Wireless Communications + Internet: Architecture & Protocols

George Xylomenos

xgeorge@aueb.gr

Mobile Multimedia Laboratory

Department of Informatics

Athens University of Economics and Business

Outline

- Convergence of two technologies
  - Explosive Internet popularity
  - Rapid adoption of wireless networks
- Internet performance over wireless
  - TCP applications (file transfer, web browsing)
  - UDP applications (media distribution)
- Enhancement approaches
- Multi Service Link Layers
  - Multi-protocol, adaptive, QoS aware solution
  - Evaluation of application performance
  - Implicit and explicit service selection

Wireless Systems

- Digital wireless systems
  - Cellular, PCS, 3G/4G
  - Wireless LANs (802.11)
  - LEO/MEO satellites, fixed wireless (802.16)
- Internet protocols: designed for
  - Wired networks: low error rate
    - TCP: any loss means congestion
  - Fixed networks: no mobility, no handoffs
- Physical layer solutions
  - Inflexible: one size fits all
  - Good for telephony, not for data

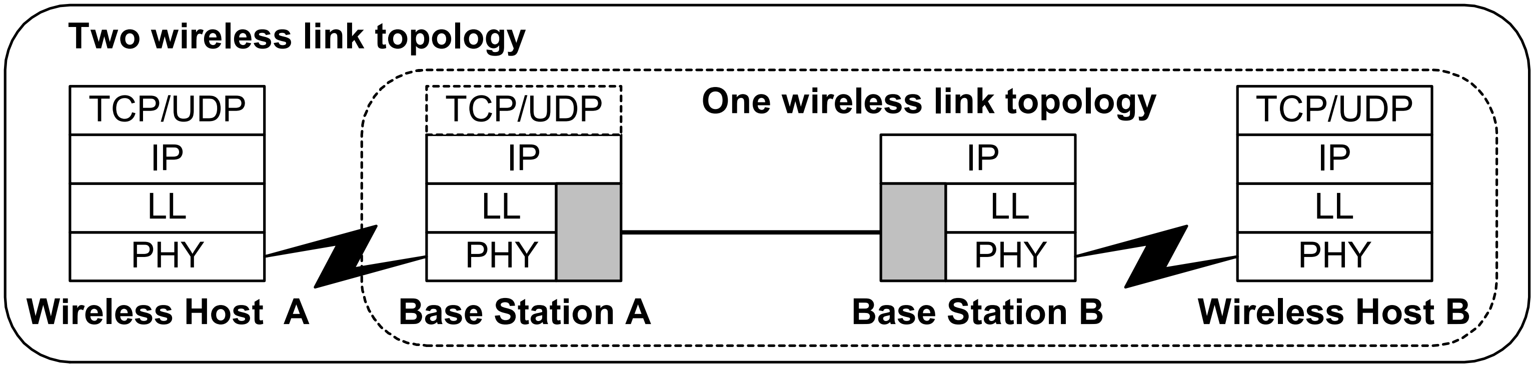
Internet Applications

- Conventional data exchange applications
  - Usually TCP based
  - Error intolerance
  - Delay tolerance
  - Jitter intolerance (TCP)
  - Example: File transfer, web browsing
- Interactive and real-time applications
  - Often UDP based (plus RTP)
  - Often multipoint (IP Multicast)
  - Some error tolerance
  - Delay intolerance
  - Example: Media distribution

Proposed Approaches

- Indirect TCP
  - Violates TCP semantics (not end-to-end anymore)
- Snoop TCP
  - Works well only in the direction towards the mobile
- Modifications to TCP
  - Compatibility: usually both ends need to be updated
  - End-to-end retransmissions for a local problem
  - Non multi-protocol: useless for non TCP applications
- Conventional link-layer schemes
  - Inflexible: one service only
  - Irrelevant for some protocols/applications

Simulation Setup



- Simulations using ns-2 with additions
- Two topologies simulated
  - One wireless link and two wireless links
  - 2 Mbps wired link with 50 ms delay
- HSCSD wireless links (also WLAN)
  - 86.4 Kbps, 100 ms delay, 100 byte packets
  - Independent losses at 1%, 2%, 5% and 10%

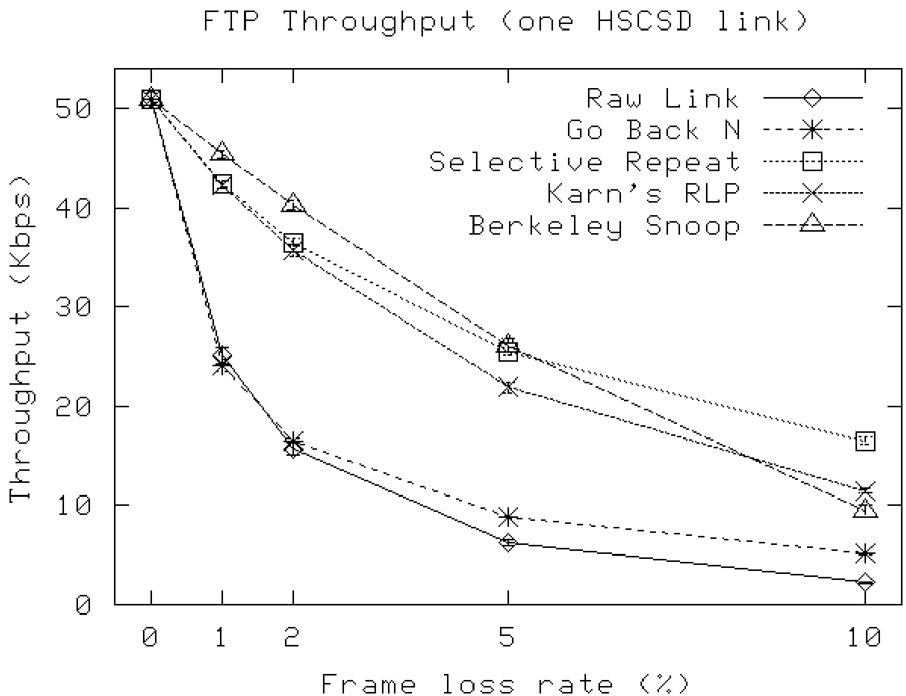
Applications

- File transfer (FTP) over TCP
  - 10 Mbytes from server to client
  - Application level throughput
- WWW browsing (HTTP) over TCP
  - 2000 sec of non-stop single-user transactions
  - Empirical distributions for object sizes
  - Server to client application level throughput
- Continuous media (CBR) over UDP
  - Two-state on-off speech source
  - 14.4 Kbps constant bit rate in active state
  - Residual loss, mean delay + 2 $\times$ standard deviation

Protocols

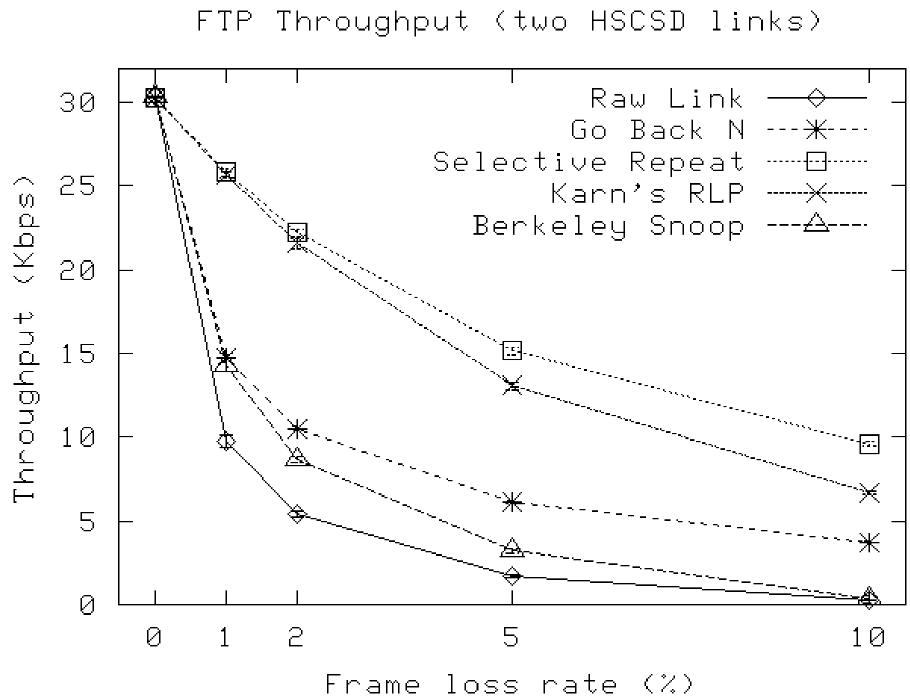
- Raw Link: native link service
- TCP enhancements
  - Go Back N: basic sliding window scheme
  - Selective Repeat: adds selective retransmissions
  - Karn’s RLP: up to 3 retransmissions per frame
  - Berkeley Snoop: TCP aware retransmissions
- UDP enhancements
  - XOR based FEC: 1 parity for 8 data frames
  - Selective Repeat: TCP oriented scheme
  - Karn’s RLP: up to 1 retransmission per frame
  - Out of sequence RLP: variant of Karn’s RLP

File Transfer



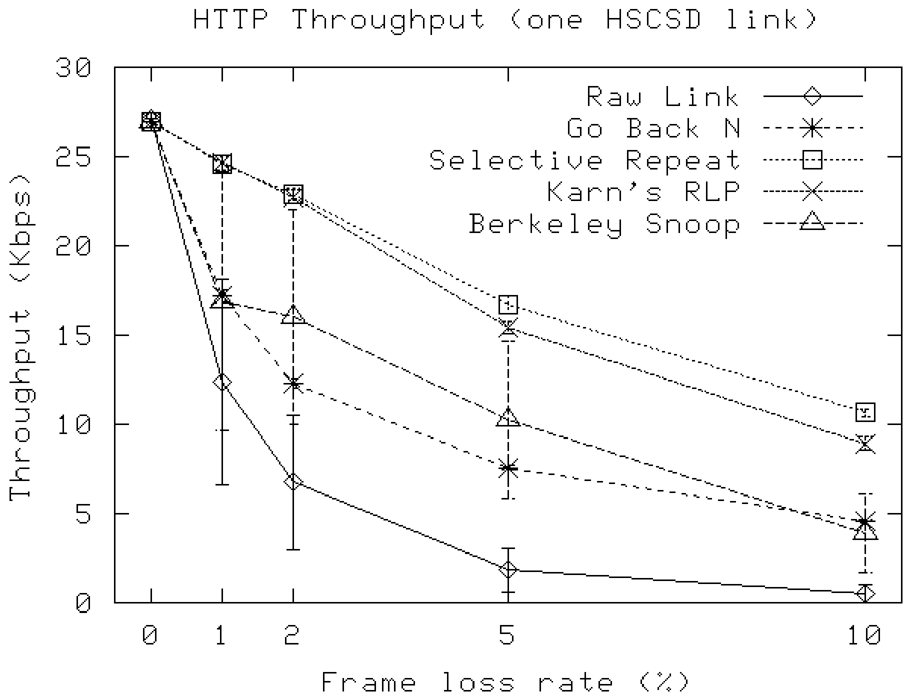
- Go Back N works terribly at any error rate
- Overhead matters for low bandwidth links
- Persistence helps at high error rates

File Transfer



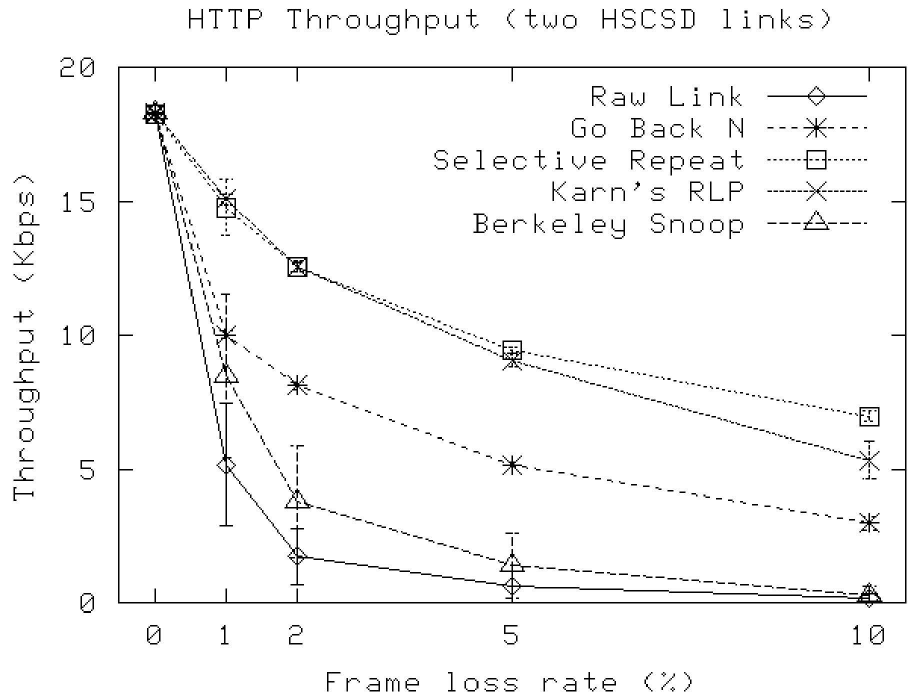
- TCP unaware schemes perform the same
- Berkeley Snoop performs very bad
- Retransmissions are needed in both directions

WWW Browsing



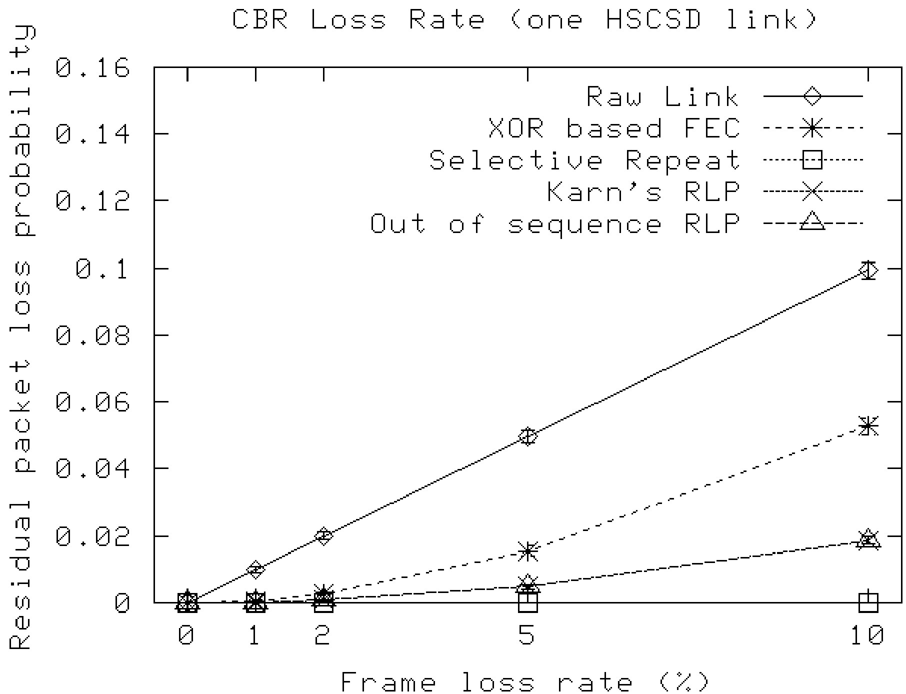
- Bi-directional traffic (requests-replies)
- Retransmissions are needed in both directions
- Berkeley Snoop has problems even in this case

WWW Browsing



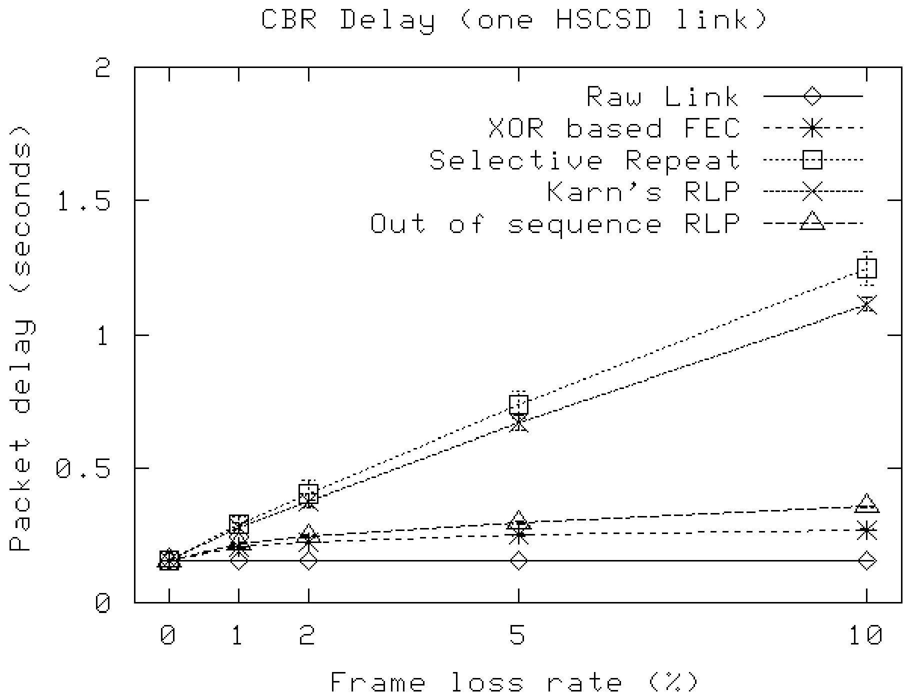
- TCP unaware schemes again perform well
- Berkeley Snoop drops below Go Back N
- File transfer cannot model interactive applications

Continuous Media



- Both RLP schemes perform identically
- XOR based FEC is too wasteful
- Selective Repeat is perfect, but do we need it?

Continuous Media



- In sequence delivery schemes are too slow
- Out of sequence RLP is close to XOR based FEC
- Both schemes do not deliver frames in sequence

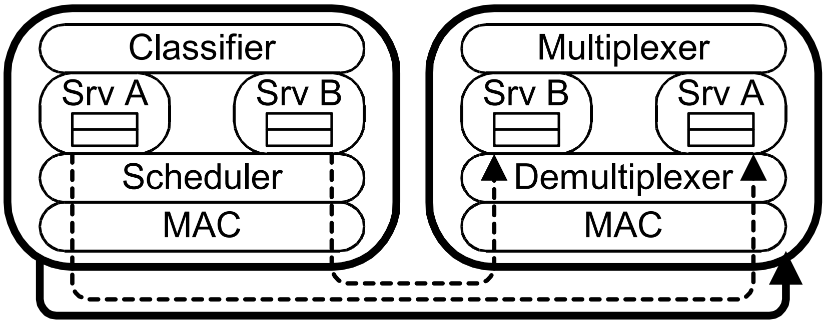
Conclusions: Single Service

- TCP enhancements
  - File transfer cannot model interactive applications
  - Both directions matter, even for downloads
  - TCP aware schemes fail for interactive applications
  - TCP unaware schemes worked for both applications
  - Excellent performance with low overhead
- UDP enhancements
  - Continuous media: low delay – medium reliability
  - Out of sequence delivery greatly reduces delay
  - Retransmissions can compete with FEC
- There is no single solution for both

Multi Service Link Layer

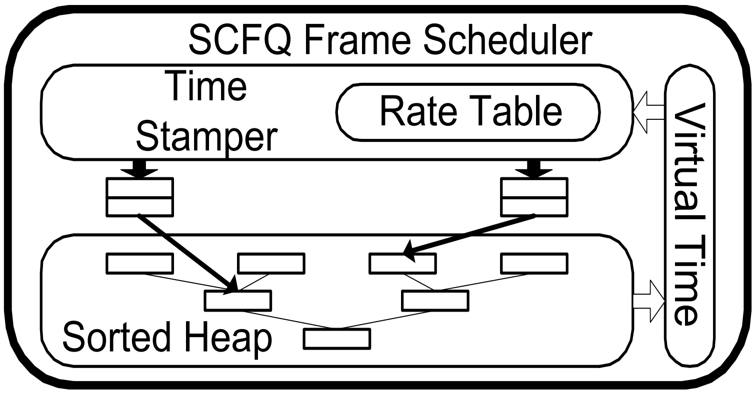
- Address the problem at its source
  - Local solution to a local problem
- Compatible with Internet architecture
  - IP and higher layers unchanged
- Aware of QoS requirements
  - Implicitly or explicitly
- Per stream/class QoS differentiation
  - Fully or mostly reliable
- Dynamic adaptation to stream/class mix
  - Variable bandwidth allocation
- Dynamic adaptation to channel conditions

MSLL Architecture



- Multiple link layer modules
- Packet classifier
  - TCP/UDP ports
  - IP ToS, DS field
- Per class load measurements
- Service class specific processing
  - Isolation between services

MSLL Scheduler

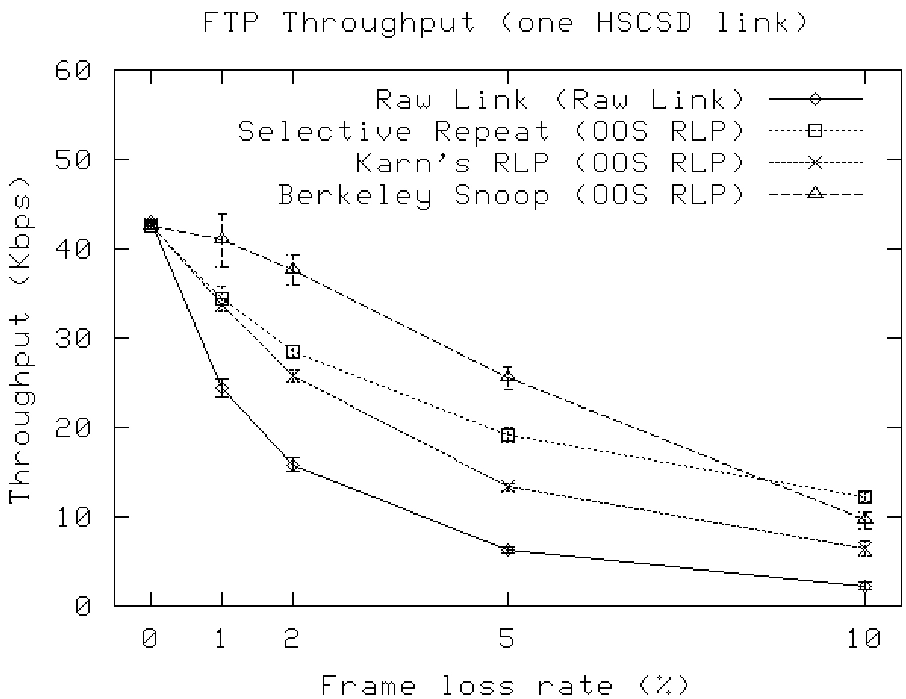


- Enforces incoming allocations
  - Protects services
  - Encourages efficiency
- Self-clocked fair queueing (SCFQ)
  - Efficient, simple, fair
  - One queue per class

Multi Service Protocols

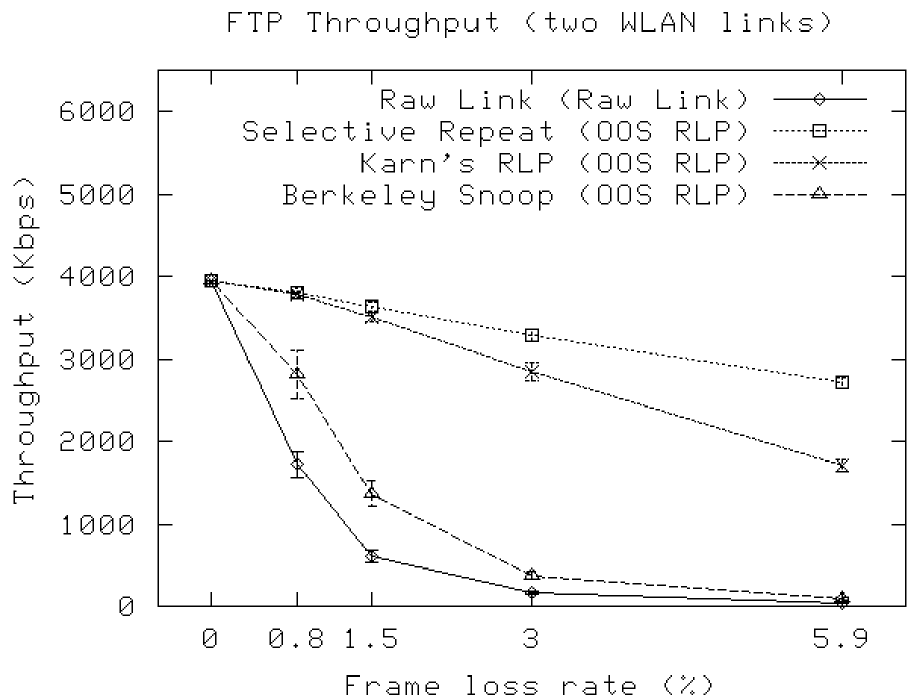
- Same protocols, two services (TCP and UDP)
- Raw Link: native link service
- TCP enhancements
  - Selective Repeat: standard selective retransmissions
  - Karn's RLP: up to 3 retransmissions per frame
  - Berkeley Snoop: TCP aware retransmissions
- UDP enhancements
  - Out of sequence RLP: variant of Karn's RLP
- TCP / UDP combinations
  - Raw link / Raw link (baseline)
  - {SR, RLP, Snoop} / OOS RLP

File Transfer



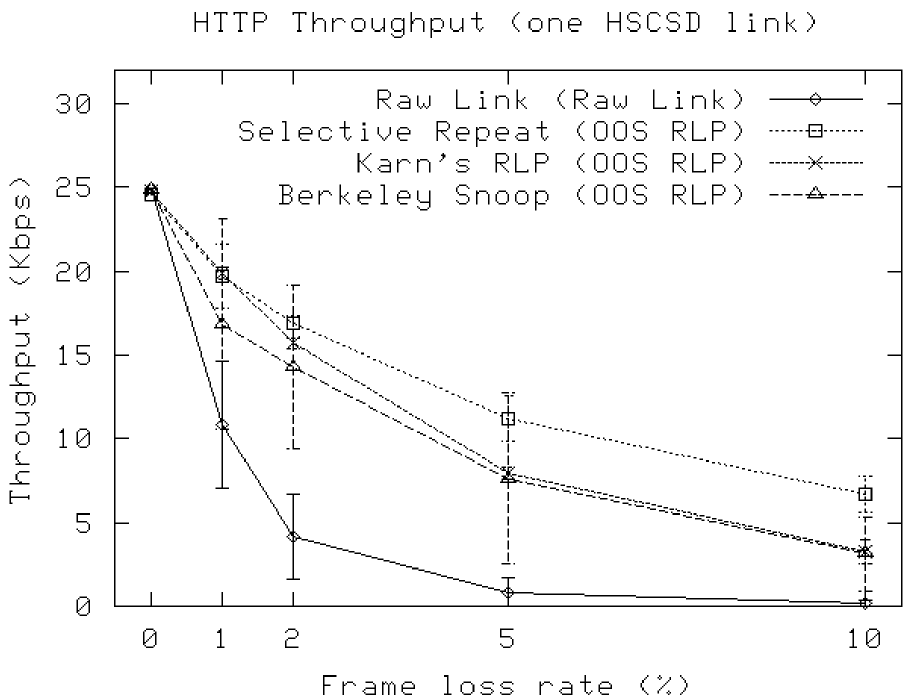
- Similar to single application tests
- Overhead matters for low bandwidth links
- Persistence helps at high error rates

File Transfer



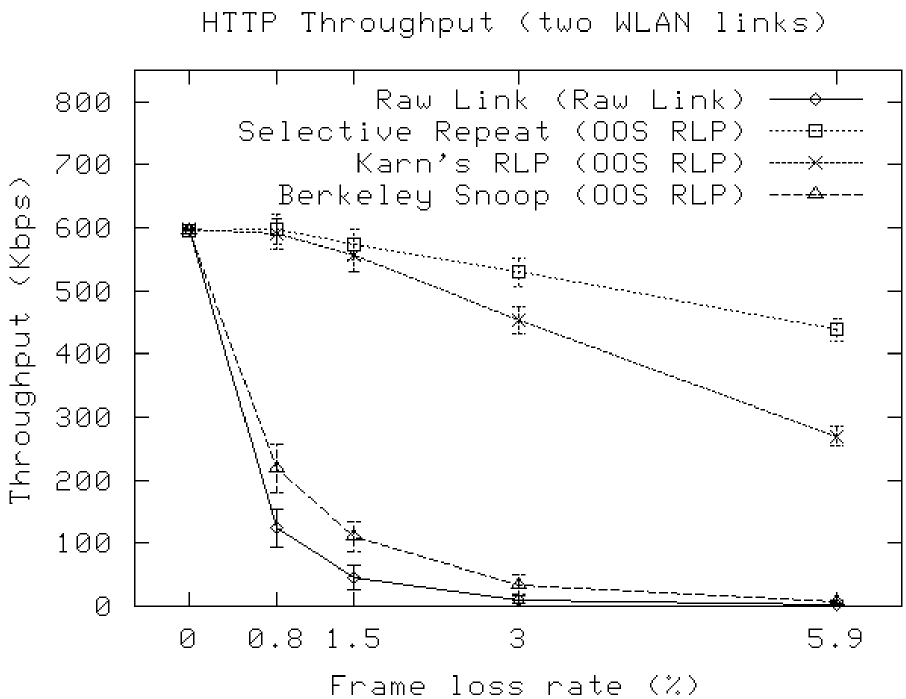
- TCP unaware schemes perform excellent
- Berkeley Snoop performs very bad
- Retransmissions are needed in both directions

WWW Browsing



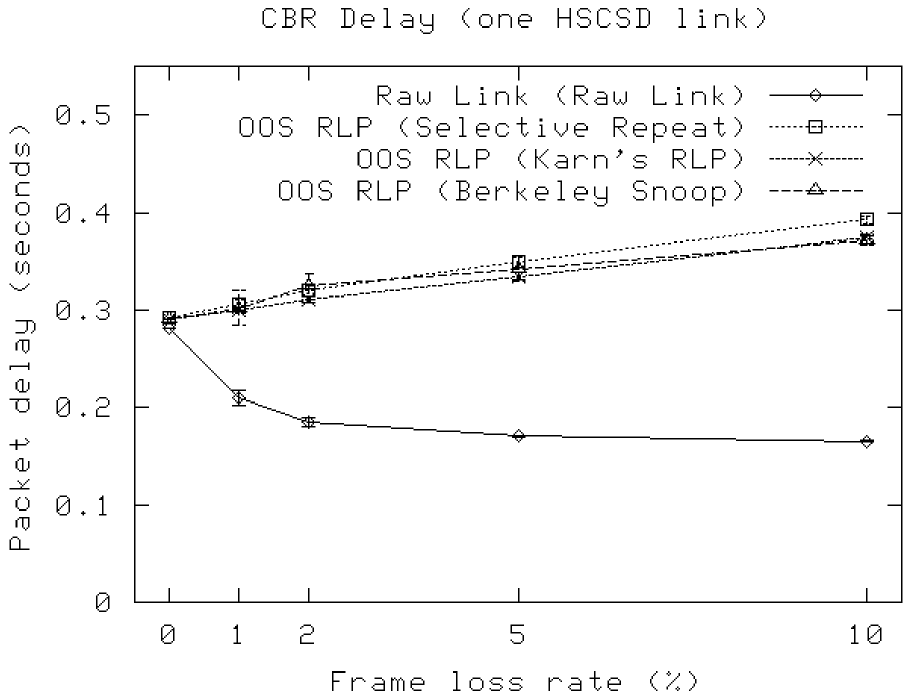
- Similar to single application tests
- Retransmissions are needed in both directions
- Berkeley Snoop has problems even in this case

WWW Browsing



- TCP unaware schemes again perform well
- Berkeley Snoop drops below Go Back N
- File transfer cannot model interactive applications

Continuous Media



- Delay is nearly the same with all TCP schemes
- Most delay is due to OOS RLP retransmissions
- The scheduler effectively protects UDP from TCP

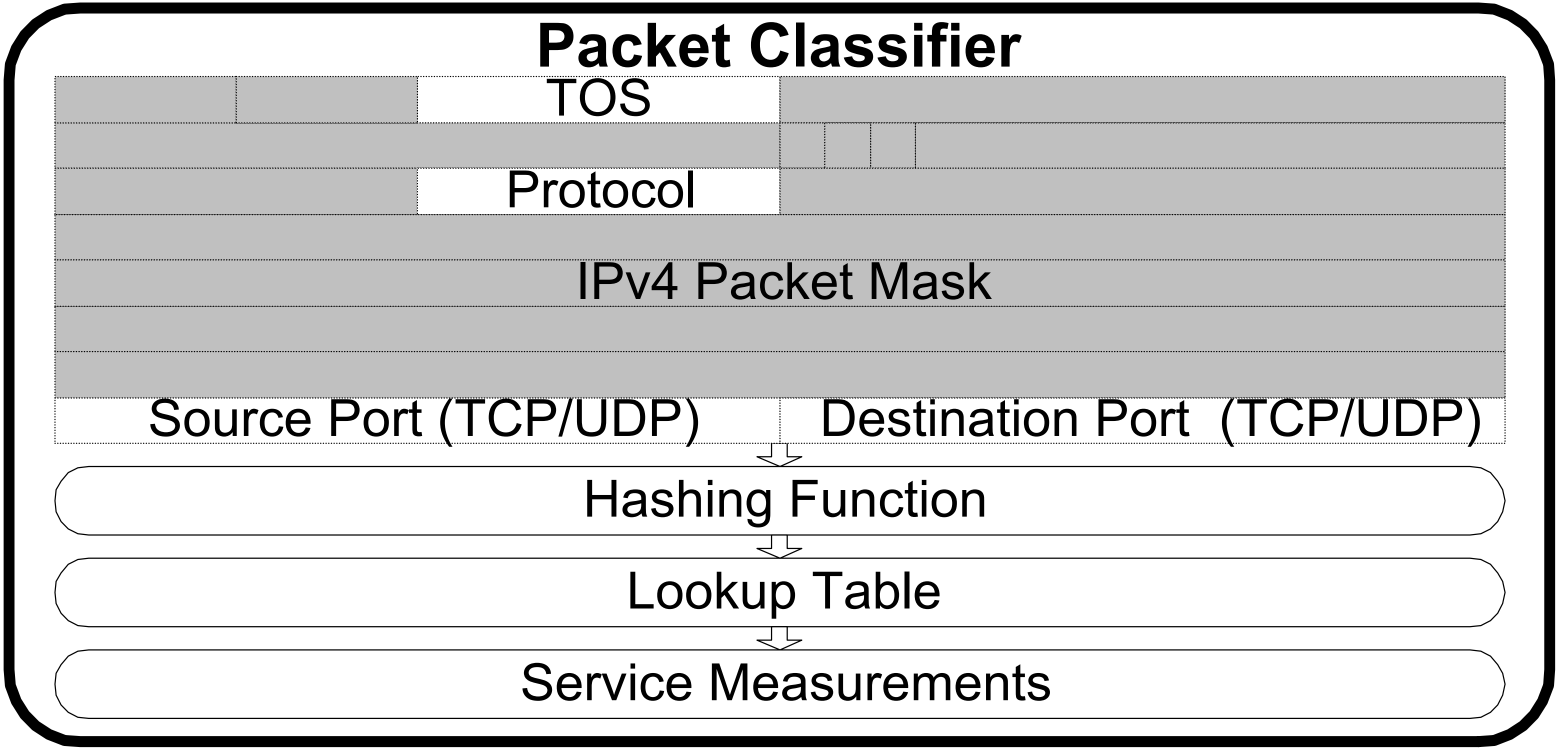
Conclusions: Multiple Services

- TCP application performance
  - The same schemes work for both TCP applications
  - One service is sufficient for all TCP applications
- UDP application performance
  - Excellent performance improvements
  - The scheduler protects UDP from TCP
- Multi-service link layer performance
  - Applications perform as in single application tests
  - Each application uses the most appropriate scheme
  - Transparent and locally customized solution
  - Supports diverse application requirements

Service Selection

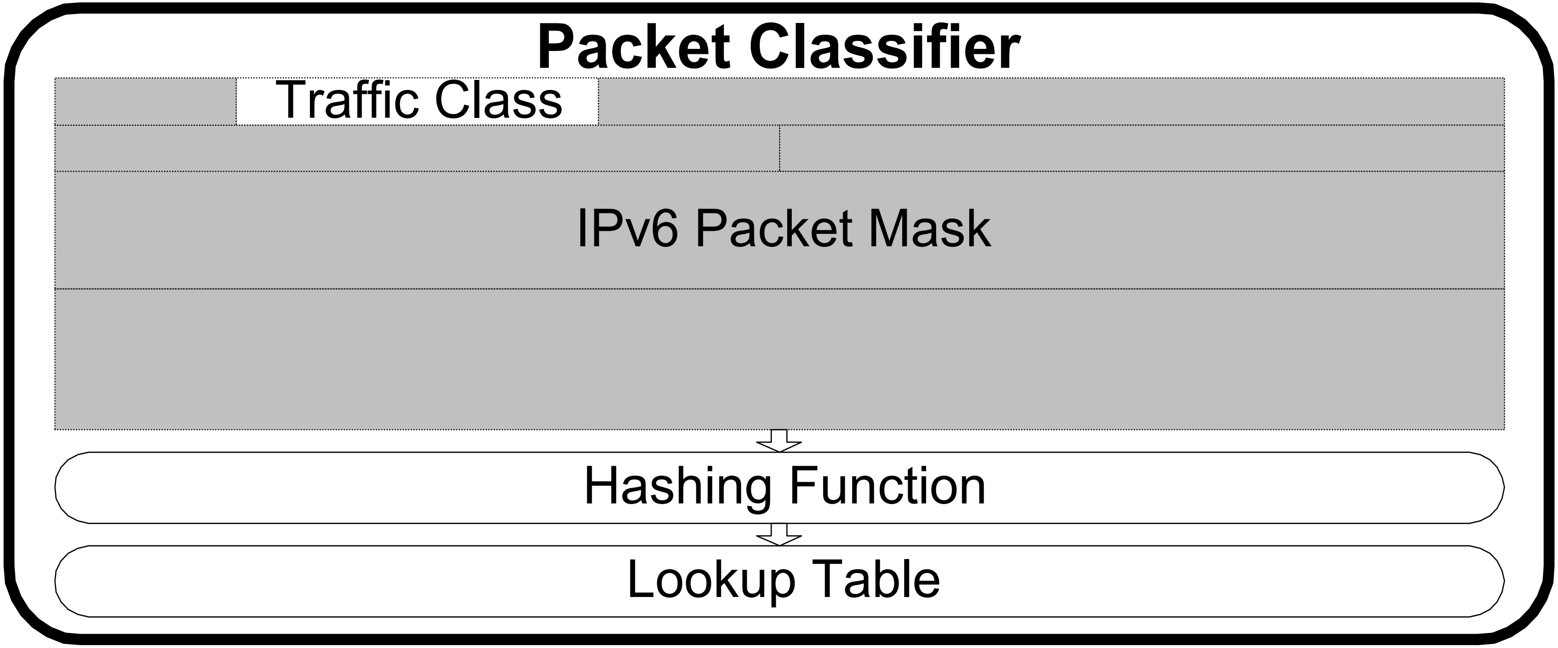
- Implicit QoS specification
  - Assigns applications to services
  - Protocol and TCP/UDP port fields
  - No changes to Internet protocols and applications
  - More immediate
- Explicit QoS specification
  - Assigns traffic classes to services
  - QoS provision
    - Integrated Services, RSVP
  - QoS differentiation
    - Differentiated Services
  - DiffServ is more flexible

Heuristic Packet Classifier



- Implicit QoS specification

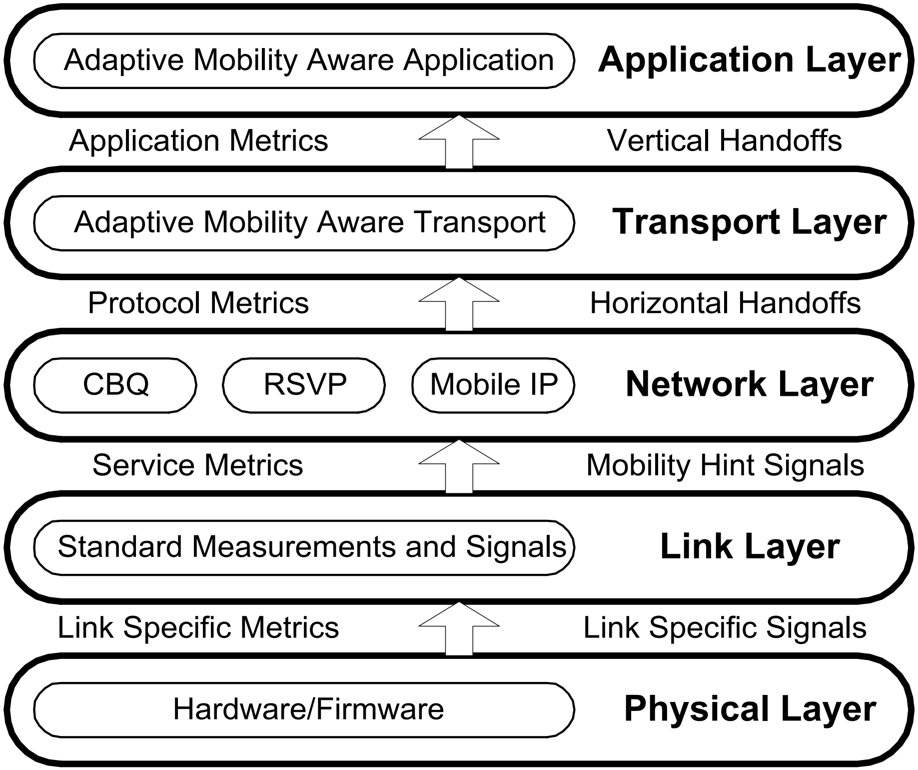
DiffServ Packet Classifier



- Explicit QoS specification
- Dynamic service selection

Measurements and Feedback

- Service selection
  - Standard metrics
  - Composition
- Notifications
  - Mobility hints
  - Adaptive schemes



Conclusions: Summary

- TCP performance severely impacted
- TCP is not the only concern
  - Real-time multimedia over UDP
- Link layer enhancements
  - Fast local recovery
  - Customized to underlying link
- Wireless links: natural choice for QoS support
- Differentiated services because
  - Bandwidth is scarce and precious
  - Link performance is variable and unpredictable