# Selection criteria for the next generation of TCP

Netlab, Caltech
Presenter: Lachlan Andrew
Team: Steven Low, David Wei and others

# Time horizon

- TCP Reno (and friends) lasted 20 years
- Design next generation to last as long

- Scalability vital; reflect this in benchmarks
  - 1Gbps common, 1Tbps cutting edge
- Future backward compatibility
  - Router-assisted has to be compatible with it
- Fix more problems than bandwidth-delay
  - Wireless?  Slow start?

# TCP friendliness is medium term

- New protocol *must* work "OK" with Reno
  - E.g., neither should starve the other
- Time between OS upgrades is years not decades
  - Fairness with *itself* is more critical (e.g. RTT fairness)
  - RTT unfairness means even TCP is not TCP friendly

- Benchmarking:
  - background traffic Reno *and*
  - background traffic of the new protocol *and*
  - mixtures

# Wireless / lossy links

- TCP Reno suffers on wireless links
  - TCP solutions like Westwood, Veno, …
  - Workarounds like “performance enhancing proxies”
  - Break end-to-end semantics (like encryption)
- Wireless Gigabit is already possible
  - Will have lossy links on fast long distance routes

# Wireless / lossy links

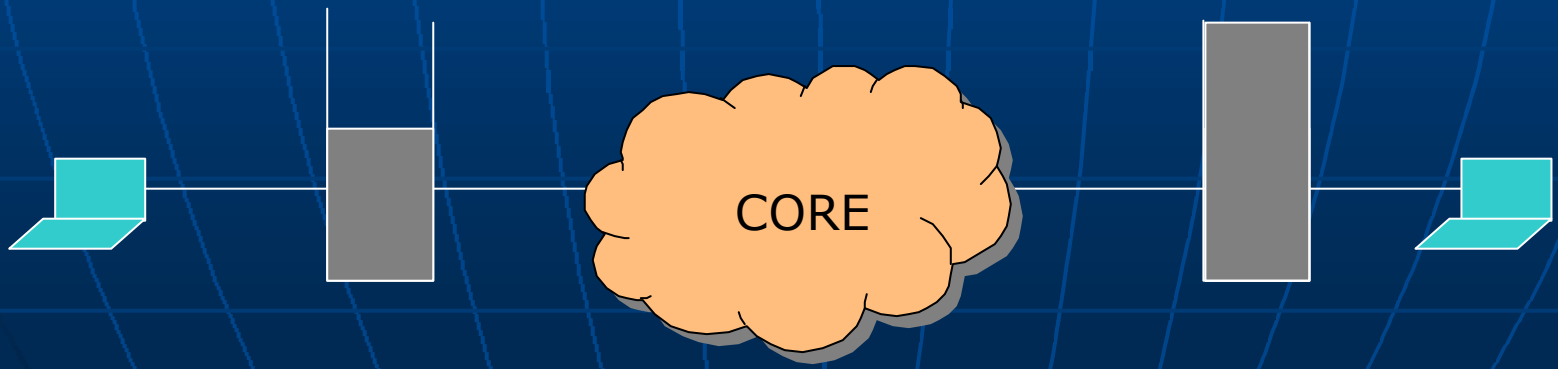
- TCP Reno suffers on wireless links
  - TCP solutions like Westwood, Veno, …
  - Workarounds like “performance enhancing proxies”
  - Break end-to-end semantics (like encryption)
- Wireless Gigabit is already possible
  - Will have lossy links on fast long distance routes
- Two impairments (either or both)
  - *Loss* passed on to TCP
  - *Random delay* due to link layer retransmissions

# Wireless / lossy links

- TCP Reno suffers on wireless links
  - TCP solutions like Westwood, Veno, …
  - Workarounds like "performance enhancing proxies"
  - Break end-to-end semantics (like encryption)
- Wireless Gigabit is already possible
  - Will have lossy links on fast long distance routes
- Two impairments (either or both)
  - *Loss* passed on to TCP
  - *Random delay* due to link layer retransmissions
- Wireless will use "multi-user diversity"
  - Wants packets from all flows buffered
    - Wired links just want packets from at least one flow
  - Further increase in jitter

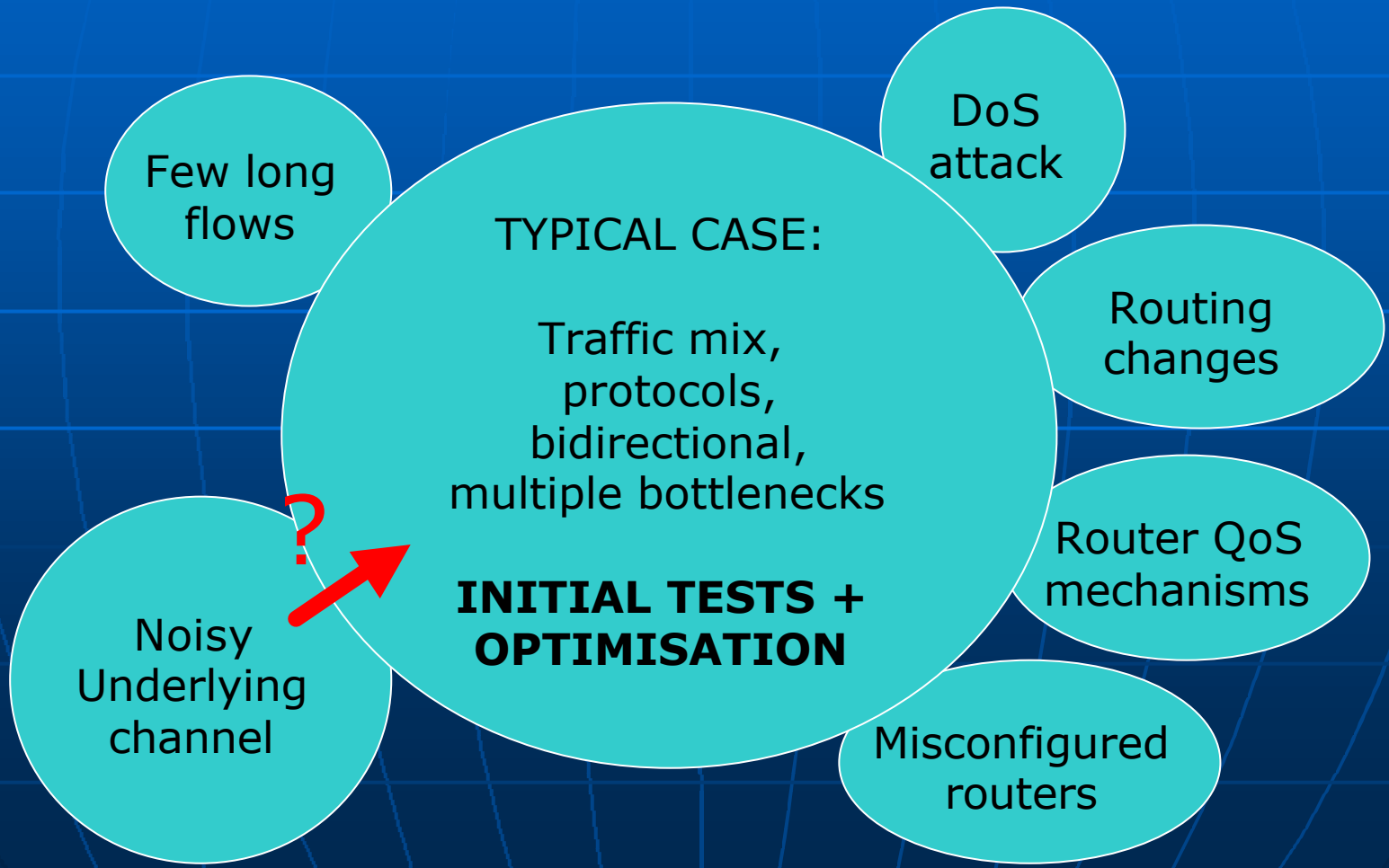
# Multiple bottlenecks

- Rarely have a single bottleneck
  - Typically access network at each end of the path
- Single bottleneck tests
  - illusion that buffer overflow = maximum observed RTT
  - over-emphasise synchronised loss
- Need balance
  - Single-link tests take less infrastructure, simulation time



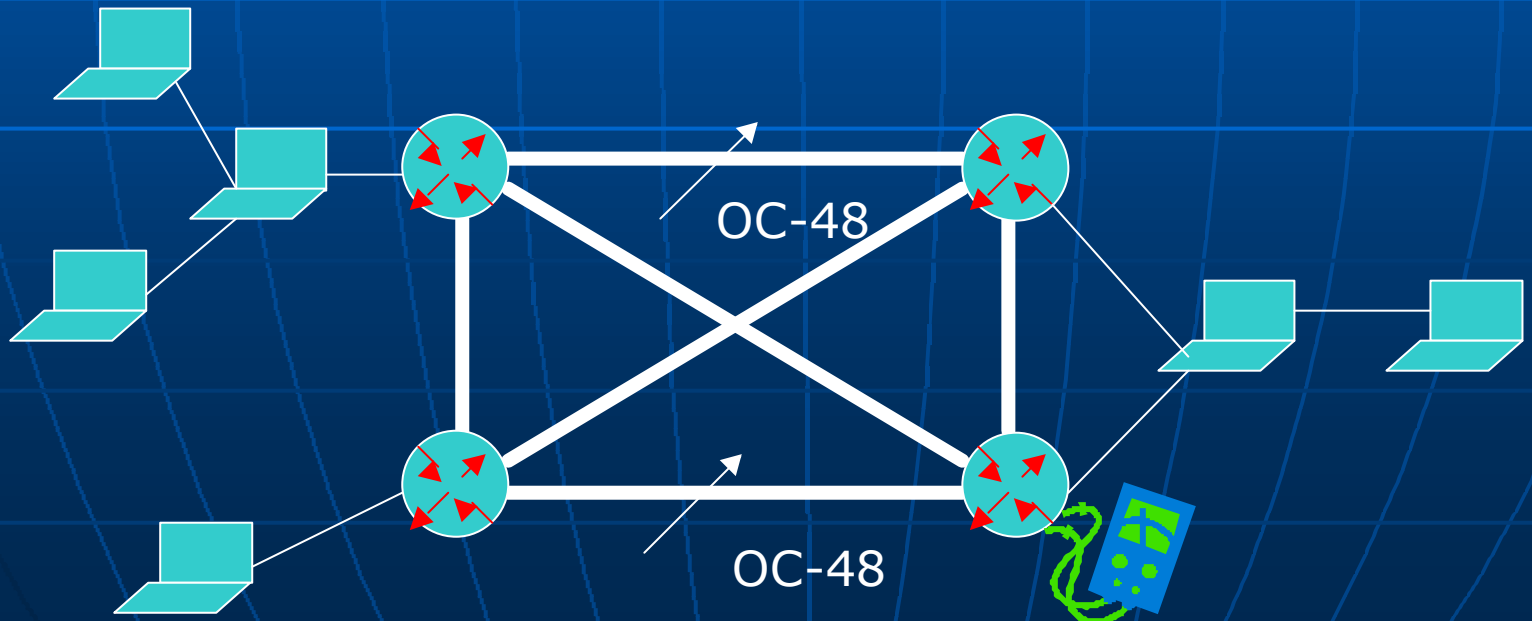
# Optimise typical, check corners

- Two-level benchmark suite – “core” and “corners”



# WAN-in-Lab testbed

- Dummynet and simulation introduce artifacts
- Also need to test on *real* equipment
- WAN with real delays, located in a single room
  - Connected to an external WAN (Ultralight)
- Open for the community to use for benchmarking



# WAN-in-Lab capabilities

| Current | Planned |
| --- | --- |
| Two 2.5G bottlenecks<br>Multiple 1G bottlenecks | Six 2.5G bottlenecks |
| Two "real" delays<br>(Emulate cross traffic delay) | Up to six "real" delays |
| | |
| End-to-end RTT, drop | Per-router delay, drop<br>(movable DAG cards) |

# Conclusions

- The new "TCP friendly" requirement
  - Response function should scale beyond now
- Consider all of TCP's weaknesses
  - Not just fast long-distance networks
- Multiple bottleneck topologies

- Simulation, emulation and real networks
  - WAN-in-Lab