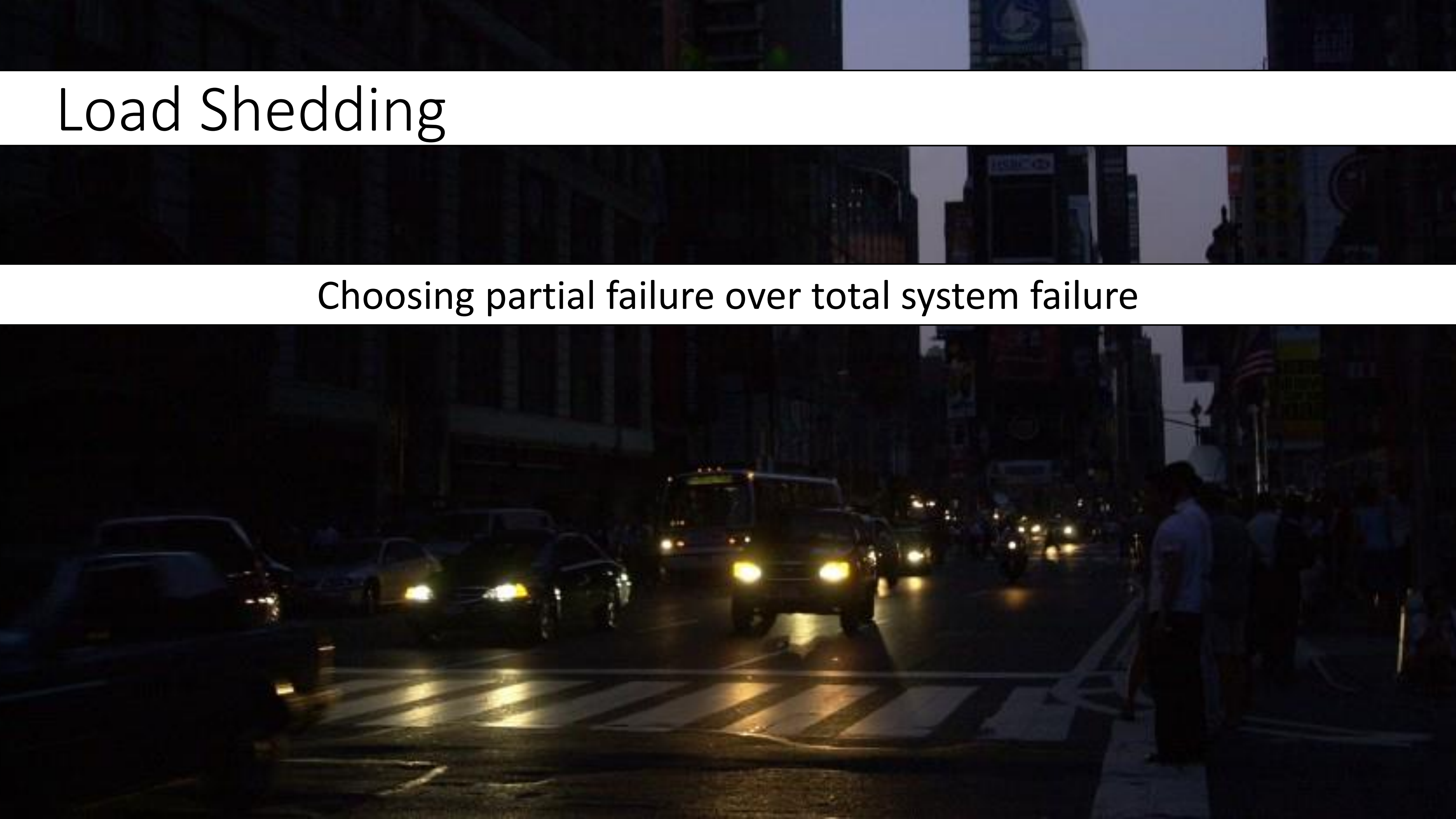
Traffic Shaping in Cloud Networks

Traffic shaping affecting performance



(a) Average runtime.

Alexandru Uta, Alexandru Custura, Dmitry Duplyakin, Ivo Jimenez, Jan Rellermeyer, Carlos Maltzhan, Robert Ricci, Alexandru Iosup (2020)
Is Big Data Performance Reproducible In Modern Cloud Networks? USENIX Networked Systems Design and Implementation (NSDI)



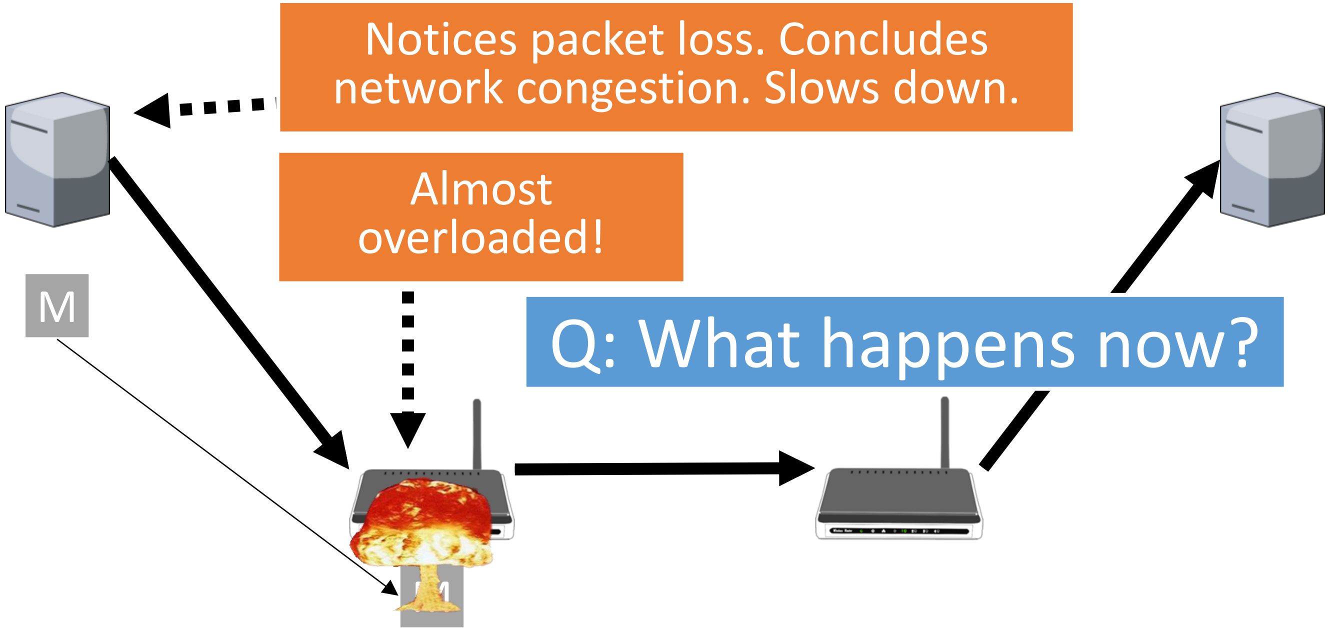
Load Shedding

Choosing partial failure over total system failure

Load Shedding
Random Early Detection (RED)

Drop packets randomly if buffer space is ***almost*** full.

Sends an *implicit* signal to the sender: slow down!



Notices packet loss. Concludes network congestion. Slows down.

Almost overloaded!

Q: What happens now?

Load shedding

Works if transmission errors are unlikely cause of packet loss.

Wired links are reliable (errors are unlikely)

Wireless channels (and other unreliable channels) need to solve transmission errors on the data link layer to hide them from network layer

Quality of Service

We will revisit this problem in later lectures. It is interesting that we encounter this problem at the link layer, but do not have the abililty to solve it without help from higher layers.

Computer networks traditionally offer *best-effort* service

Tries to get data from A to B, but no promises

Q: How is this solved in practice?

Hosts provide *reliable delivery* using retransmissions

Q: What is the problem with this approach?

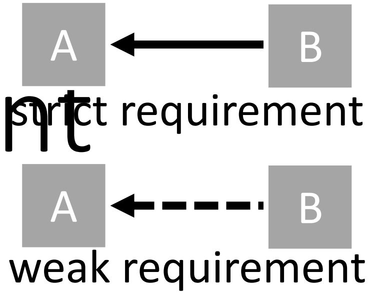
Does not work (well) for many applications:

LEAGUE of LEGENDS
MINECRAFT
STADIA
FORTNITE
skype™
twitch
NVIDIA GEFORCE NOW™

Quality of Service and its parameters

Bandwidth	Maximum data rate. Measured in ***bits per second***
Delay	Time it takes to get from source to destination
Jitter	Variation in packet delay. 0 jitter means delay is constant
Packet loss	Probability of packets being dropped

Different applications have different requirements



Bandwidth

Delay

Jitter

Packet loss

File sharing

Spotify

Audio on demand

NETFLIX

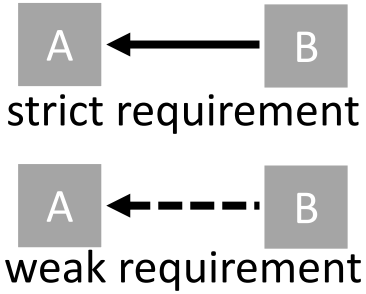
Video on demand

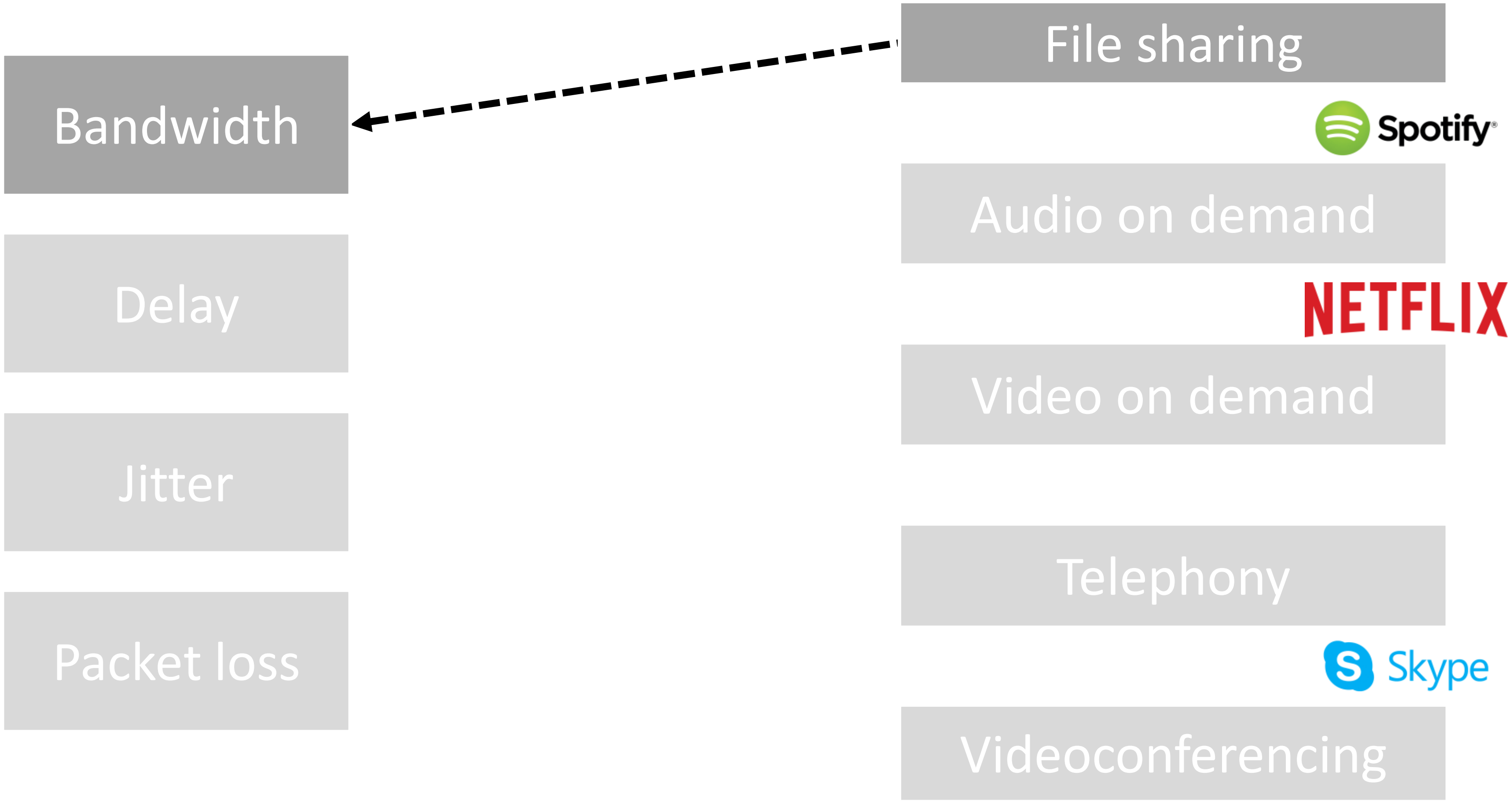
Telephony

Skype

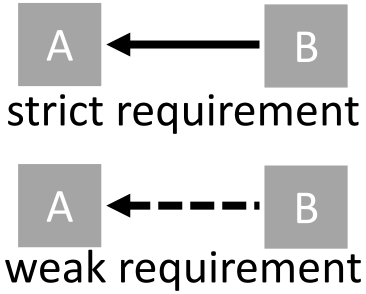
Videoconferencing

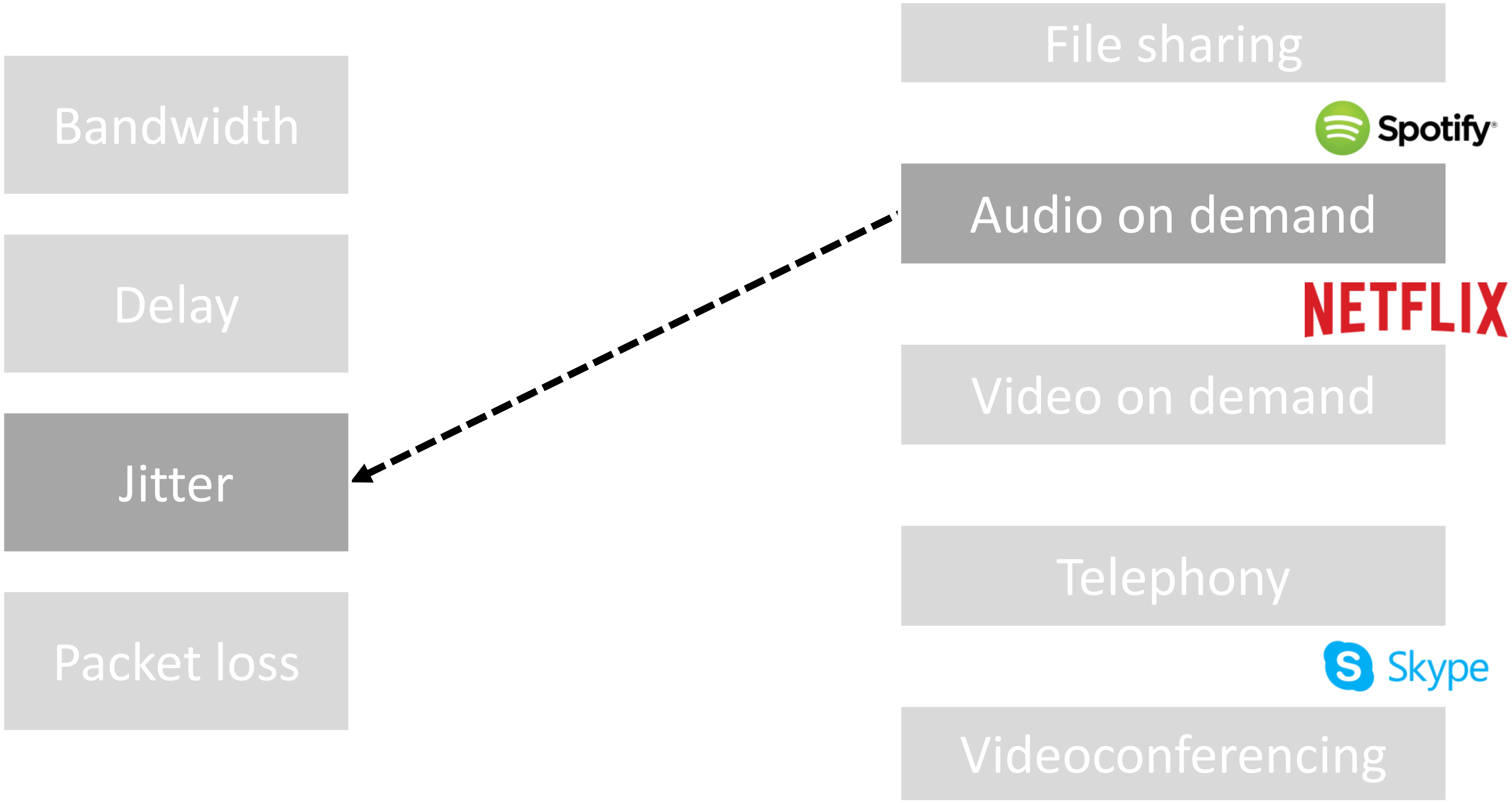
Different applications have different requirements



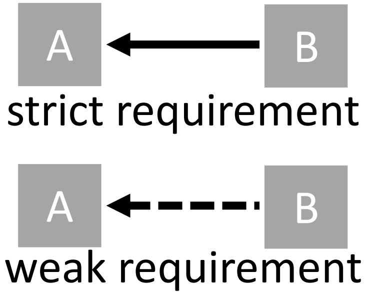


Different applications have different requirements





Different applications have different requirements


A

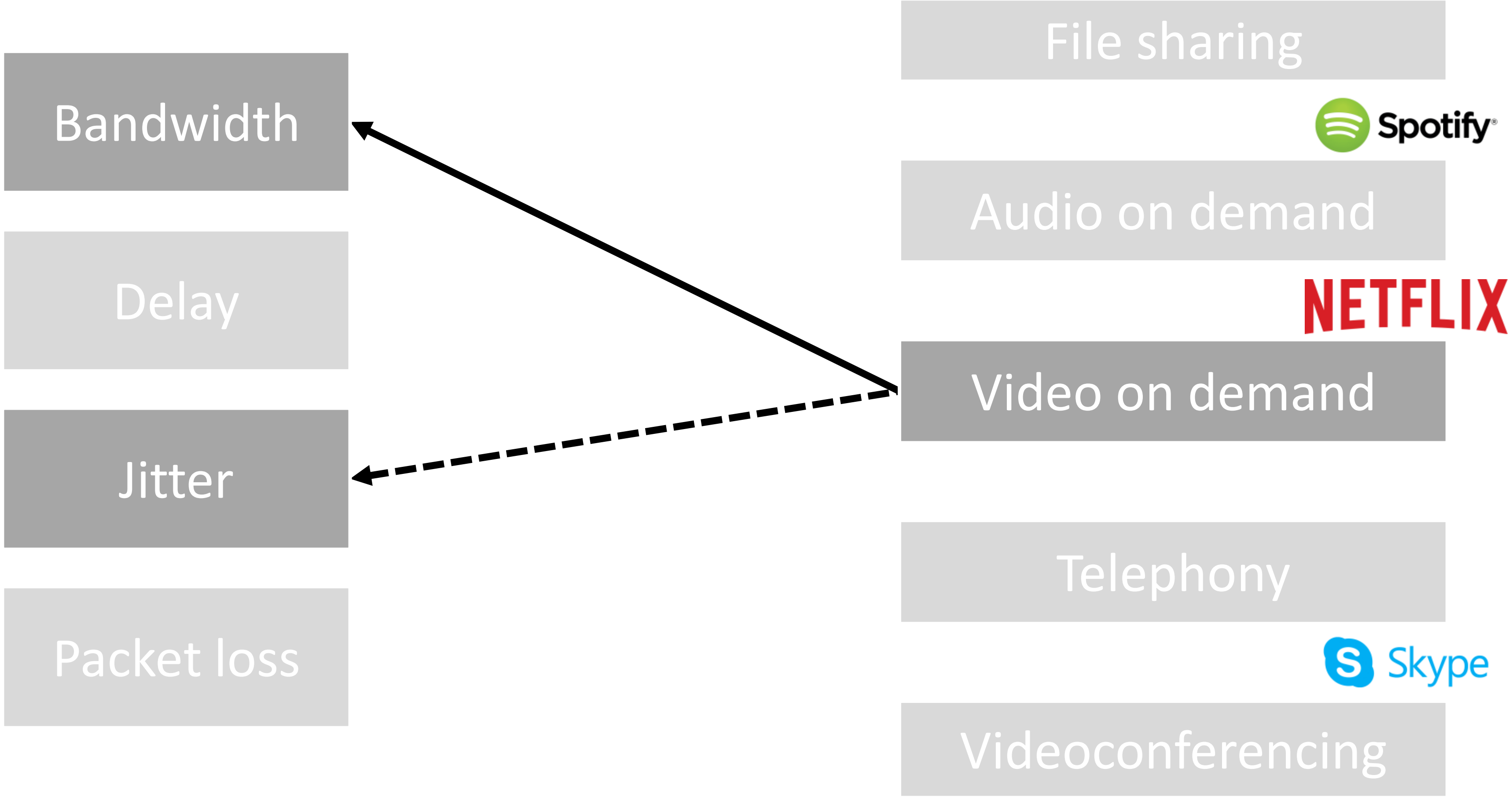
B

strict requirement

A

B

weak requirement


Bandwidth

Delay

Jitter

Packet loss

File sharing

Spotify

Audio on demand

NETFLIX

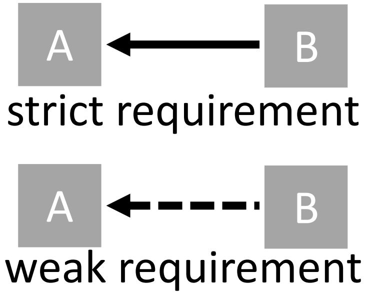
Video on demand

Telephony

Skype

Videoconferencing

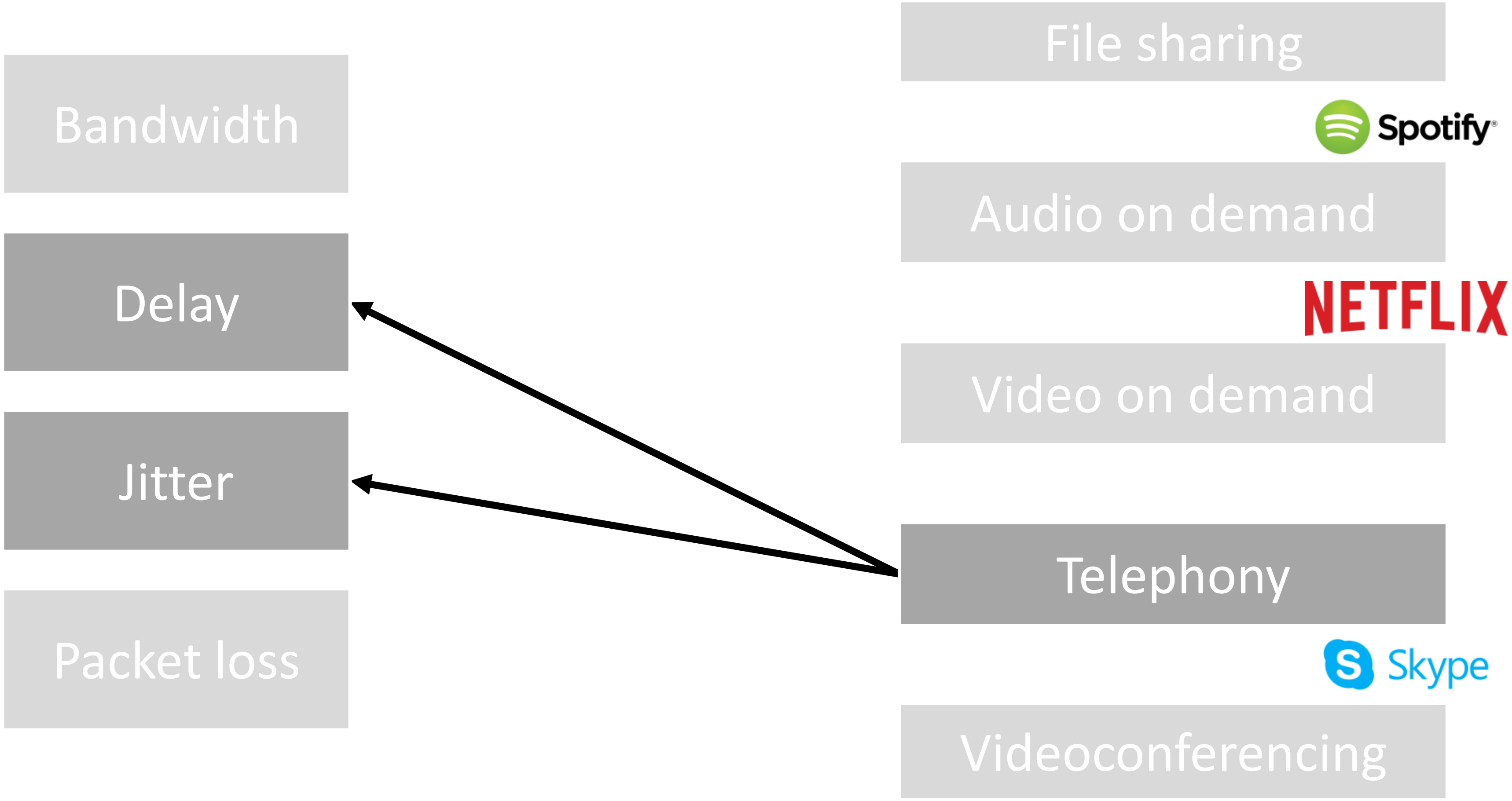
Different applications have different requirements


A ← B

strict requirement

A ← - - - B

weak requirement


Bandwidth

Delay

Jitter

Packet loss

File sharing

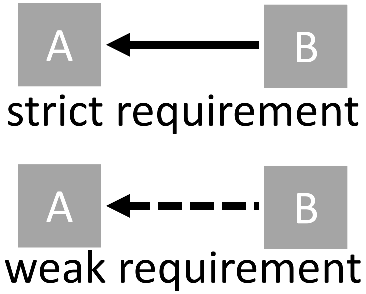
Audio on demand

Video on demand

Telephony

Videoconferencing

Different applications have different requirements

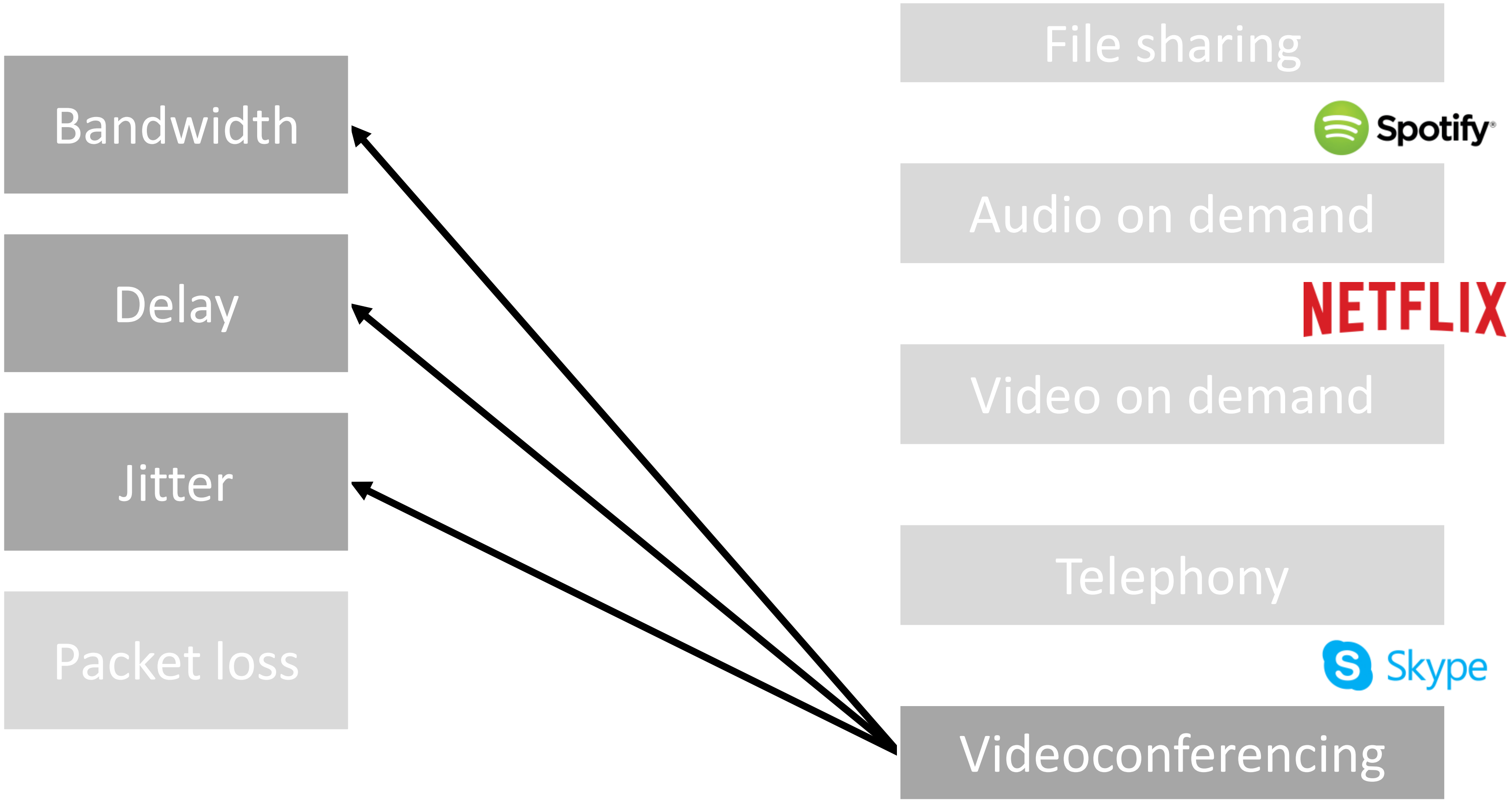


A ← B

strict requirement

A ← - - - B

weak requirement



Bandwidth

Delay

Jitter

Packet loss

File sharing

Spotify

Audio on demand

NETFLIX

Video on demand

Telephony

Skype

Videoconferencing

Quiz Time!

10-15(?) minutes

Correct answers without explanation do *not* get points!

Please ***do not*** use external resources, including:

- ChatGPT (forget AI, use and train your **RI** [Real Intelligence!])
- Anything on or via the Internet (the Web, chat apps, etc)
- Answers from your neighbors
- The book / slides

Network Layer Summary

Networking

- Routing Algorithms:
    - Distance Vector
    - Link State
    - Hierarchical
- Problem of scale: too many addresses
    - Not enough address space (solved by IPv6)
    - Routing tables too large (problem reduced by aggregation)
- Network configuration
    - Obtaining an address (DHCP)
    - Looking up corresponding MAC address (ARP)

Internetworking

- Different networks have different properties
- Using a common protocol (IP).
- Tunneling through networks with other protocols.
- MLPS supports multiple protocols, for faster switching
- Within Autonomous Systems (e.g., OSPF)
- Between Autonomous Systems (e.g., BGP)

Resource Management

- Connectionless and Connection-oriented approaches
- Congestion Control (RED, ECN, etc.)
- Traffic Shaping (Token Bucket, Leaky Bucket)