Managing Delay and Jitter in Mesh Networks through Path-Aware Distributed Transmission Scheduling

Justin Yackoski, Chien-Chung Shen (yackoski,cshen@cis.udel.edu)
DEGAS Networking Group, University of Delaware

Motivation

- Low end-to-end delay and jitter is required by real-time, streaming applications like VoIP and IPTV
- Wireless networks have significant per-hop and per-packet delays due to MAC layer
- Performance of multi-hop flows easily degrades below tolerable levels

Contributions

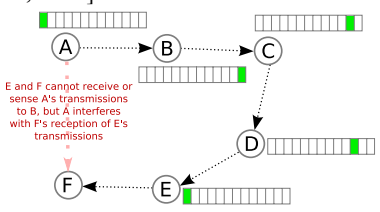
- Method for minimizing per-hop delay and eliminating per-packet delay by exploiting periodic nature of data
- Distributed method of resolving interference and selecting locally optimal transmission times
- Full-path Long-life Adaptively Synchronized Hop Reservation (FLASHR) protocol

Issues with CSMA

- Contention process results in per-packet costs at each hop in a path
- Costs incurred forever, even if traffic has repeating pattern
- Some packets experience low delay, all experience high jitter

Issues with TDMA

- Tight clock synchronization needed
- No jitter, but per-hop delay can be high
- Distributed, load-based slot assignment scheme exists [Wolf et al., 2006] but scheduling is not done with regard to end-to-end delay
- Determining contention relationship is difficult and can change due to combined interference
- Slot assignment with optimal delay is NP-Hard [Ramanathan, 1999]

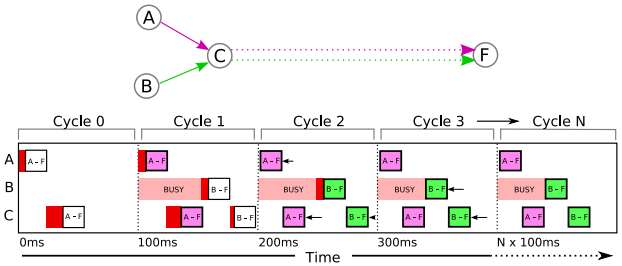


Worst case delay for assignment scheme which does not consider delay: (one cycle - one slot) x # of hops

FLASHR Protocol Overview

- Split time into cycles – size agreed upon in advance
- Slots can have any length and start at any offset
- Nodes monitor channel, mark portions of cycle in use based on local clock – **no tight synchronization needed**
- Nodes contend for new slots in unused portions of the cycle as needed
- Successfully used slots are *re-used* in next cycle *without* contention
- Nodes own slots for transmission of data in a particular flow
- Each node searches for locally optimal schedule by attempting earlier slots – minimizing delay at the node
- Once end-to-end delay for a flow is below T_{qos}, nodes in flow stop attempting new slots

Example

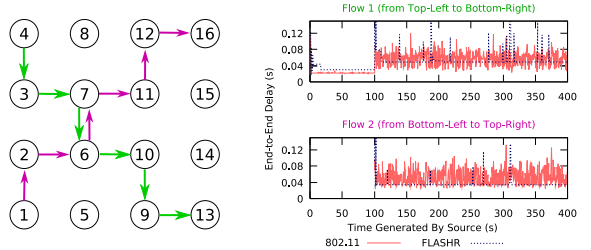


- At the start of each cycle, nodes A and B have 1 packet to send to F (we omit intermediate nodes past C for simplicity)
- Cycle 0 – A contends with B, successfully uses new slot for flow A→F, as does C. B observes part of cycle is busy
- Cycle 1 – slots for A→F are re-used without contention (B assumes channel is in use). B then successfully uses new slot for flow B→F, as does C
- Cycle 2 – A and C successfully attempt to use slots for A→F at slightly earlier times to remove initial backoff delay. B observes this and updates portion of cycle which is busy
- Cycle 3 – slots for A→F are re-used without contention. Slots for B→F are successfully used at slightly earlier times.
- The schedule is then repeated unchanged without any unnecessary delay or contention until network conditions change

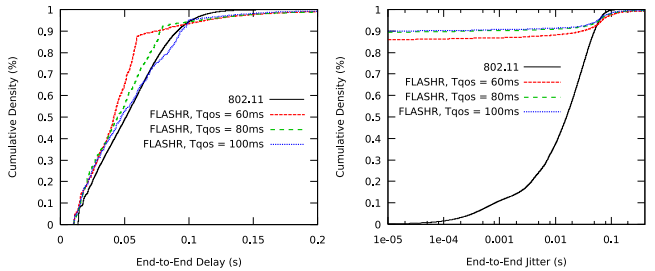
Simulation Results

General Setup: QualNet 3.7 – 4x4 grid with 225m spacing, 250m transmission range, 2Mbps data rate. Each CBR flow creates one 512-byte packet every 500ms. Both FLASHR and 802.11 use the same routes for fair delay comparisons.

Representative sample of individual packet delays: two 6-hop flows, T_{qos} = 50ms, flow 1 begins at 0s, flow 2 begins at 100s, sources generate packets approximately 1.5ms apart



Distribution of delay and jitter: 50 trials with four flows between randomly chosen nodes 4-6 hops apart, flows start at times randomly distributed over 40ms and run for 600s



Conclusions and Future Work

- FLASHR re-uses good schedules found via CSMA
- FLASHR has higher jitter during schedule converging and packet loss but <10μs jitter for 90% of packets
- Tradeoff between delay and jitter – finding better schedule requires more time/effort
- Working on better heuristics for selecting candidate slots to reduce both convergence time and delay/jitter tradeoff

Thanks

- NSF CNS-0335302 and CNS-0347460