



# Managing End-to-End Delay for VoIP Calls in Multi-Hop Wireless Mesh Networks

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
- Multi-hop VoIP calls are not well supported

## Contributions

- Methods for minimizing per-hop and per-packet delays
- Method for scheduling transmissions based on end-to-end delay QoS requirements
- Full-path Long-life Adaptively Synchronized Hop Reservation (FLASHR) protocol

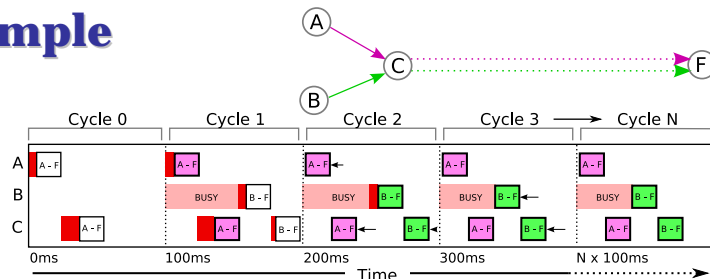
## Issues with Existing Schemes

- CSMA:**
  - Contention process results in per-packet, per-hop costs, even if traffic has cyclical pattern
  - Delay unpredictable
- TDMA:**
  - Tight clock synchronization needed
  - Scheduling is done in each local neighborhood without regard for end-to-end delay, scheduling with optimal delay is NP-hard [Ramanathan, 1999]
- Hybrid:**
  - [Singh et. al., 2007], [Carlson et. al., 2006]
  - Per-hop delays reduced, but not fully
  - End-to-end delay is # hops in route multiplied by per-hop delay, and is **not actively or intentionally managed**

## 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, are “owned” after successful use
- Owned slots are *re-used* in next cycle *without* contention
- Destination application notifies FLASHR when end-to-end delay for a flow meets desired QoS, propagated backward along route
- Each node searches for earlier slots for flow until notified that QoS is met
- Implicit ACKs used when possible, explicit ACKs scheduled independently from DATA
- Per-flow state and small buffer at intermediate nodes

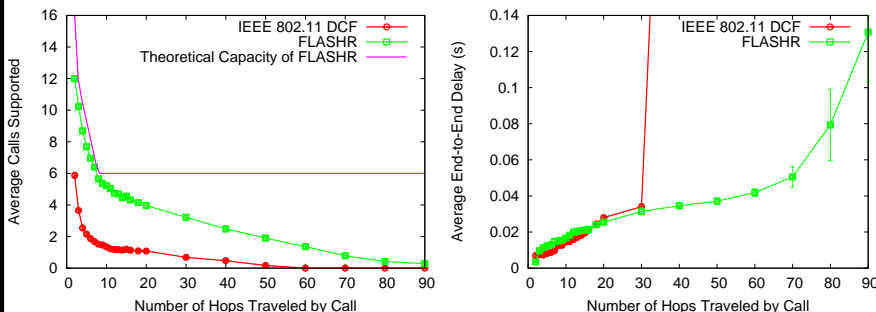
## Example



- At the start of each cycle, nodes A and B have 1 packet to send to F
- 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 and C successfully use new slots for flow B→F
- 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
- In subsequent cycles, the schedule is further optimized for each flow until the end-to-end delay QoS constraint is met.

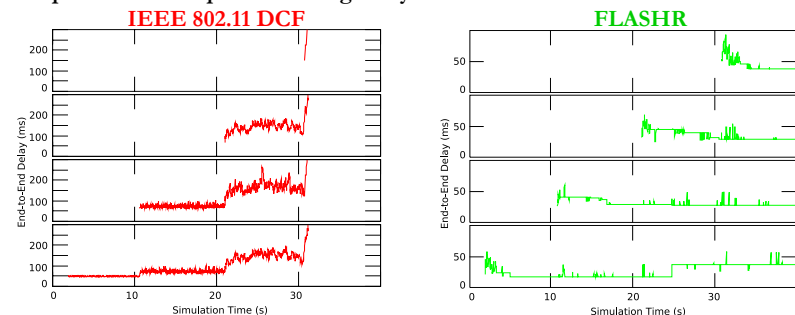
## Simulation Results

**General Setup:** QualNet 3.7 – Nodes 150m apart, 200m xmit range, 450m CS range. Use 38-byte iLBC codec (used in Skype). Call supported if <8% loss in last 1s, no consecutive loss, <50ms delay. Line topology of varying length, calls placed only between endpoints.



- FLASHR achieves near-optimal capacity at less than 10 hops
- IEEE 802.11 DCF and FLASHR have comparable delay below 30 hops, but FLASHR supports 2-4x as many calls
- Above 30 hops, FLASHR more aggressively reschedules transmissions to meet QoS, allowing support for up to 70 hop calls.

Sample run of 50 hop line showing delays for both directions of each of two calls



Thanks

NSF CNS-0335302 and CNS-0347460