Performance of Transport Protocols over Wireless Links

Pathapati Kamakshi Sirisha
Mrudula Putcha
Wang Rui

Department of Electrical Engineering & Computer Science
EECS 822 Project Presentation

{kamipks, mrudula, wangrui}@ittc.ku.edu
Transport Protocols over Wireless Links

Outline

• Introduction
• Motivation
• Nearly- Reliable Mode
• Quasi- Reliable Mode
• Summary
• Simulation Results
• Conclusion
Transport Protocols over Wireless Links

Introduction

• Introduction
• Motivation
• Nearly- Reliable Mode
• Quasi- Reliable Mode
• Summary
• Simulation Results
• Conclusion
Transport Protocols over Wireless Links

Motivation

- Introduction
- Motivation
- Nearly- Reliable Mode
- Quasi- Reliable Mode
- Summary
- Simulation Results
- Conclusion
Transport Protocols over Wireless Links

Nearly- Reliable Mode

- Introduction
- Motivation
- Nearly- Reliable Mode
- Quasi- Reliable Mode
- Summary
- Simulation Results
- Conclusion
Nearly- Reliable Mode

Introduction

- Highly reliable but does not guarantee delivery
- Split ARQs used to give high level of reliability
- Gateway immediately returns ACK to the source
- The gateway buffers segments after ACKing them
- More bandwidth efficient than traditional TCP
Nearly- Reliable Mode

Figure 3 [RS2009]
Nearly- Reliable Mode
Performance measurement

- Simulations compare E2E and HBH ARQ
- Nearly- Reliable mode uses HBH ARQ
- Trade off reliability against bandwidth efficiency
Nearly- Reliable Mode
Software Architecture

- P2P channel script modified to make channel noisy
- This scenario used to simulate wireless channel
- Various nodes connected with P2P links
- This used to simulate multi-hop scenario
- ARQ mechanism enabled on each node
- E2E ARQ achieved by not using split ARQs
Transport Protocols over Wireless Links
Quasi- Reliable Mode

- Introduction
- Motivation
- Nearly- Reliable Mode
- Quasi- Reliable Mode
- Summary
- Simulation Results
- Conclusion
Quasi- Reliable Mode

Introduction

- Does not use ACKs and ARQs
- Provides open loop error recovery
- Uses mechanisms like FEC
- Provides no guarantee of delivery, only statistical reliability
- Tradeoff
  - bandwidth overhead against overhead of retransmissions
Quasi- Reliable Mode
Quasi- Reliable Mode
Performance measurement

- Simulations compare FEC and non-FEC modes
- Reed-Solomon FEC used
- Trade off
  - bandwidth overhead due to redundant bits
  - against time overhead due to retransmissions
- Efficiency of FEC in correcting errors
  - bit errors
  - burst errors
Quasi- Reliable Mode
Reed-Solomon Codes

- RS- FEC code emulated
- Represented as RS \((n, k)\) with s-bit symbols
  - \(k\) code words of s-bit symbols encoded into n code words
- Most popular used is
  - RS \((255, 223)\) with 8 bit symbols
- Theoretical limit on number of errors
  - \(t = (n-k)/2\)
Quasi- Reliable Mode
Software Architecture

- RS- FEC code emulated
- Errors introduced using RateErrorModel (ns-3)
- RateErrorModel modified to suit emulation
- Packets marked as corrupted and sent to Layer4(L4)
- FEC emulated at L4 decides on reconstruction
- FEC emulated by modifying the UDP header
- Efficiency of FEC calculated based on this
Acknowledgements

- Justin P. Rohrer, ResiliNets Research Group
- Asifuddin Mohammad, ResiliNets Research Group
- Egemen Cetinkaya, ResiliNets Research Group
- Professor James Sterbenz
References
