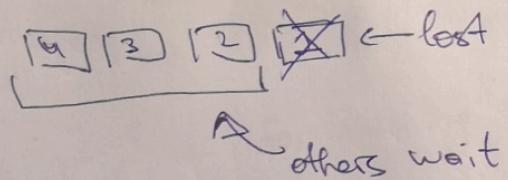


UDP/TCP

UDP

No Head-of-line blocking! (HOL)

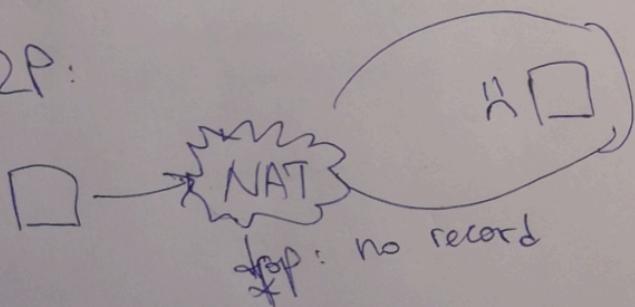


UDP/TCP

□ → X
packet drop
why?

may be NAT dropped your record
fix: keepalive

P2P:



Channel capacity:

$$C = BW \cdot \log_2 \left(1 + \frac{S}{N} \right)$$

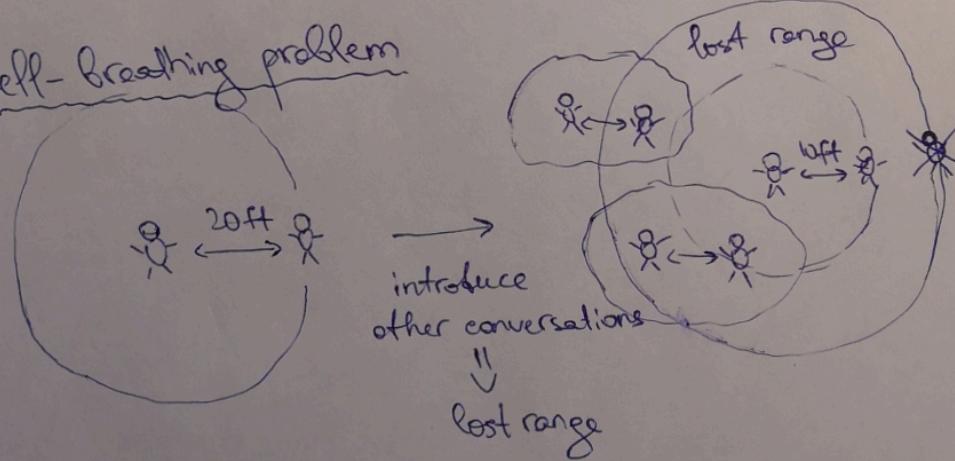
Bandwidth available (hertz)
noise

Bandwidth

~~x2 BW for wireless technology $\Rightarrow \times 2$ bitrate~~

Low-frequency: travel farther, require large antennas, many clients competing for access.

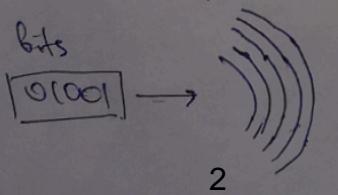
High-frequency: won't travel far, can transfer more data

Signal powerCell-breathing problemNear-far problem

Strong signal makes it impossible for the receiver to detect weaker signal

Real-World analogy: crowded party.

modulation scheme matters:



HTTP History

HTTP/0.9: One-Line Protocol (1991)

Request:

<method> <path> single ASCII string

Response: HTML as ASCII character stream

<html><head>...

- ASCII
- over TCP/IP link
- designed for HTML docs
- connection closed after every request

HTTP/1.0: Informational RFC (1996)

Visually looks like modern HTTP.

Key changes:

- Headers in request and response
- Status line in the response
- Response formats (not only HTML)

These changes enabled new features: content encoding, multi-part types, authorization, caching, proxy behaviors.

} Requiring TCP connection per request imposes performance penalty because of the Three-Way Handshake and Slow Start

HTTP/1.1: Internet Standard (1999)

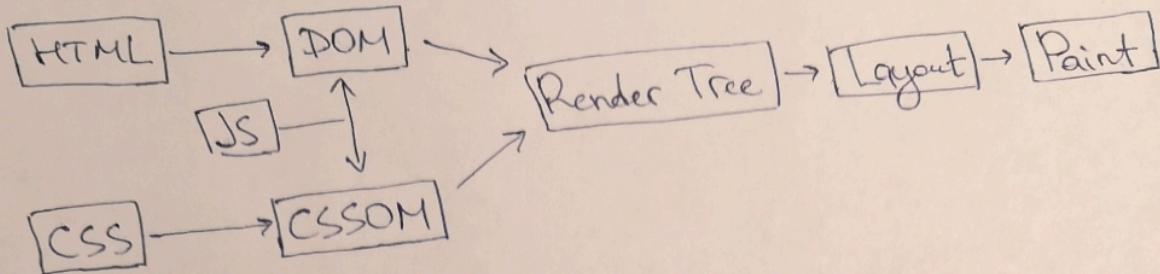
The most important feature from the point of view
of performance is keep-alive option.

keep-alive \leftrightarrow reuse TCP connection for future requests

HTTP/2: Improving Transport Performance

Primer on Web Performance

Browser architecture



Human perception

0-100ms: instant

100-300ms: small perceptible delay

300ms+: loading

In the most cases latency is a bottleneck,
not the bandwidth

Navigation Timing API is useful for
real-user performance metrics

Browser Optimization

- Document-aware optimization (e.g. resource prioritization)
- Speculative optimization (e.g. pre-fetching)