University of Glasgow

School of Computing Science

60 YEARS OF COMPUTING AT GLASGOW

Some Thoughts on the Future of Internet Transport

Colin Perkins

Slides for panel at IFIP Networking workshop on Future of Internet Transport, 12 June 2017

New applications drive transport evolution…

- Multimedia, augmented- and virtual-reality → latency sensitive
  - AQM, ECN, congestion control → don't fill the queues
  - Path discovery, NAT traversal, happy eyeballs for connection racing, multihoming, multipath, peer-to-peer → find and use "best" available path
  - Time-bounded partial reliability, avoiding HoL blocking → new APIs & UDP-based protocols
- Privacy and confidentiality:
  - Snowden revelations → "encrypt all the things", but is this the right model?
  - Technical capabilities and societal norms still evolving
  - How to manage/debug the network? How to support legitimate security service requests?
- Security and robustness:
  - Design APIs and protocols to reduce vulnerabilities to hacking – make use of new language features and type systems, avoid risky protocol constructs
  - Reflect protocol messages, types, and states in APIs and code structure – implementations should be validated as matching the specifications
  - Generating parsing and serialisation code from a specification should be a solved problem

…within the constraints of an ossified network

- Must recognise that network ossification is a good thing:
  - It means the Internet succeeded
  - The network is critical infrastructure – it should be hard to change
  - Network ossification, backwards compatibility, is essential – even if it complicates protocol evolution
- Packet formats must be retained, as must protocol semantics that are visible to the network
- End-to-end behaviour that doesn't impact the network can change
  - Layer boundaries can change; packet formats can be reinterpreted – UDP as a substrate?

How to Manage Transport Evolution?

- The rate of change in protocols and APIs is increasing
- Backward compatibility and feature interactions lead to increasingly baroque designs

- We can envisage new protocols – but can we keep up?
  - How can we rapidly and effectively write *correct* protocol specifications?
  - How can we ensure implementations match those specifications?
  - Do we have the right tools to design, validate, and implement protocols for the future?