NETWORKS: THE INTERNET & BEYOND

Richard Harvey

GRESHAM COLLEGE

NETWORKS: THE INTERNET & BEYOND

Richard Harvey

IT Livery Company Professor of Information Technology, Gresham College

UEA

www.prof-richard.org

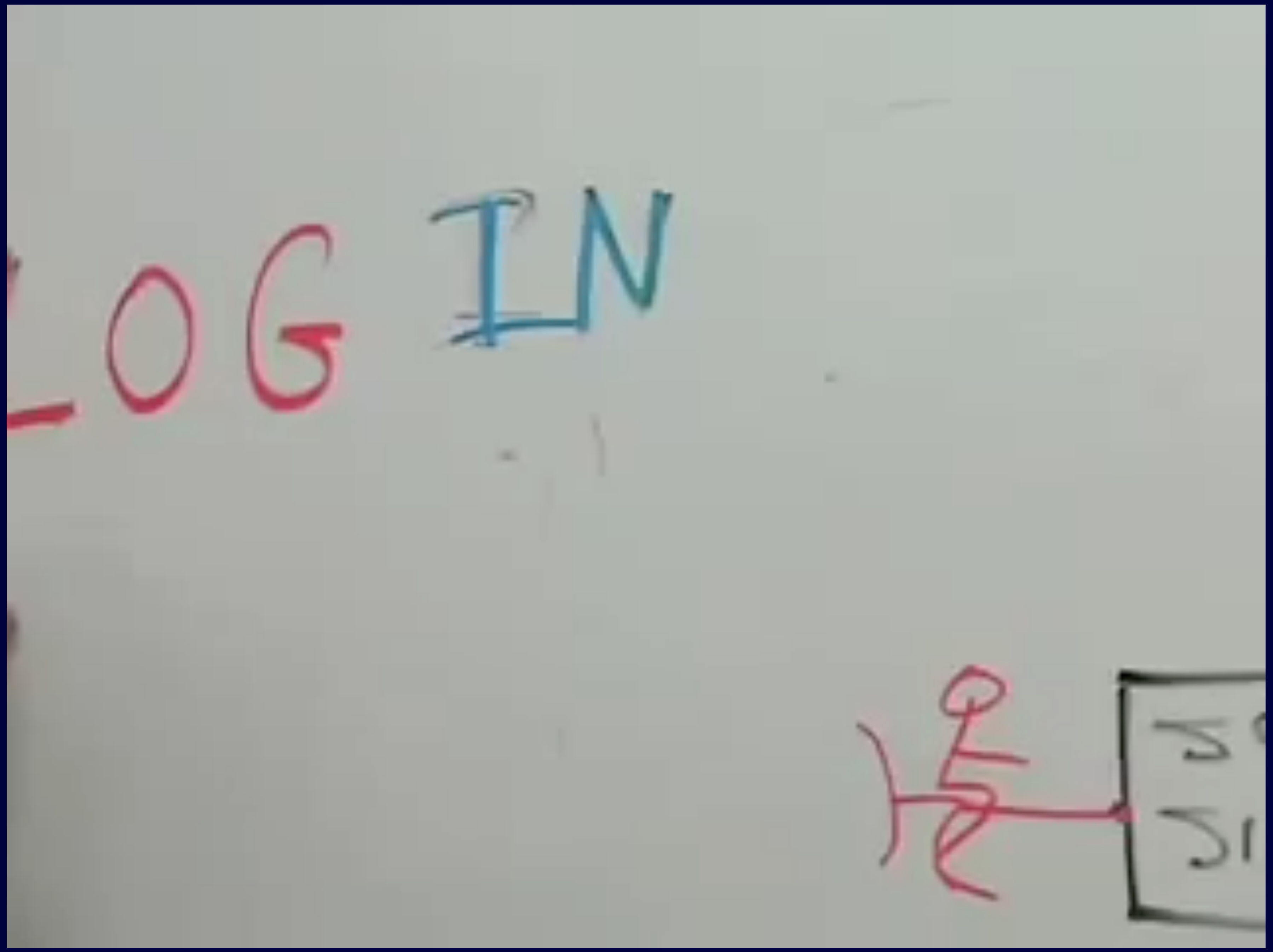
The start of electronic comms

What hath GOD wrought?

(1844) *First telegraph message, 24 May*. 24 May. [Manuscript/Mixed Material] Retrieved from the Library of Congress, https://www.loc.gov/item/mcc.019/.

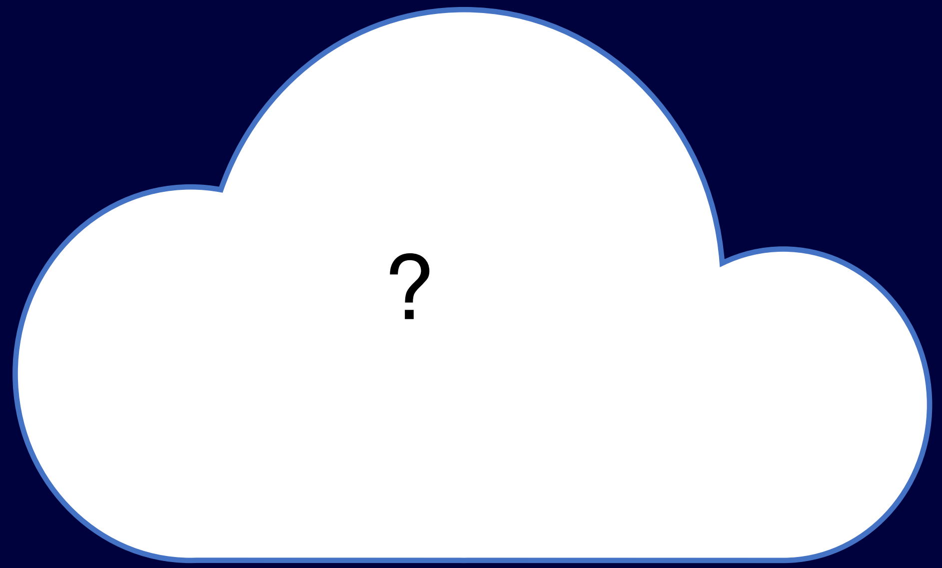
The start of the internet

The message was meant to be “login” but the system crashed after two characters….



The first Internet connection, with UCLA's Leonard Kleinrock
https://www.youtube.com/watch?v=vuiBTJZfeo8&t=390s

Datacomms versus telecoms

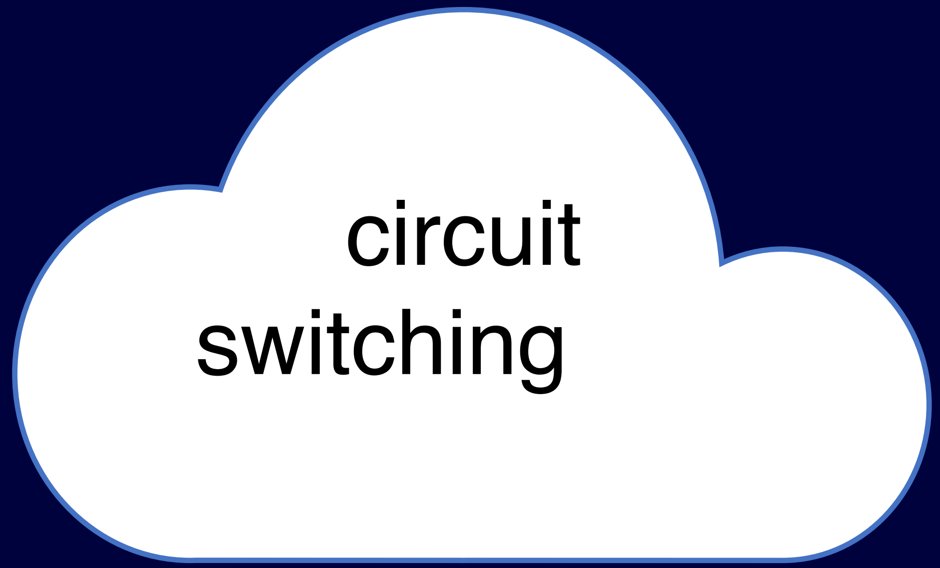


Data are bursty

No causality

More bandwidth = faster transmission

Data loss intolerable



Data are streamed

Signals are causal

Signal has finite bandwidth

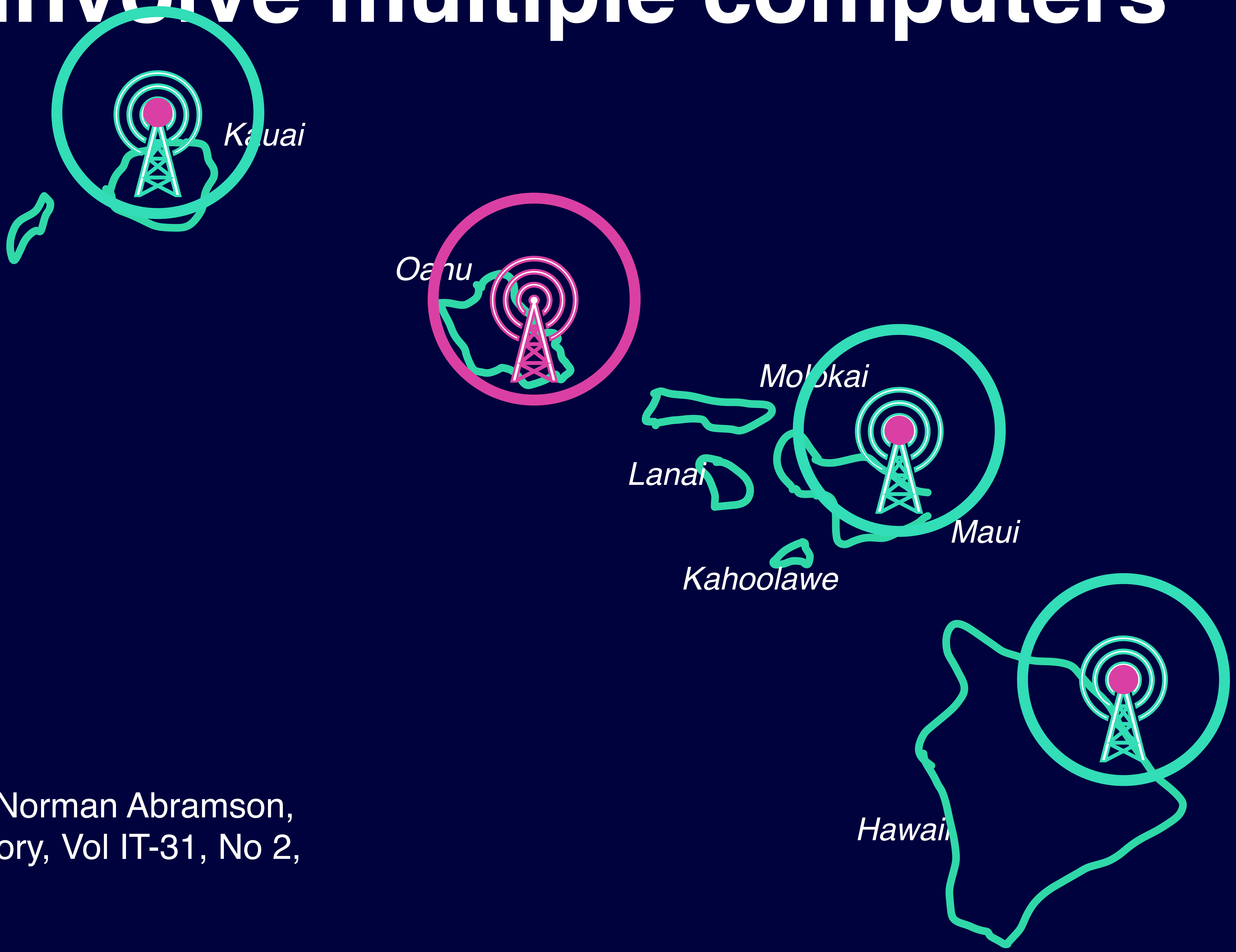
Some signal loss may be tolerable

Point-to-point links well known …

"Datel modem with telephone",
BT Digital Archives,
Finding number TCB 4 17/E 30065,
9th December 1964

http://www.digitalarchives.bt.com/Calmview/Record.aspx?src=CalmView.Catalog&id=BTA%2f5+PH%2f1%2f1%2f30%2fE+30065&pos=13

but networks involve multiple computers



The Development of the ALOHANET, Norman Abramson, IEEE Transactions on Information Theory, Vol IT-31, No 2, March 1985, pp 119 - 123

Byzantine, or other, generals

IP

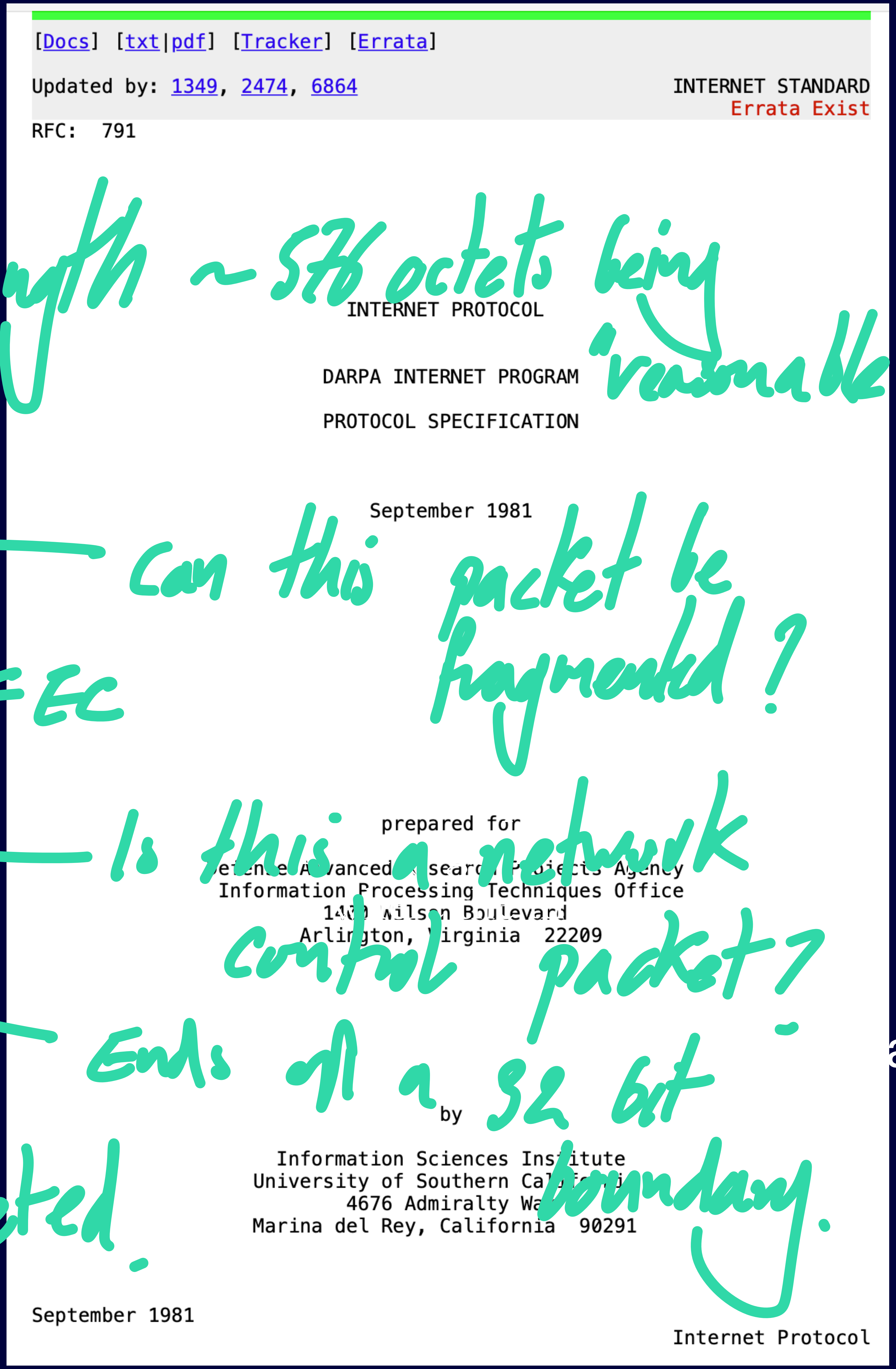
A summary of the contents of the internet header follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Version|  IHL  |Type of Service|          Total Length         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Identification        |Flags|      Fragment Offset    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Time to Live |    Protocol   |         Header Checksum       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Source Address                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Destination Address                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Options                    |    Padding    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example Internet Datagram Header

Figure 4.

Note that each tick mark represents one bit position.


[Docs] [txt|pdf] [Tracker] [Errata]

Updated by: 1349, 2474, 6864 INTERNET STANDARD

Errata Exist

RFC: 791

INTERNET PROTOCOL

DARPA INTERNET PROGRAM

PROTOCOL SPECIFICATION

September 1981

prepared for

Defense Advanced Research Projects Agency

Information Processing Techniques Office

1400 Wilson Boulevard

Arlington, Virginia 22209

by

Information Sciences Institute

University of Southern California

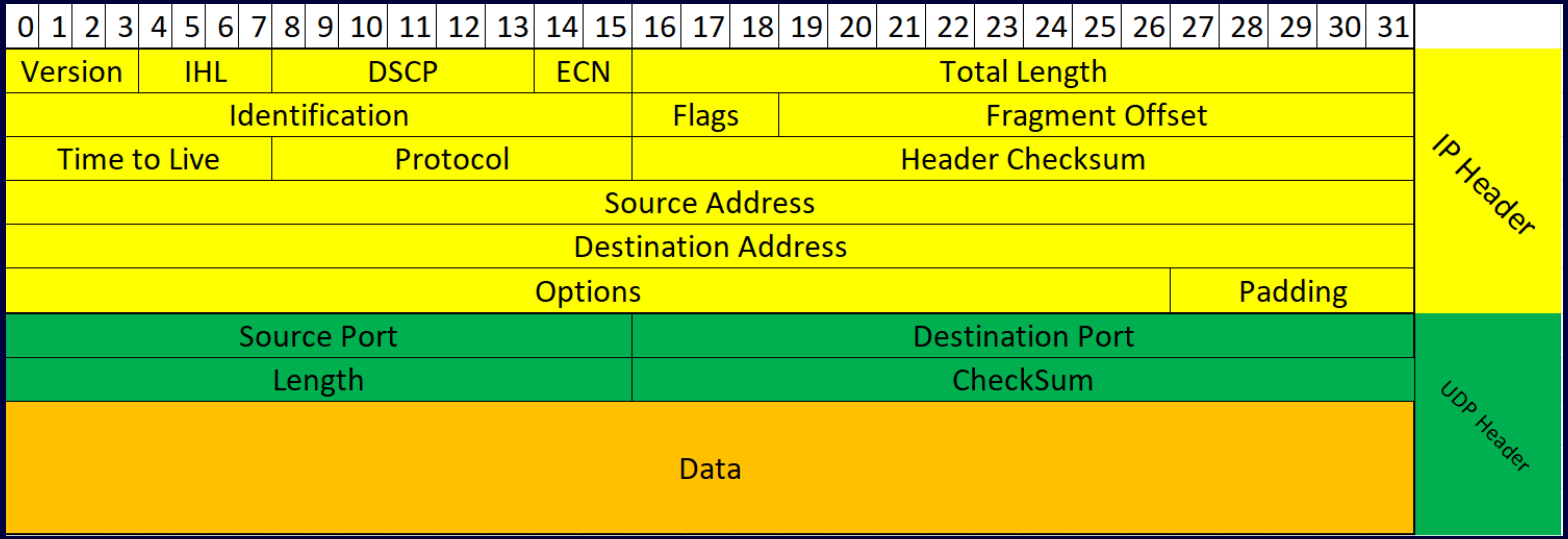
4676 Admiralty Way

Marina del Rey, California 90291

September 1981 Internet Protocol

- 4 for IPv4, 6 for IPv6.
- header length
- header + data length ~ 576 octets being "reasonable"
- Can this packet be fragmented?
- FEC
- Is this a network control packet?
- Security / routing / etc.
- Ends on a 32 bit boundary.
- used to assemble fragments.
- after which packet may be deleted.

UDP: a packet within a packet



UDP is a “fire and forget” protocol

Forward Error Correction (FEC)

- Can you send a message that knows if it has been corrupted?
- Can you send a message that carries enough information to correct any errors?
- Simplest form - parity - send an extra bit which is 1 if there are odd numbers of 1s (odd parity) (or even parity if preferred)
- Next simplest - checksum - add the number of ones and send that.
- A topic in its own right: Hamming codes and their brethren
- FEC essential when there is no back channel (as with UDP) but, when we have a backchannel we can use ARQ.

TCP is like a bucket-brigade

TCP is an ARQ protocol

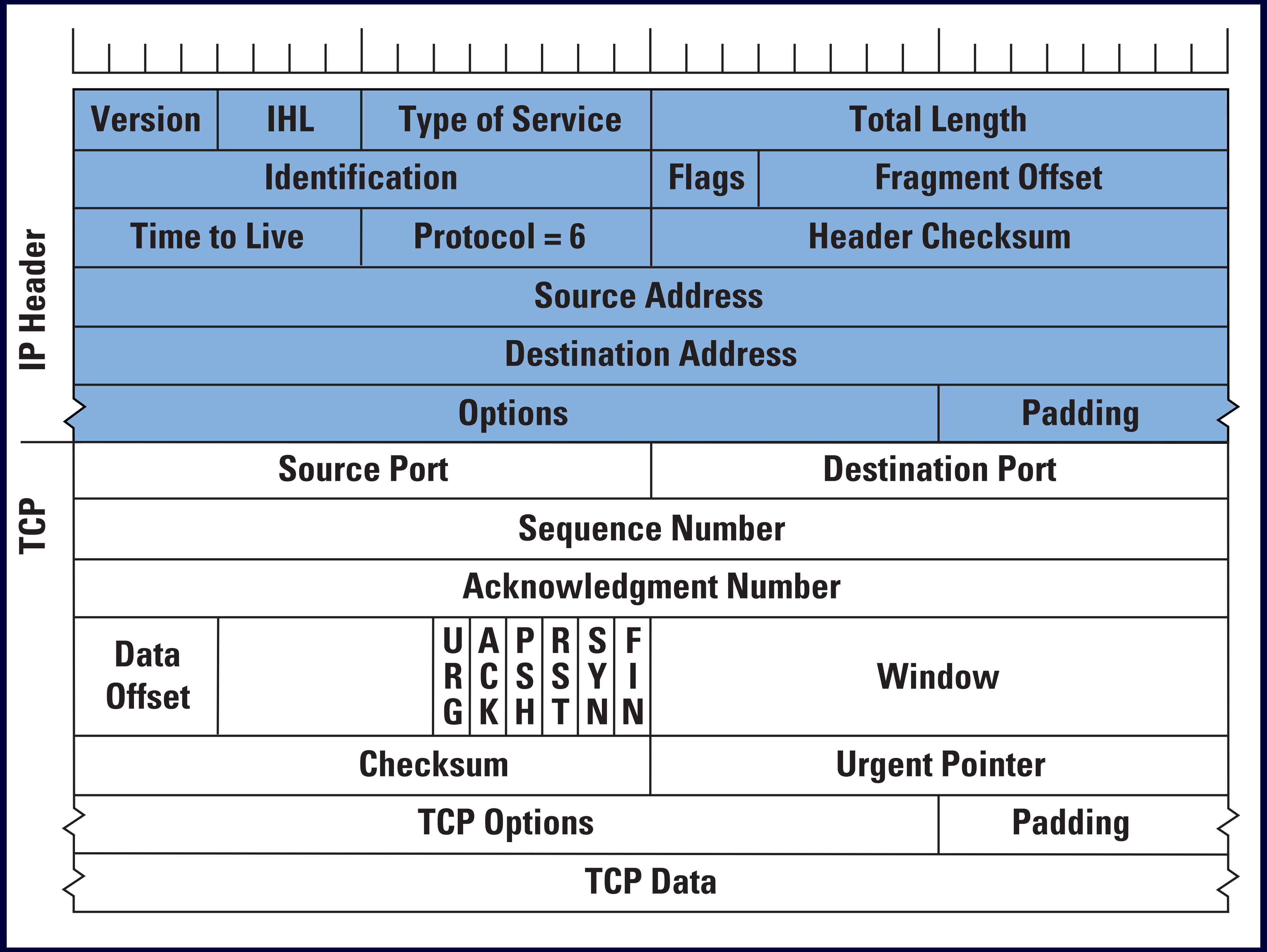


Figure 1 from *The Internet Protocol Journal*, June 2000, Volume 3, Number 2 published by Cisco Systems Inc

The three-way handshake

```
1) A --> B  SYN my sequence number is X
2) A <-- B  ACK yr sequence number is X
3) A <-- B  SYN my sequence number is Y
4) A --> B  ACK yr sequence number is Y
```

Handshake diagram from RFC 793

Implementing TCP/IP

RFC 1149 IP over Avian Carriers (IPaAC)

One of the many essential steps in the creation of what would become known as the internet occurred in 1968 when ARPA contracted BBN Technologies to build the first routers, known as Interface Message Processors or IMPs, which enabled ARPANET to become operational the following year. (Photo courtesy of Steve Jurvetson under a CC BY 2.0 license)

Immediate questions?

- How does anyone know an address?
- How do we manage congestion?
- Isn’t it slightly risky that anyone along the route can read the packets?

Immediate questions?

- How does anyone know an address?

DNS, static and dynamic addresses, NAT and IPv6

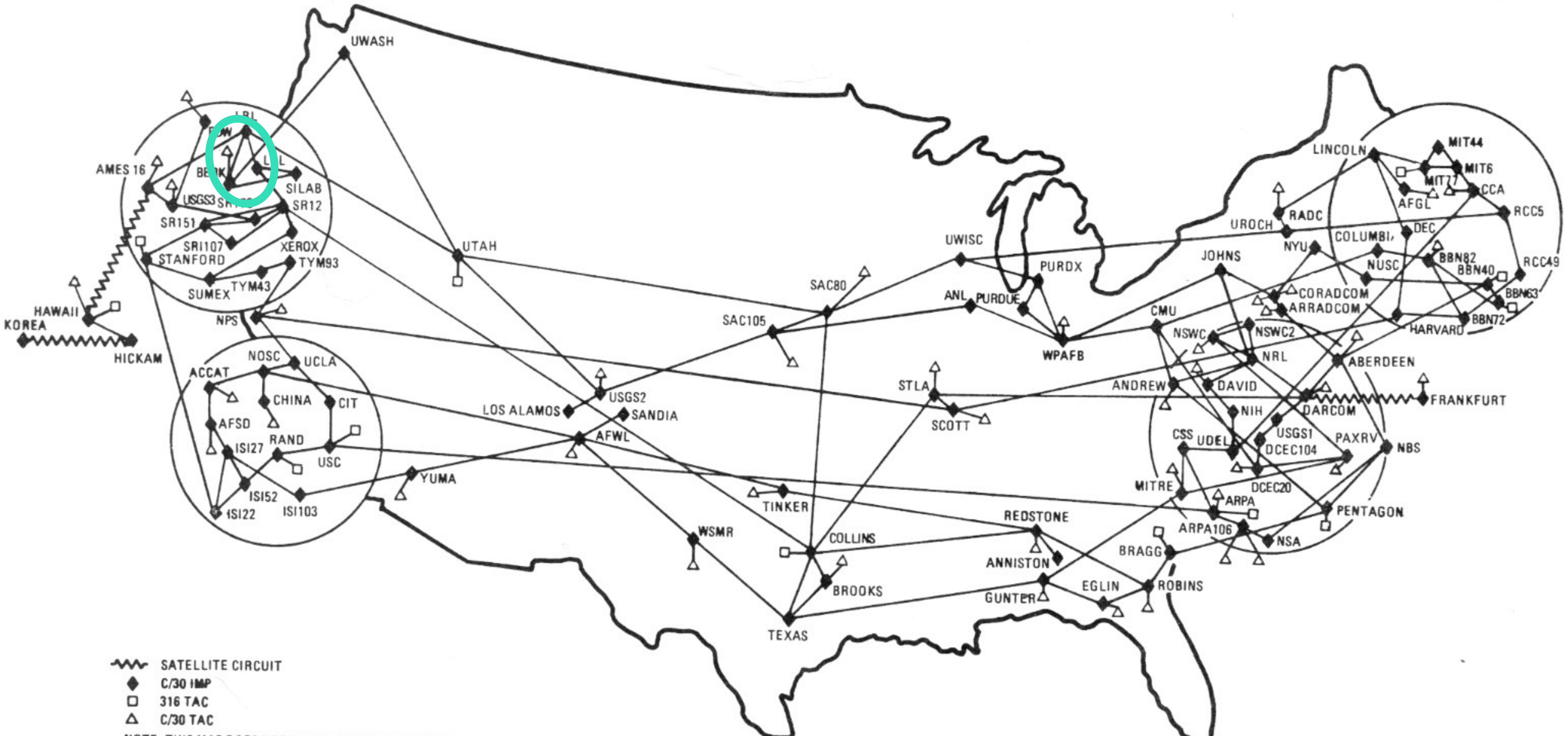
- How do we manage congestion?

Congestion control - coming up

- Isn't it slightly risky that anyone along the route can read the packets?

Yes! See later

ARPANET/MILNET GEOGRAPHIC MAP, APRIL 1984



NOTE: THIS MAP DOES NOT SHOW ARPA'S EXPERIMENTAL SATELLITE CONNECTIONS.
NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES

TINKER= OKLAHOMA CITY

A TCP window

A TCP window

A larger TCP window

A larger TCP window

TCP windows

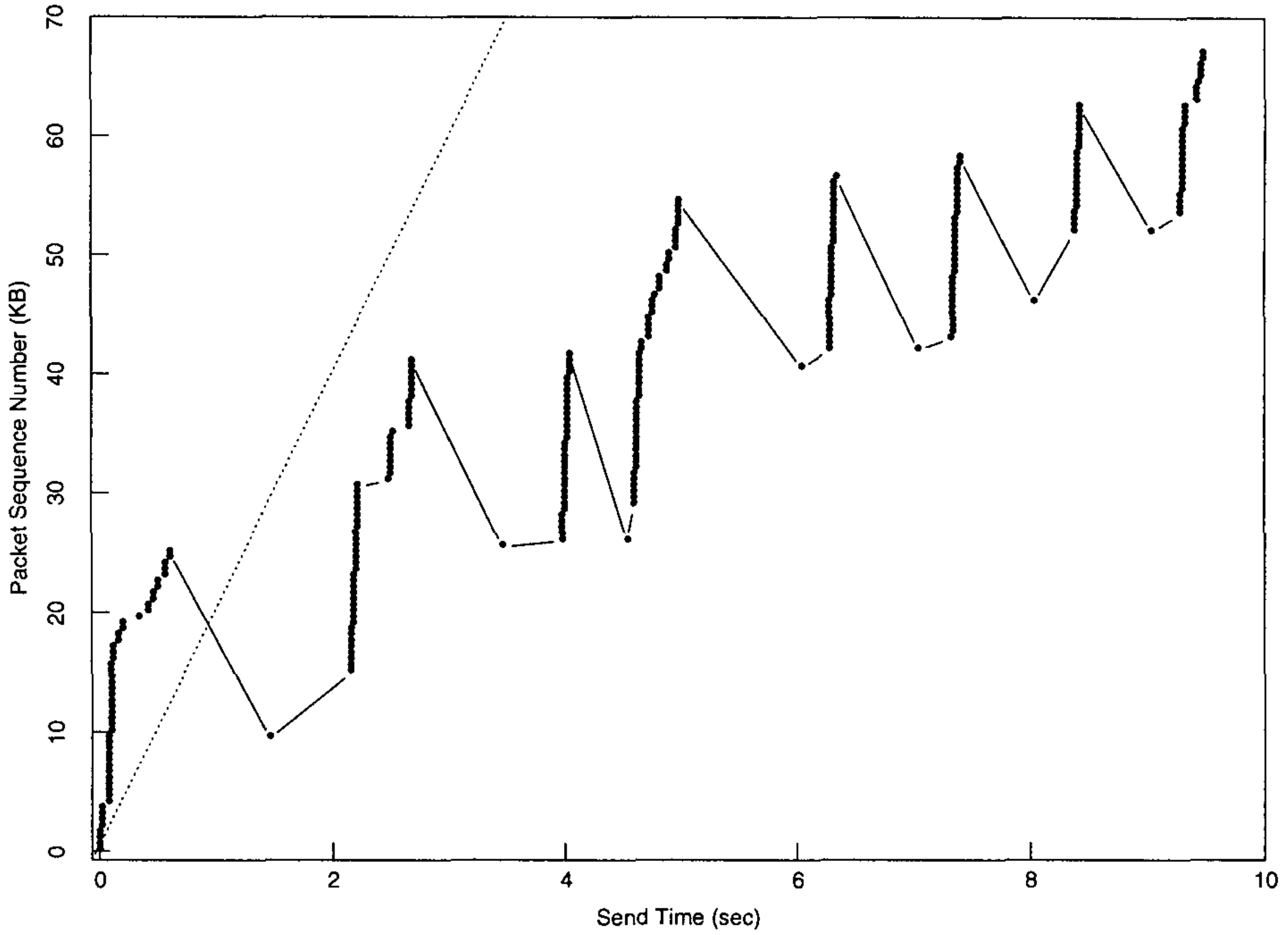
- TCP is "self-timed" via the ACKs
- Larger windows make more efficient use of the link

But…

- Large bursts of data can either increase collisions or cause buffers to overflow

- Which will lead to lots of retransmits.

TCP crashed the internet in 1986



Trace data of the start of a TCP conversation between two Sun 3/50s running Sun OS 3.5 (the 4.3BSD TCP). The two Suns were on different Ethernets connected by IP gateways driving a 230.4 Kbs point-to-point link (essentially the setup shown in fig. 7).
Each dot is a 512 data-byte packet. The x-axis is the time the packet was sent. The y-axis is the sequence number in the packet header. Thus a vertical array of dots indicate back-to-back packets and two dots with the same y but different x indicate a retransmit. 'Desirable' behavior on this graph would be a relatively smooth line of dots extending diagonally from the lower left to the upper right. The slope of this line would equal the available bandwidth. Nothing in this trace resembles desirable behavior.
The dashed line shows the 20 KBps bandwidth available for this connection. Only 35% of this bandwidth was used; the rest was wasted on retransmits. Almost everything is retransmitted at least once and data from 54 to 58 KB is sent five times.

Figure 3: Startup behavior of TCP without Slow-start

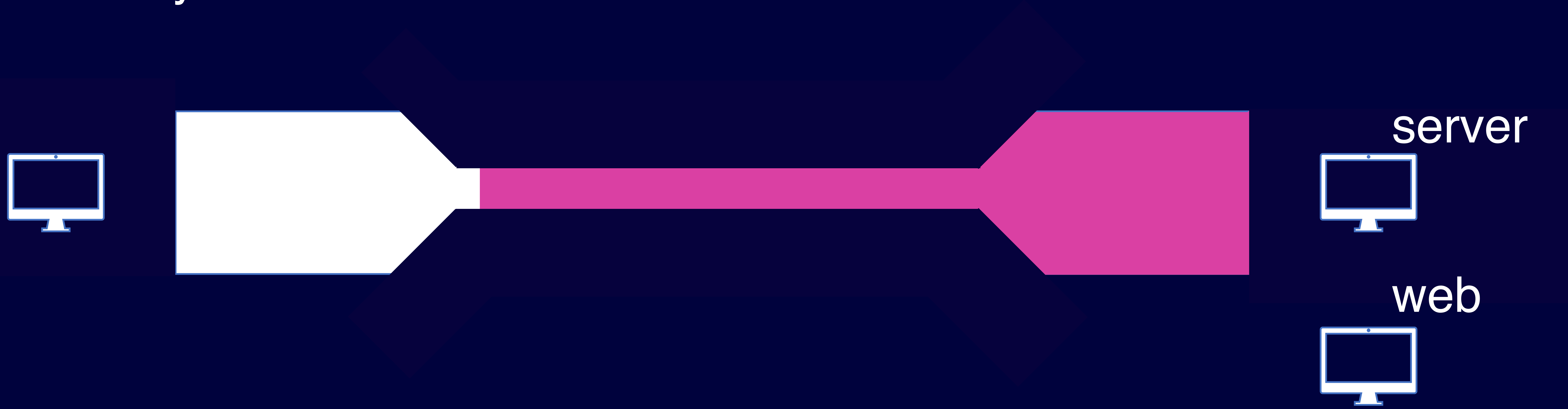
From "Congestion avoidance and control", Vin Jacobsen, ACM SIGCOMM, Vol 18, No 4, August 1988

TCP Tahoe: AIMD

1. start transmitting single packets
2. if ACK received then step-up to 2 packets
3. if ACK received then increase to 3 packets
4. repeat additive increase until we reach receiver's advertised buffer size
5. continue transmission until we lose an ACK
6. deduce congestion
7. halve window and got to Step 2.

Congestion is real but rarefied

- viz “bufferbloat” (see bufferbloat.net Jim Gettys & Dave Taht)
- Large hidden buffers (“dark” buffers) cause havoc with the latency



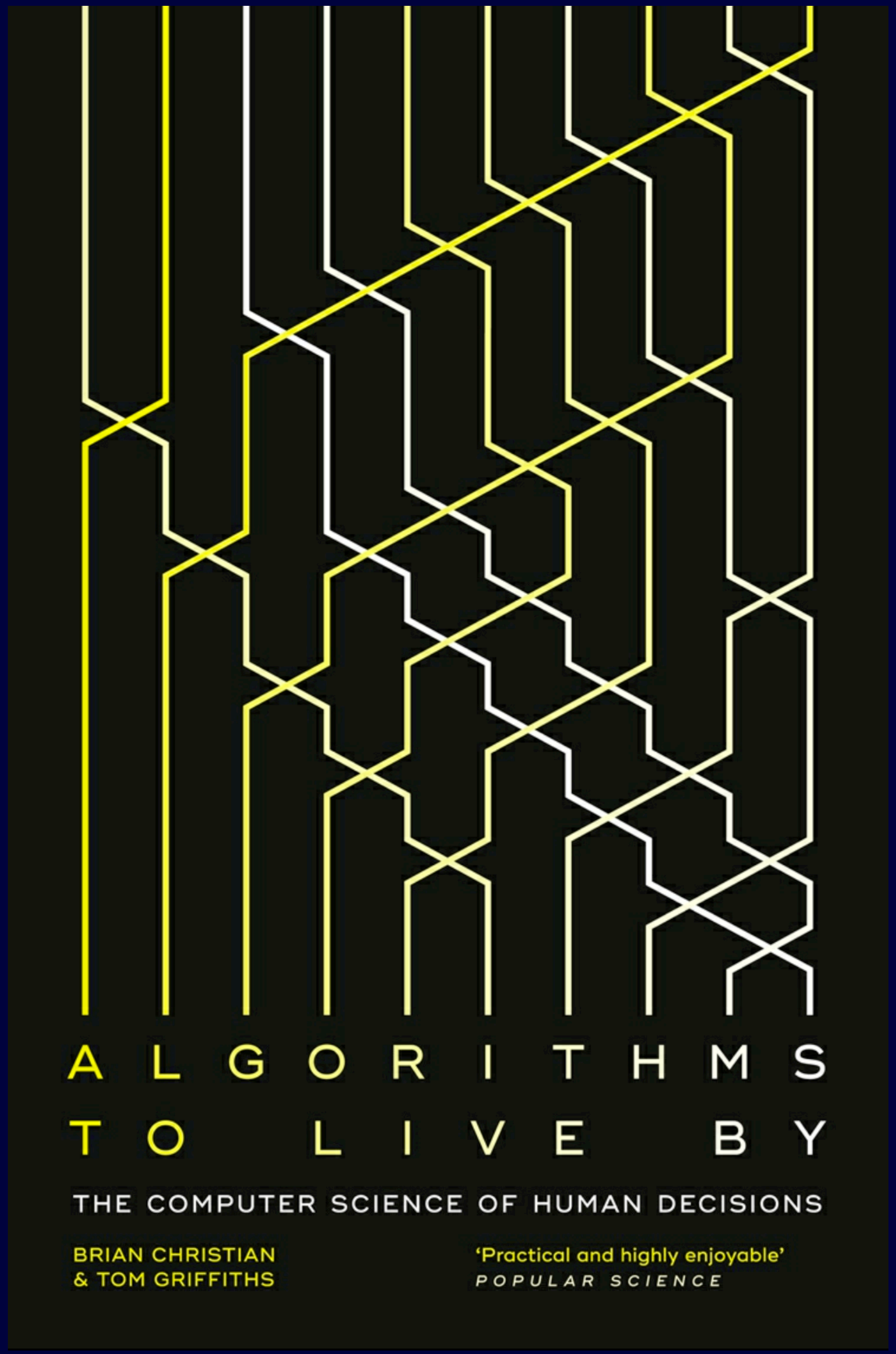
Taildrop - an under-rated solution

Use a short buffer and drop packets when its full

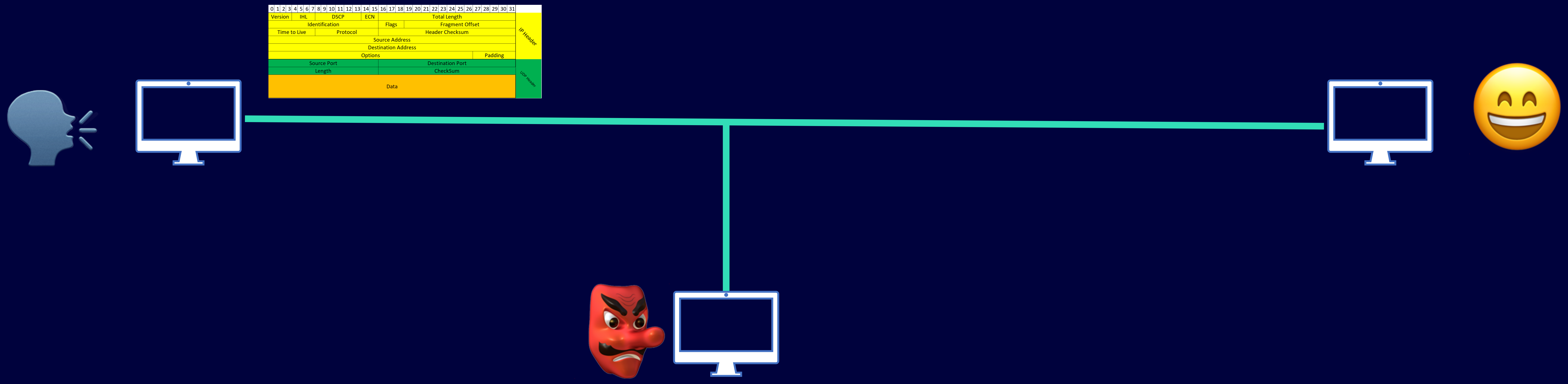
Dropped packets trigger the multiplicative decrease

TCP adapts and latency recovers

Taildrop is useful in other situations……

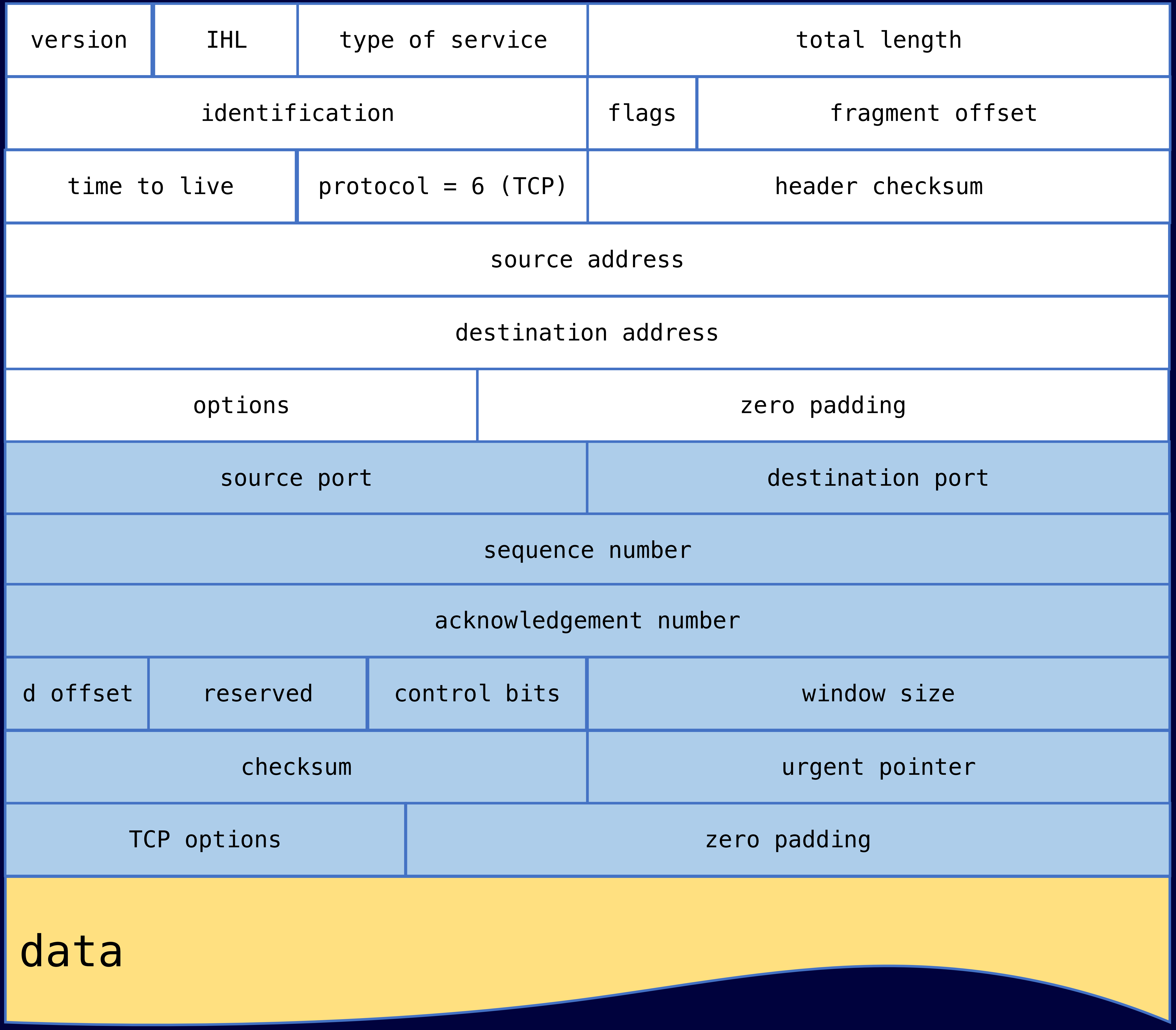


What about security?

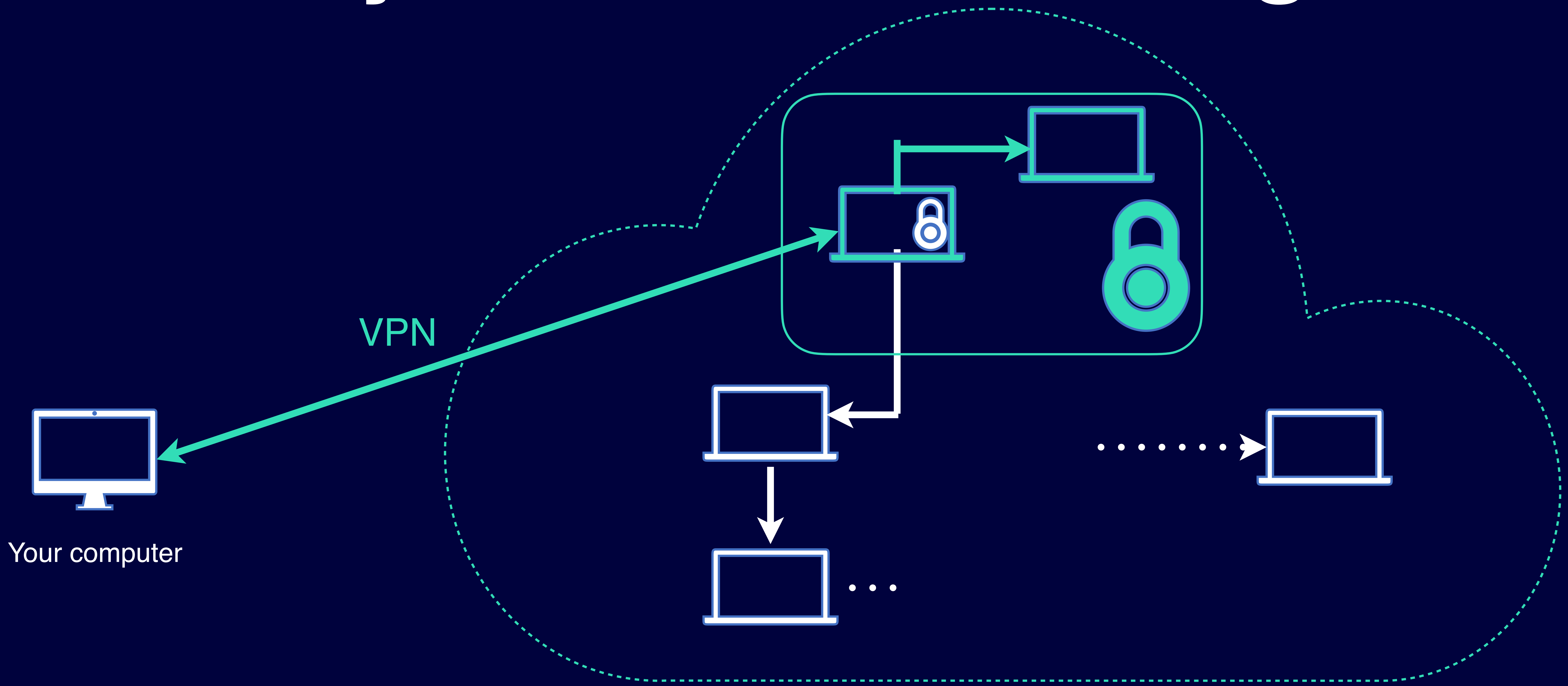


“Man in the middle” attack

Security solution 1: encryption



Security solution 2: routing



The Onion Router (Tor)

eg: I measured 10 hops from my house to www.google.com with a latency of 6ms

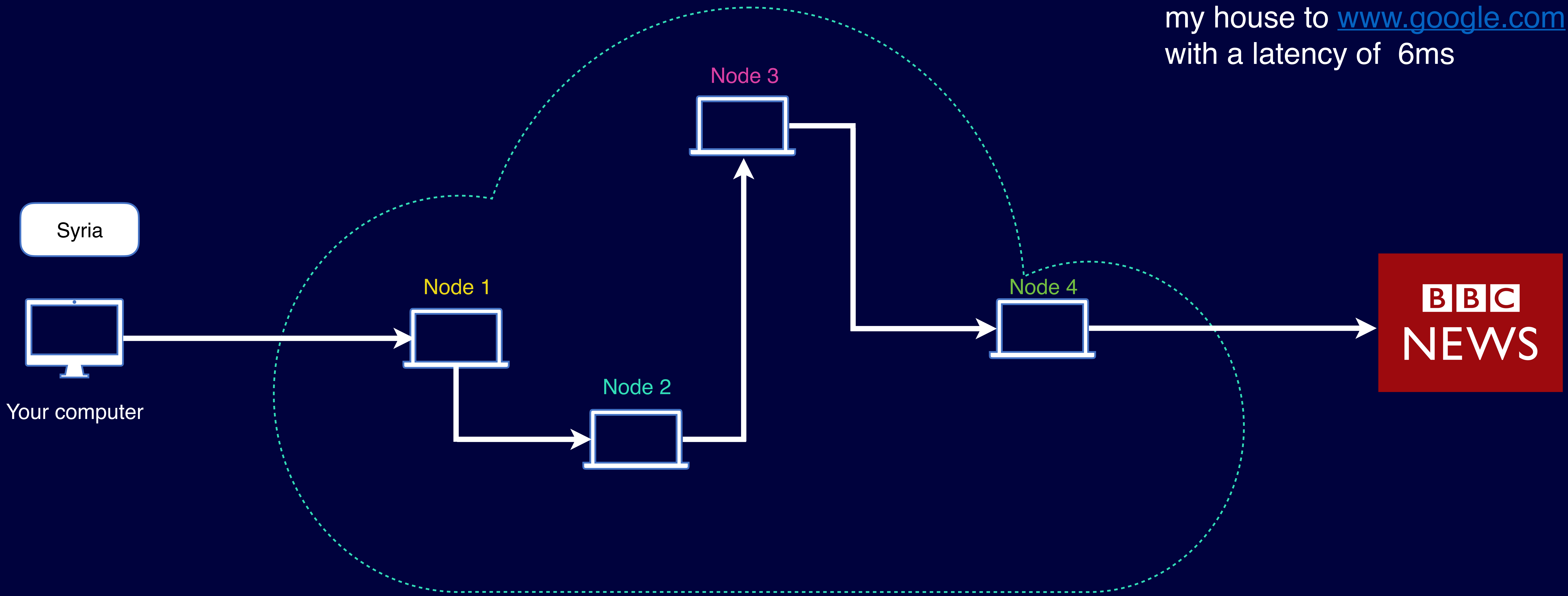


Diagram adapted from “How does Tor really work,” https://skerritt.blog/how-does-tor-really-work/

The Onion Router (Tor)

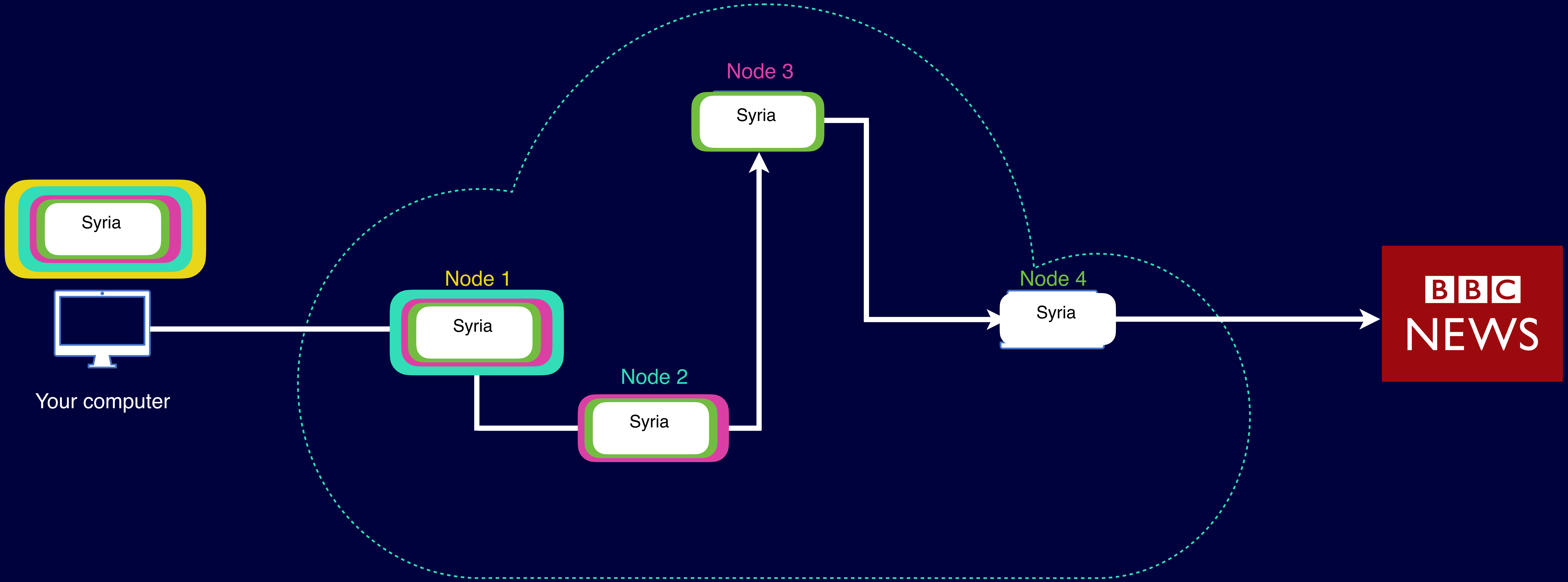


Diagram adapted from “How does Tor really work? The Definitive Visual Guide (2020),” https://skerritt.blog/how-does-tor-really-work/

Emerging themes

Wireless: enormous consumer pressure but … congestion is tricky

Latency: the Achilles heel of TCP

IoT: see lecture by Martin Thomas

Security and privacy: deserves a lecture in its own right

Next lecture

The future of computer security

25th May 2021 at 18:00 (6pm)

Thanks and kudos to the Worshipful Company of Information Technologists who sponsor these lectures.