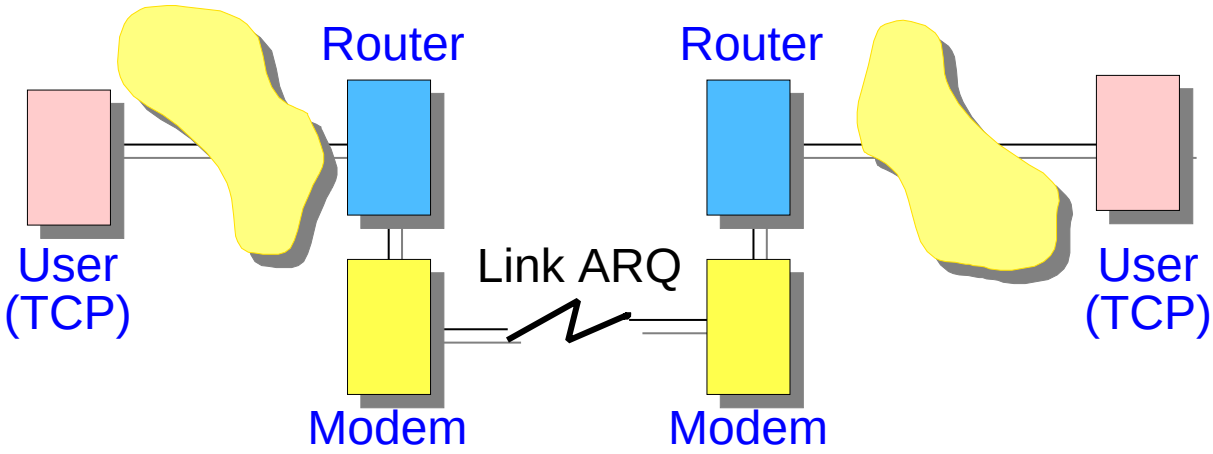
Link ARQ issues for IP traffic
draft-ietf-pilc-link-arq-issues-01.txt

Gorry Fairhurst
Department of Engineering
University of Aberdeen

Lloyd Wood
Cisco Systems Ltd

PILC WG
IETF-50
Minneapolis, USA



ARQ Persistence

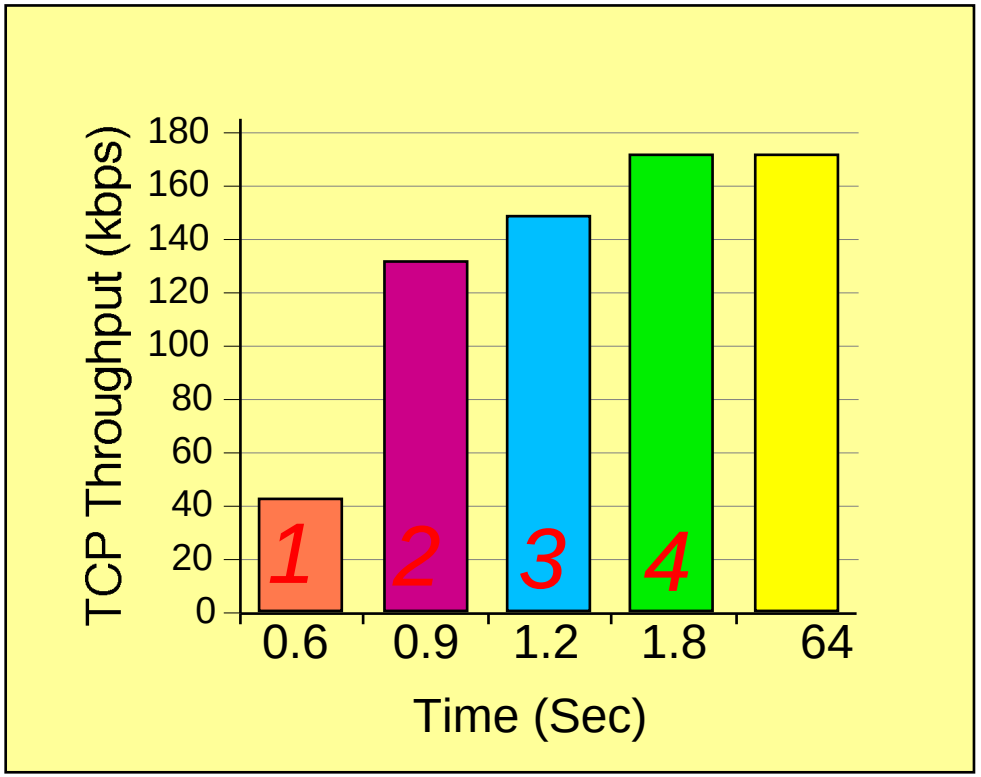
G. Fairhurst & L Wood, IETF-50, Minneapolis

IP doesn't *require* strict reliability

IP flows benefit from: (i) low loss
(ii) timely delivery

Types of link ARQ:

- None
- Low Persistence (e.g. 802.11)
- High Persistence (e.g. irDA)
- Perfect Persistence (e.g. HDLC)



Average throughput for one TCP bulk flow (5 MB)
Link rate = 2 Mbps,
Frame size = 52 B,
Link RTT = 600 ms
Frame error rate = 0.1

Persistency needed depends upon anticipated error rate / duration

Edits applied to -01

Many small “fixes” to wording

Incorporated feedback to list / authors

Clarification of persistence in shared links

- Ethernet example changed
- Persistence impacts utilisation

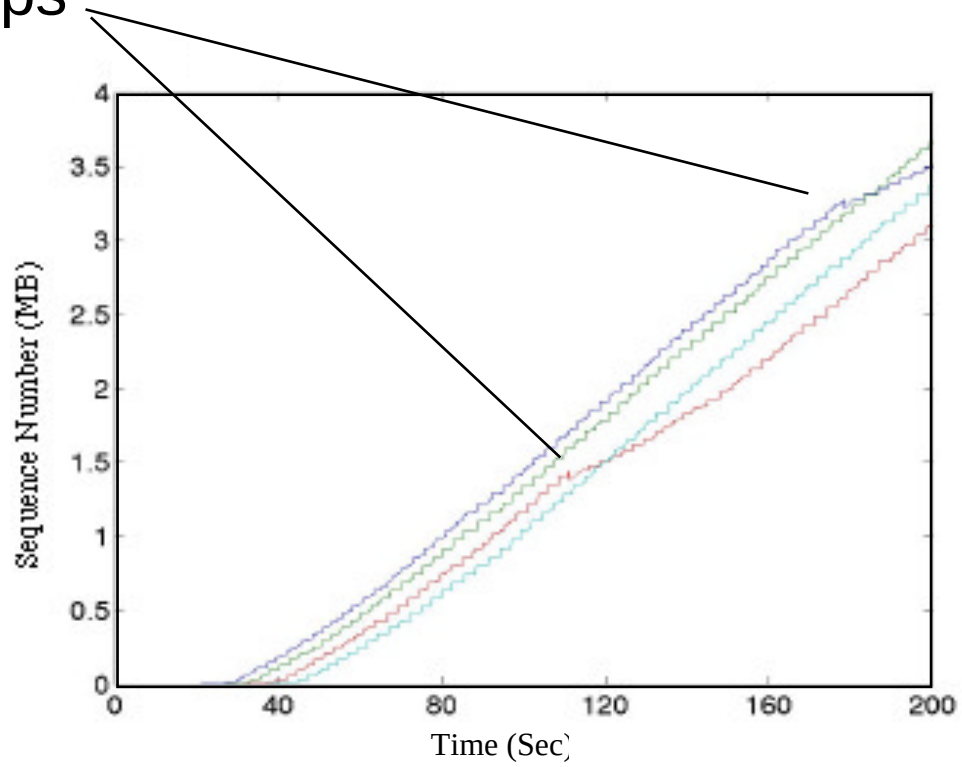
Eliminated 64 sec constraint

- Not clear how this applies to link layer

Key Issue 1: Sharing - Low Persistence

G. Fairhurst & L Wood, IETF-50, Minneapolis

Speed bumps



Low persistence ARQ, 4 TCPs
Link rate = 2 Mbps,
Frame size = 52 B,
Link RTT = 600 ms
Frame error rate = 0.2

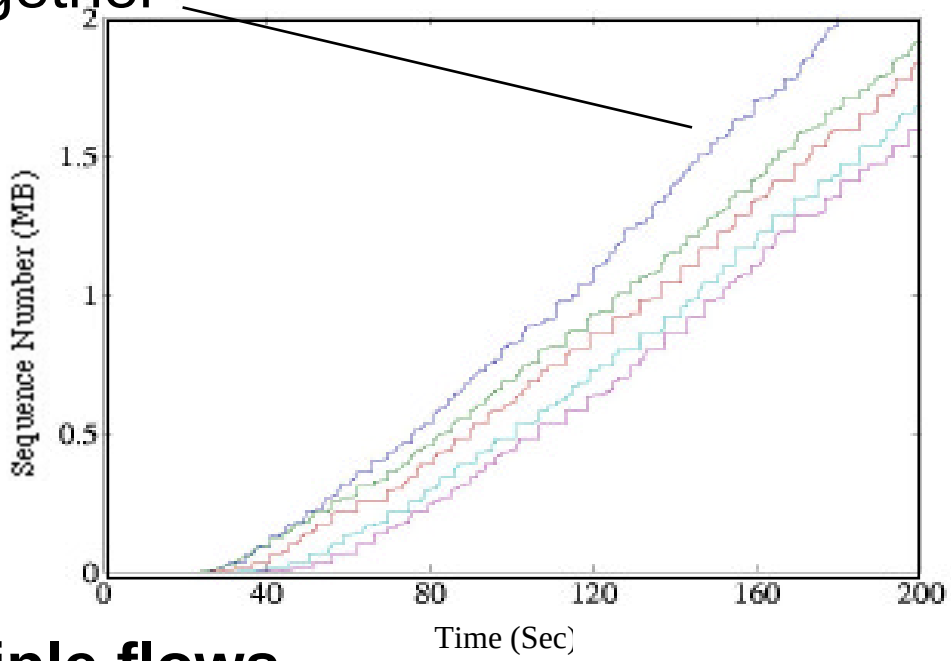
Single link, multiple flows

- Bounded impact on path RTT
- Some loss
- Speed bumps

Key Issue 1: Sharing - High Persistence

G. Fairhurst & L Wood, IETF-50, Minneapolis

All flows suffer together



High persistence ARQ, 4 TCPs
Link rate = 2 Mbps,
Frame size = 52 B,
Link RTT = 600 ms
Frame error rate = 0.2

Single link, multiple flows

- Link ARQ jitter impacts all sessions
- Reduction in throughputs of other sharing flows

Proposed solutions with high persistence

- Requires “fine grain” differentiation, per flow processing
- Research issue with large numbers of flows

Bumps & Bursts

G. Fairhurst & L Wood, IETF-50, Minneapolis

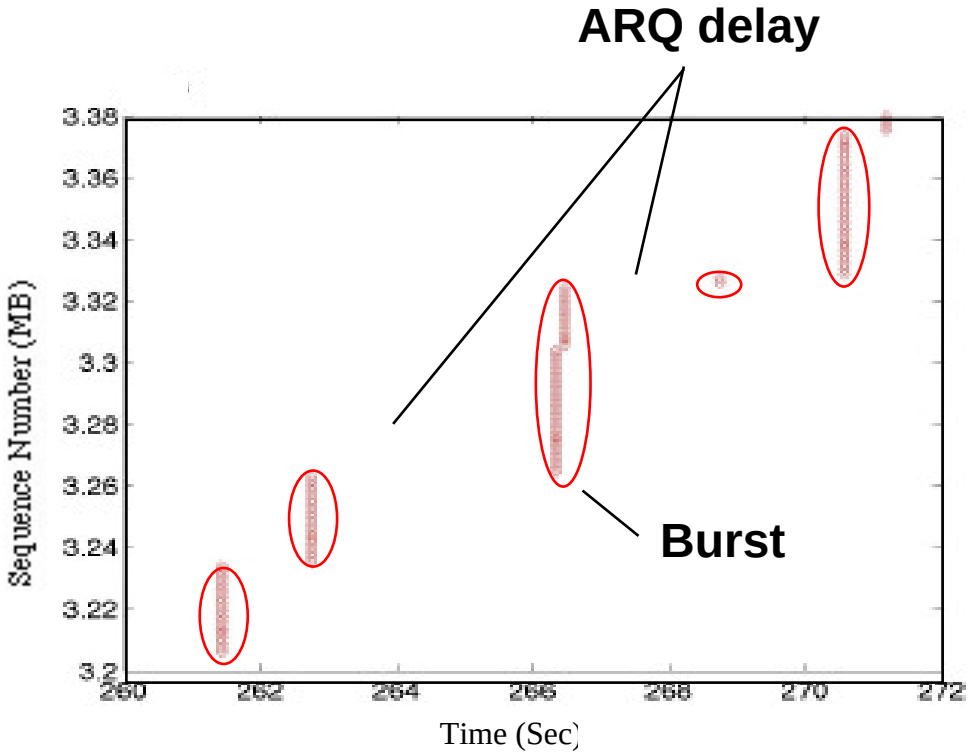
High Persistence ARQ (§2.2)

cwnd remains open
RTO grows with increased link jitter
"Microscopic" TCP transmit bursts

Low Persistence ARQ (§2.3)

cwnd reduces after TCP retransmission
Bounded impact on RTO
"Macroscopic" speed bumps

Loss reduces *average throughput*



TCP with High Persistence ARQ

High persistent ARQ, Single TCP
Link rate = 2 Mbps,
Frame size = 52 B,
Link RTT = 600 ms
Frame error rate = 0.2

Key Issue 2: Classification

Not all applications need high persistence ARQ

- Delay-sensitive flows suffer (e.g. RTP/UDP)

Implicit differentiation is a hard problem (ARQ § 3.2)

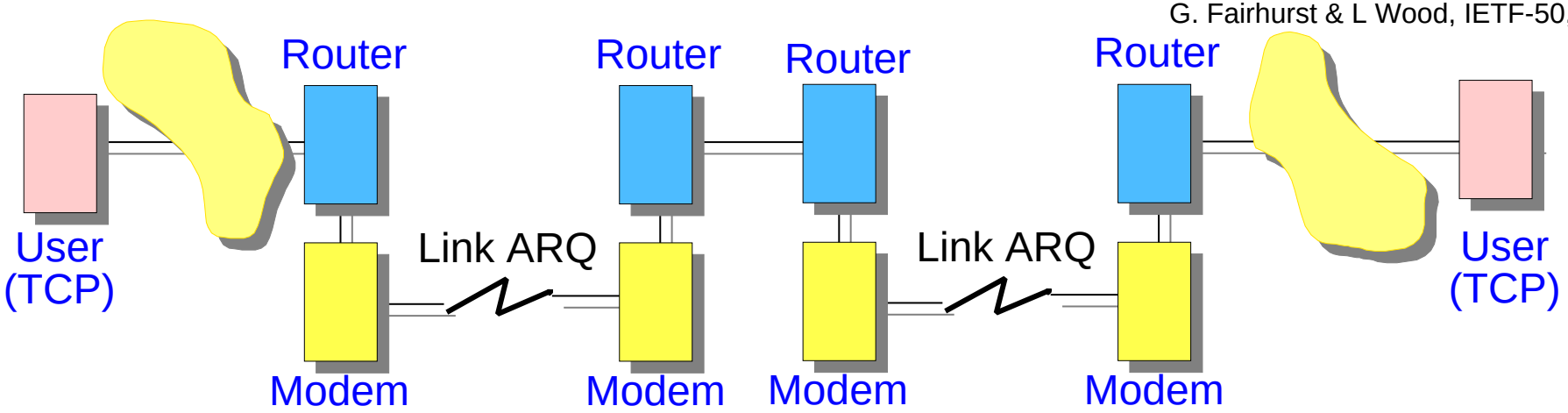
- New applications require adding new interpreters
- Cost per packet needs considered (not fast-path decision)

How does link map flow to ARQ behaviour?

- Flow type does not imply ARQ persistence (semantic gap)

Without this, difficult to advocate hi-persistent approach

Key Issue 3: Multiple Links along Path



Today's edge link is tomorrow's transit-to-a-cloud link

Don’t know how many links along path

- After RTO, TCP will give up / retransmit
- Can’t be sure of the path delay

There may also be congestion loss

Link ARQ shouldn't adversely delay end-to-end feedback

- TCP congestion control, ECN, TFRC ...

Key Issue 4: Shared Channel

Persistence usually low

- Stability
- Shadowing effects

Variable retransmit delay

- Need to prevent congestion: Back-off delay
- “cost” of retransmission: Access delay

Many different schemes

Recommendations

Link ARQ is a useful tool (among others)

Low Persistence:

- Simpler (and fewer buffers)
- More predictable
- Safe

High Persistence:

- More complexity (e.g. per-flow ARQ, Classifiers)
- Set of caveats

Flow Management:

- Improves sharing between IP flows (e.g. per-flow ARQ)

Guidance required to get trade-offs correct

Safest* approach for IP is *low persistence

Edits planned

Clarify perfect persistence - HDLC/irDA example

Clarify MAC wording

Persistence in shared (contention) channels

Outage behaviour (developed from link text)

Impact on multicast, SCTP, RTCP retransmit...

Incorporate any feedback to list / authors

Current ID

draft-ietf-pilc-link-arq-issues-01.txt
(March 2001)