Time Division Hashing

A New Scheduling Scheme for Wireless Ad-Hoc Networks

Winnie Cheng, I-Ting Angelina Lee, Neha Singh

MIT
Computer Science and Artifiical Intelligence Laboratory
March 2, 2005

Outline

- Overview
- Proposed Algorithm
- Implementation
- Simulation Results
- Conclusions and Future Work

Outline

- Overview
- Proposed Algorithm
- Implementation
- Simulation Results
- Conclusions and Future Work

Overview

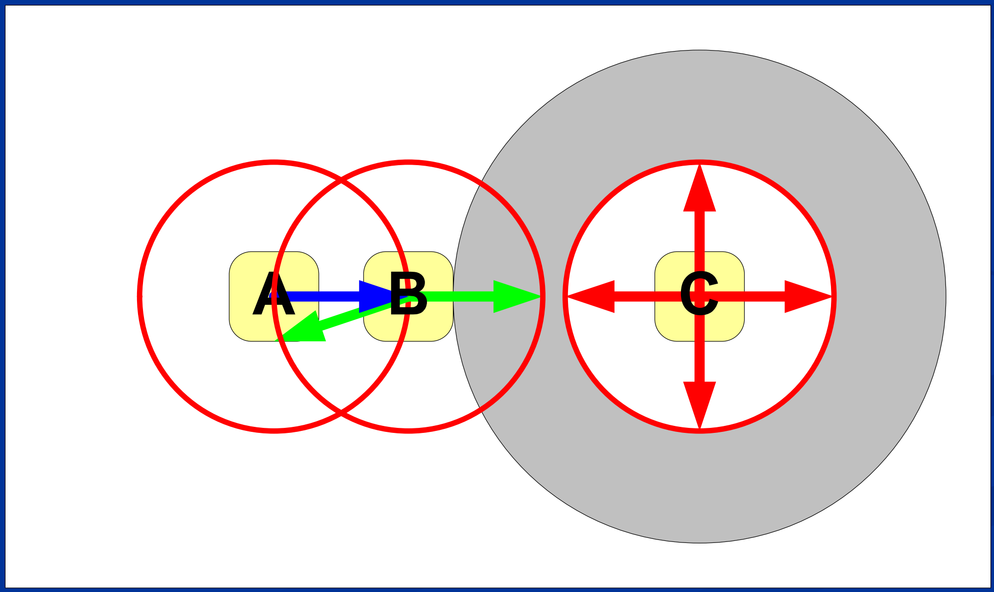
- Studied performance issues of current 802.11
- Proposed a new scheduling algorithm
  - TDH: Time Division Hashing
- Implemented TDH in NS-2
- Evaluated TDH in a variety of topologies

Background

- 802.11 – protocol for wireless communication
- RTS/CTS – Request To Send/Clear To Send
  - Handshake mechanism in 802.11 to gain access to the channel
  - Used to minimize packet collisions between neighboring nodes
- Exponential backoff used when collision is detected

Problems with Current Scheduling Scheme

- Low bit rate of RTS/CTS [spec802.11]
  - ~5-10 times lower than the data rate
- Ineffectiveness of detecting hidden nodes in interference range [KXu03]

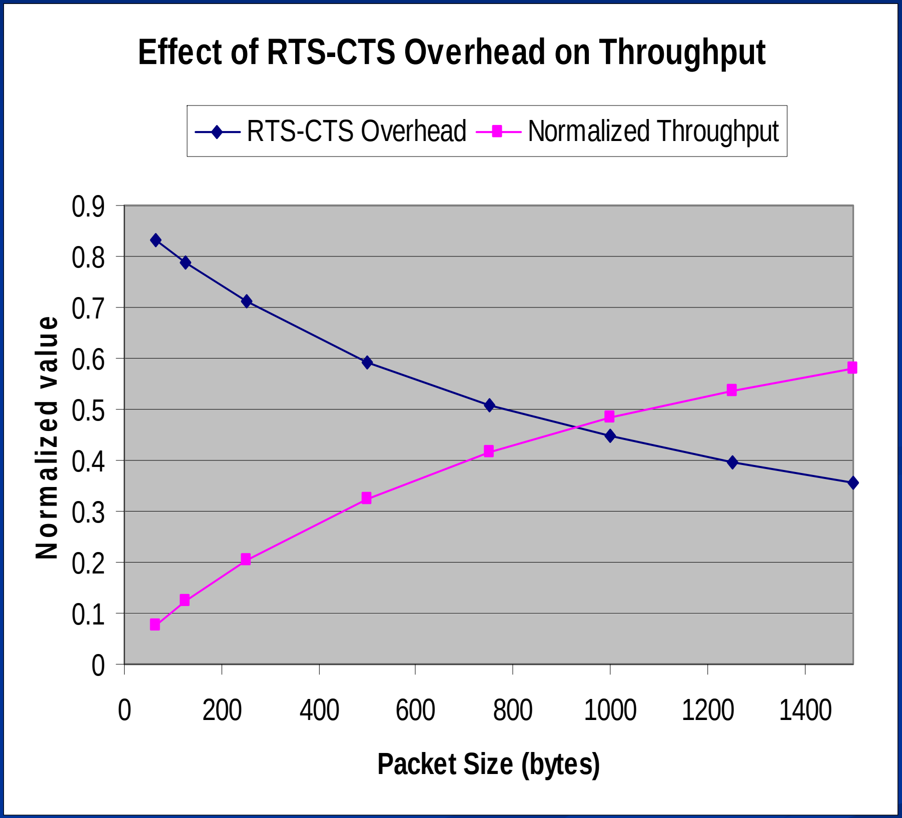


Problems with Current Scheduling Scheme

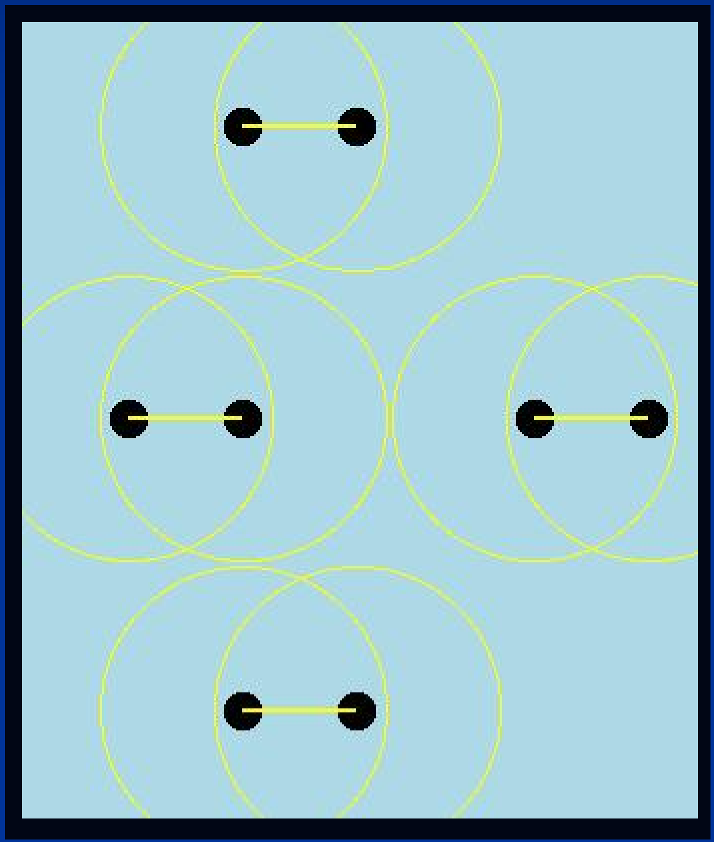
- Performance degradation due to exposed terminal problems [SXu02]
- Aggressive (Exponential) backoff causes long idle time [Li01]
- Unfairness with transmitter-based contention [Bharghavan94]

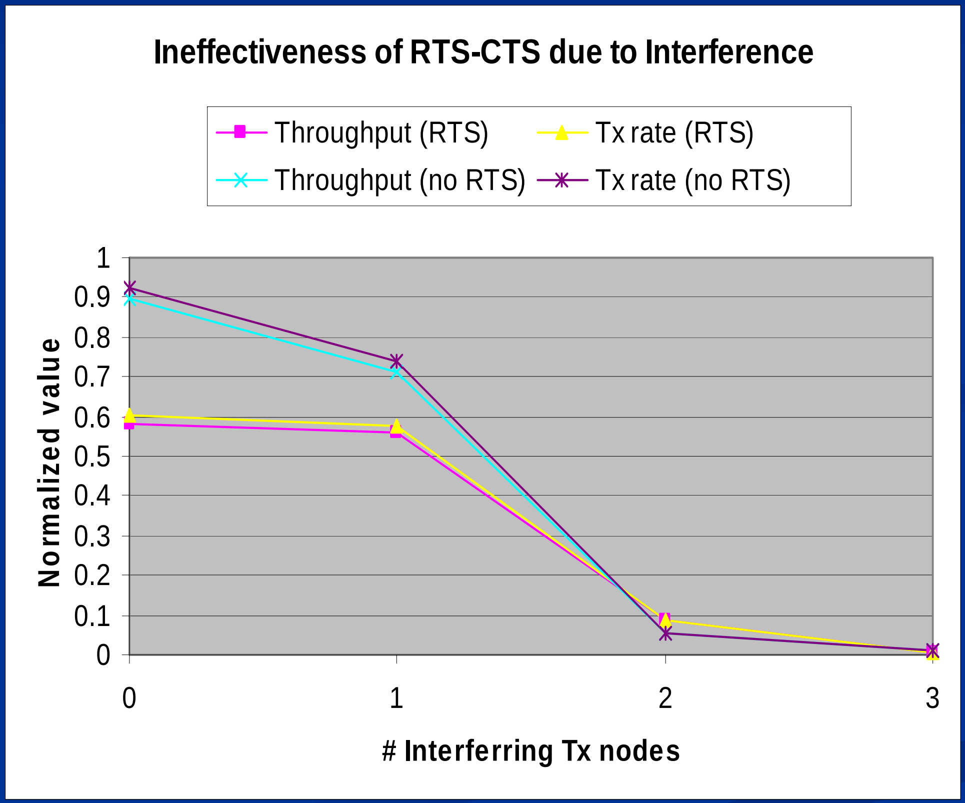
RTS/CTS Overhead

- 11Mbps Channel Bit Rate, 1Mbps RTS/CTS Transmission Rate
- Due to RTS/CTS overhead, throughput is at max 58% for 1500-byte packets



Interference Problem





Outline

- Overview
- Proposed Algorithm
- Implementation
- Simulation Results
- Conclusions and Future Work

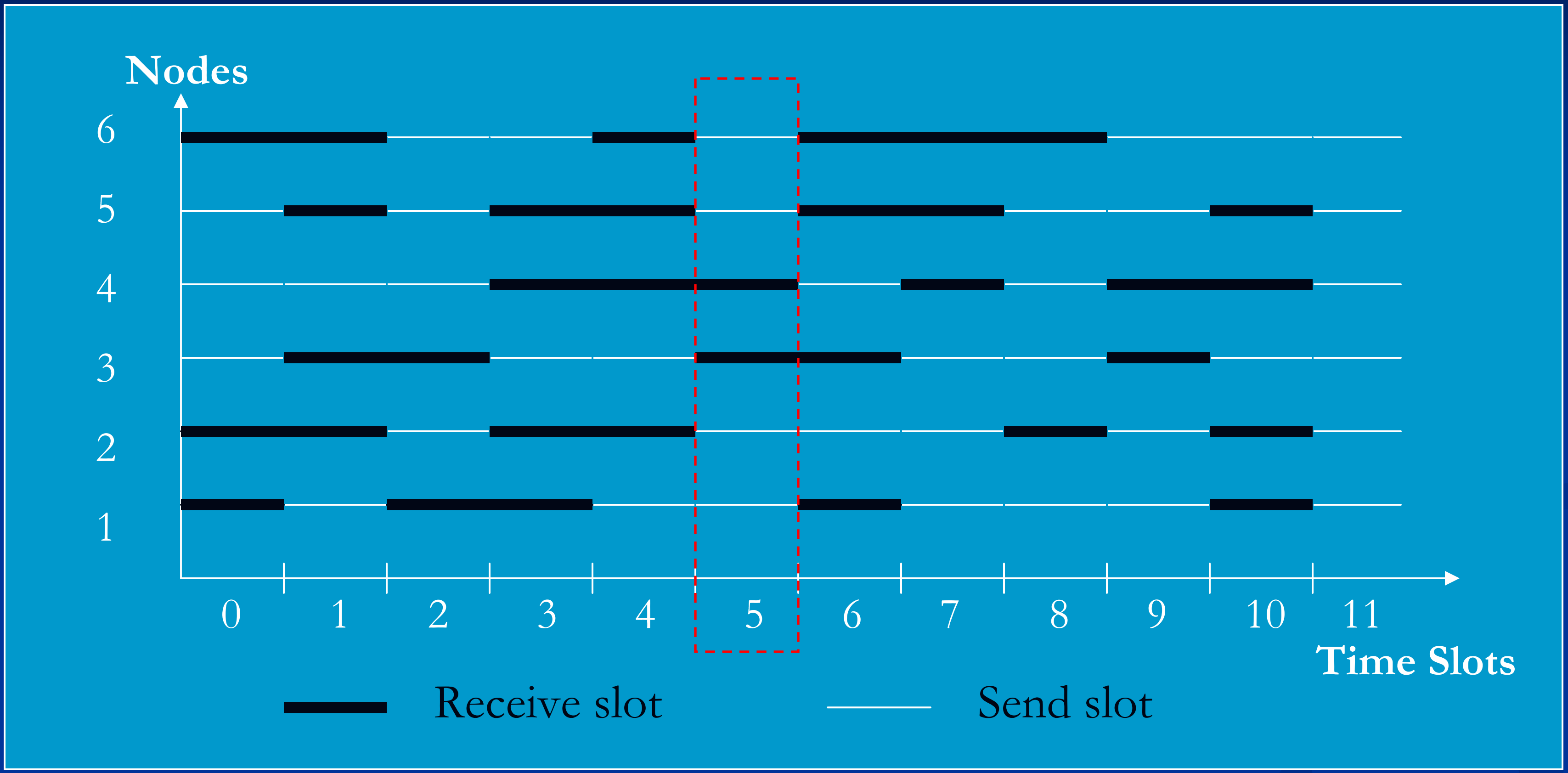
Proposed Scheme: TDH

- Based on Tim Shepard's clock exchange scheduling algorithm for Packet Radio Networks [Shepard95]
- Fixed-sized time slot and deterministic sending/receiving schedule
- Eliminate the overhead from RTS/CTS and long channel idle time due to aggressive backoffs

Time Division Hashing

- Time is *divided* into fixed-sized slots
- Each node has a unique seed value that is known to all other nodes
- Status (sending/receiving) of a node for a specific time slot is decided deterministically by hashing
- Parameters of the algorithm
  - **p** : probability that the node is in 'send' mode
  - **n** : number of slots per second
  - **s** : random seed value of a node

Deterministic Send/Receive Schedule



Node 2 can send a package to 3 in time slot 5 if they are neighboring.

TDH Algorithm

- Initialize seed **s** to random value
- **PrevTimes**: table containing the time for which the last packet was scheduled for each neighbor
- **ScheduledTimes**: list of all the time slots for which packets have already been scheduled
- **SchedulePacket** (recvID, currrentTime)
  - **t** = beginning of next time slot from currentTime
  - if (PrevTimes(recvID) > **t**)

    then t = PrevTimes(recvID) + slotLength
  - s_{recv} = seed of receiver

How TDH Works

- loop
  - increment **t** till a slot is found that is not in ScheduledTimes
  - senderMode = hash (**s** + **t**)
  - destMode = hash ($\mathbf{s_{recv}}$ + **t**)
  - if (senderMode <= **p** and destMode > **p**)
    - Schedule packet for **t**
    - Update ScheduledTimes and PrevTimes
    - return **t**

  else
    - **t** = **t** + slotLength

Features of TDH

- Unlike [Shepard95], TDH requires no explicit seed exchange between neighbors (the unique MAC address ID can be used as the random seed)
- The parameter **p** (send probability) can be tuned to achieve the best performance for a particular topology
- The packet and the ACK are exchanged in the same time slot
- The size of a time slot is dependent on the packet size

  slotLength = time to transmit packet + time to transmit ACK
  + 2 * (time to switch radio)

- If no packet is scheduled to be sent during a send time slot, receiver is automatically turned on to listen to the traffic

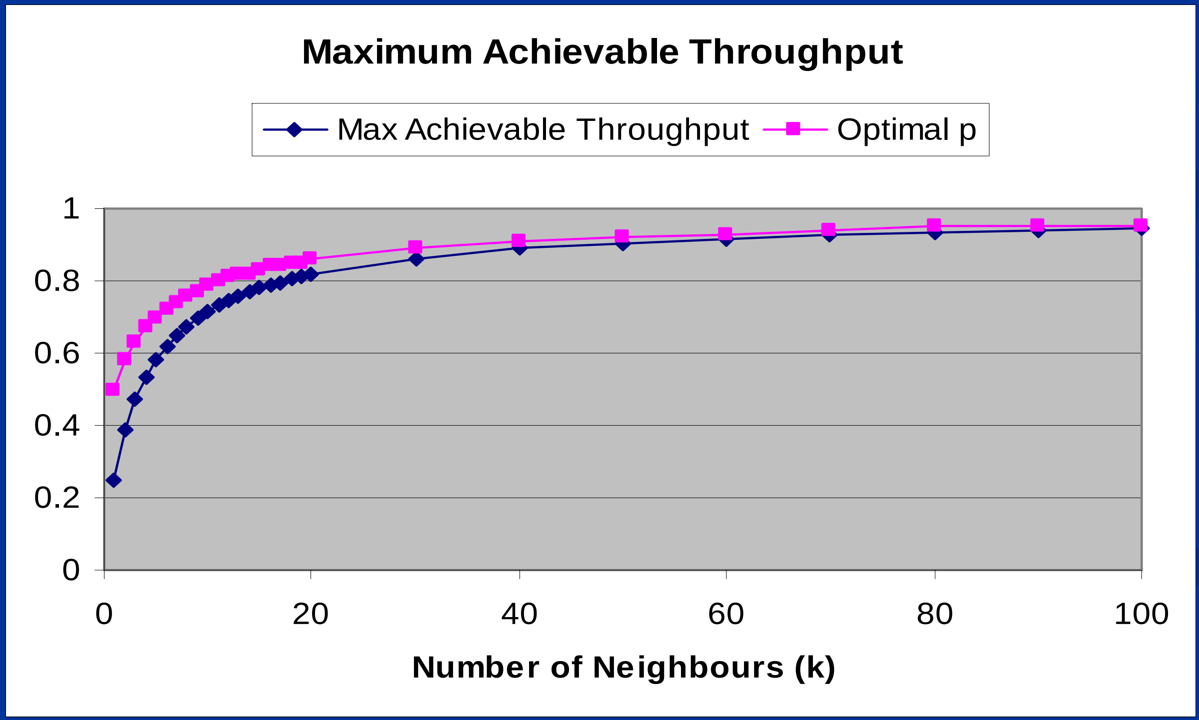
Theoretical Throughput

p = Probability of a node sending, **k** = Number of neighbor

Max Throughput

= (Prob of the sender in Tx) * (Prob of at least one neighbor in Rx)

$= p\ (1- p^k) = p - p^{k+1}$



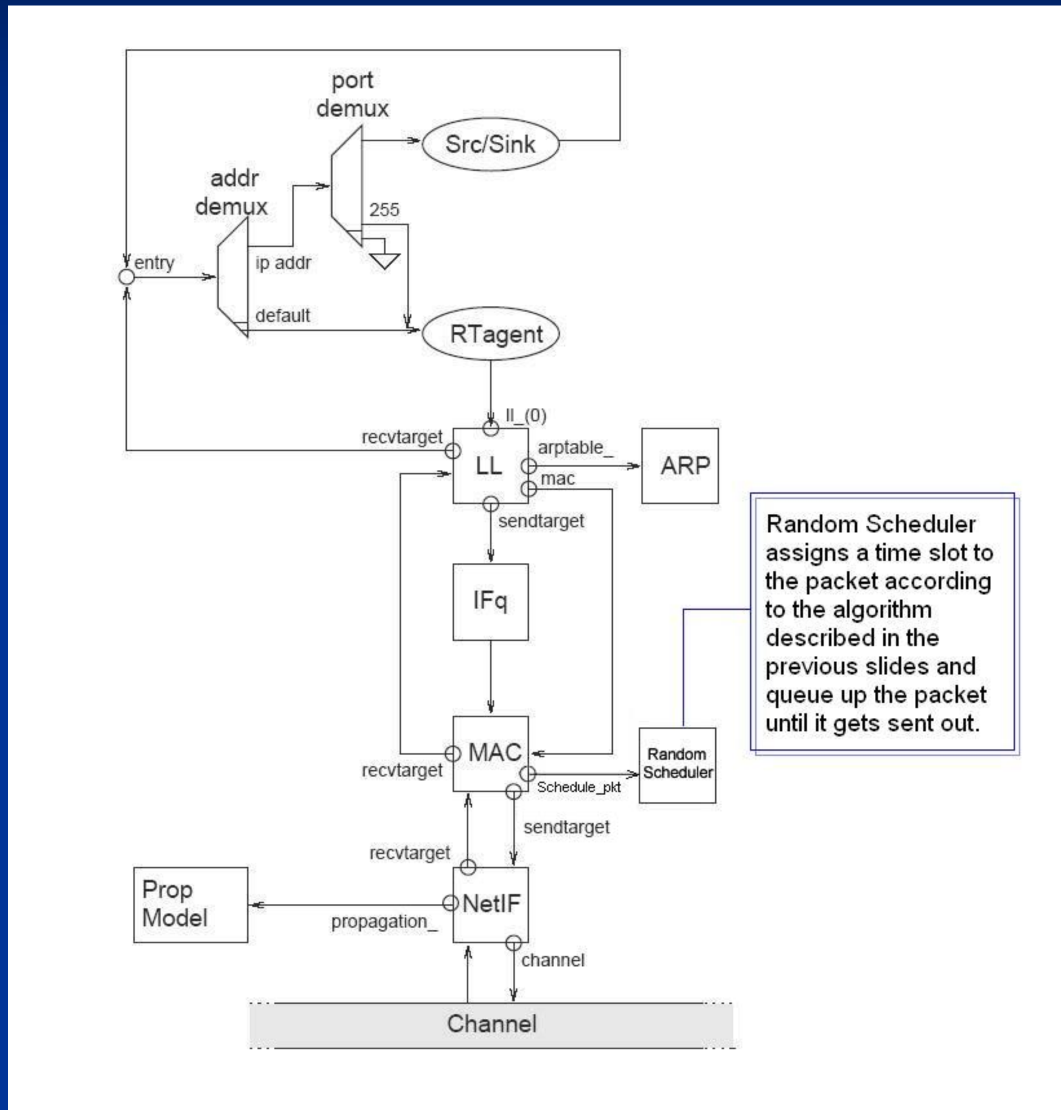
Outline

- Overview
- Proposed Algorithm
- Implementation
- Simulation Results
- Conclusions and Future Work

Implementation Strategy

- **Code Base:** The CMU Monarch Project's Wireless and Mobility Extensions to *ns*.
- **Our Extension:** Complete rewrite of MAC layer protocol
  - Remove exchange of RTS / CTS
  - Eliminate backoffs and use of Network Allocation Vector
  - Implement and employ TDH scheduling algorithm

Schematics of a Mobile Node



Issues Encountered

- Support for multiple outstanding packets in the MAC layer
- Management of state for the scheduled packets
- Flow control with the link layer
- Prevention of packet reordering

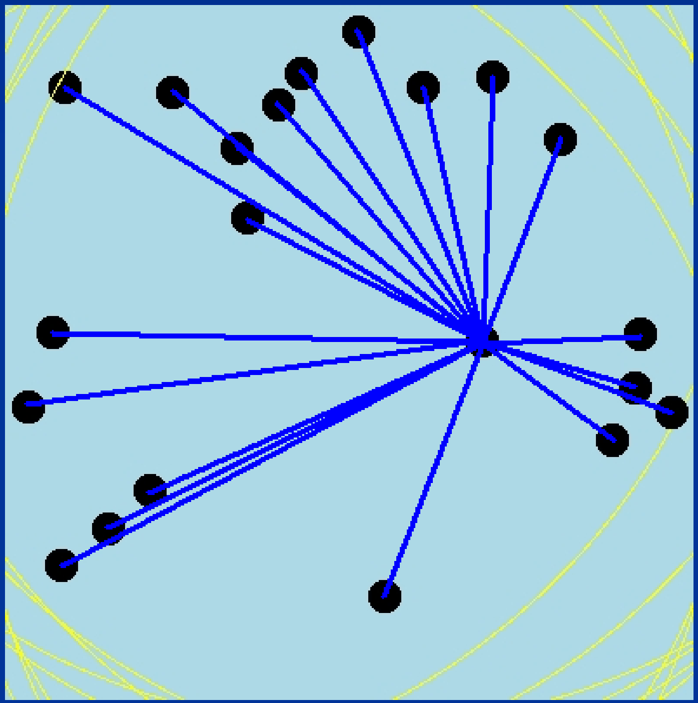
Outline

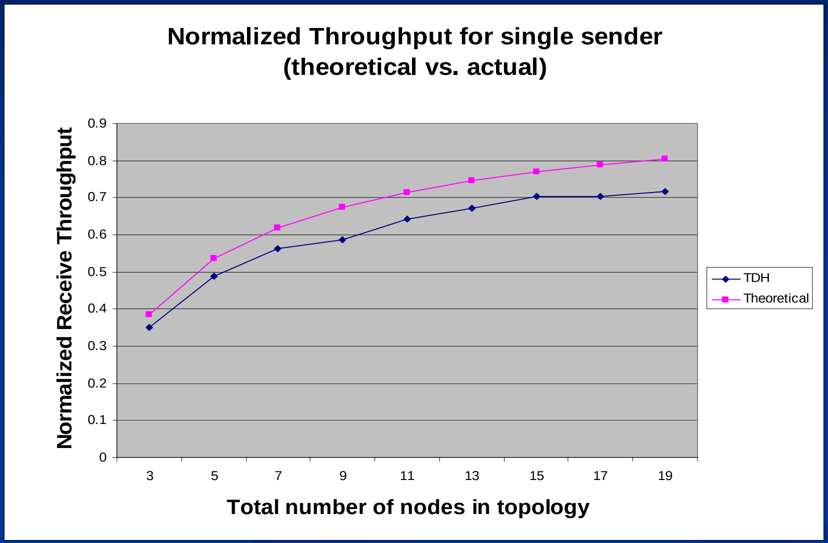
- Overview
- Proposed Algorithm
- Implementation
- Simulation Results
- Conclusions and Future Work

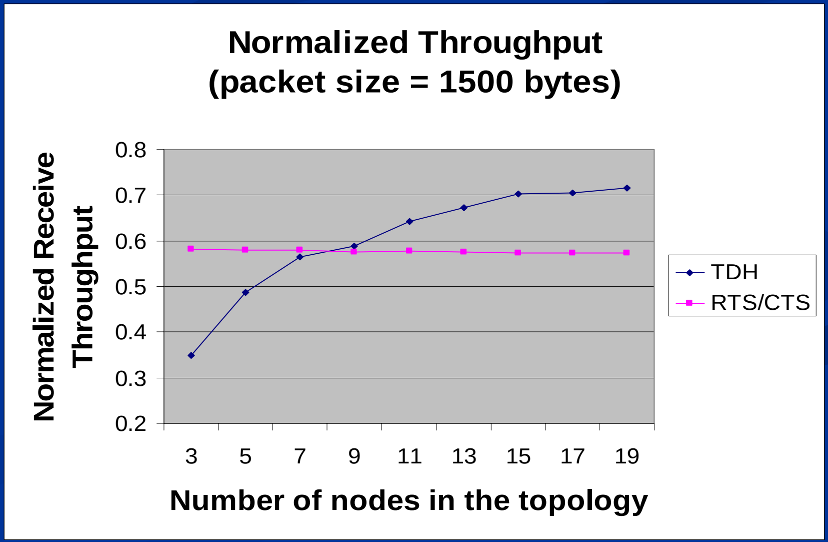
Simulation Results

- **Topologies studied:**
  - Single sender
  - Access Point
  - Clique
  - Chain
  - Random

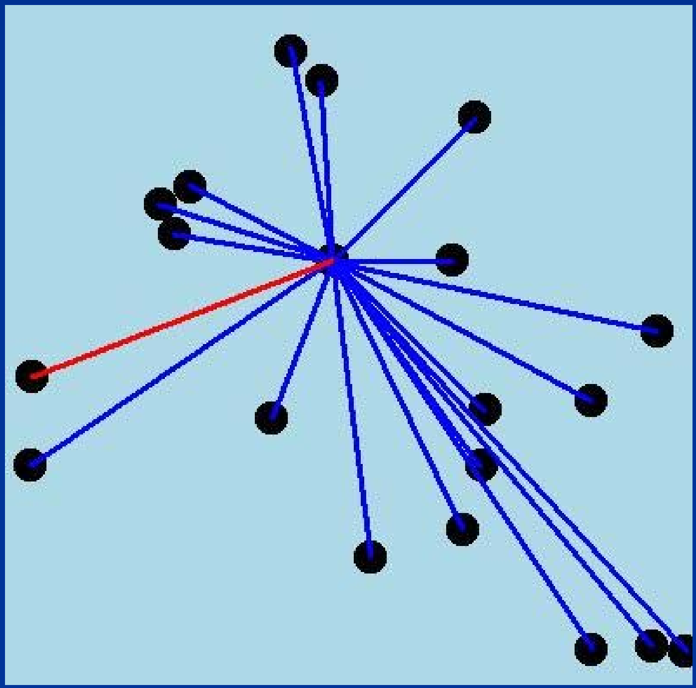
Single Sender Topology

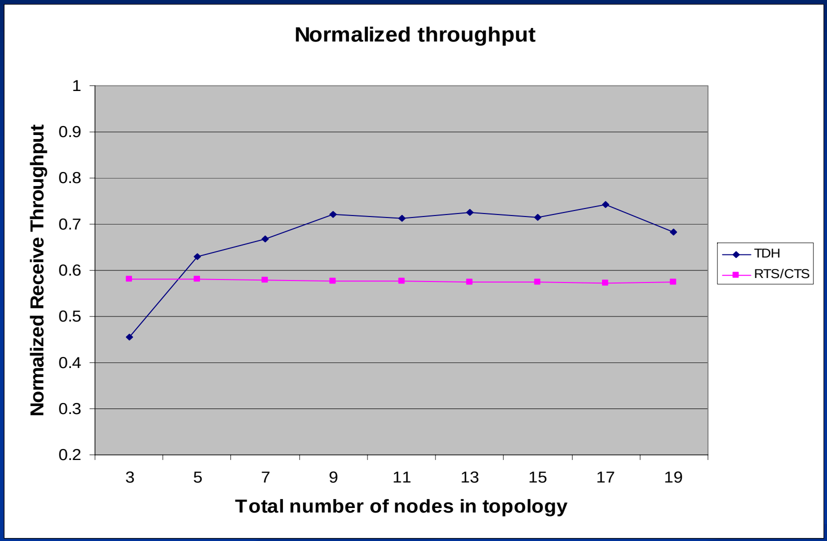


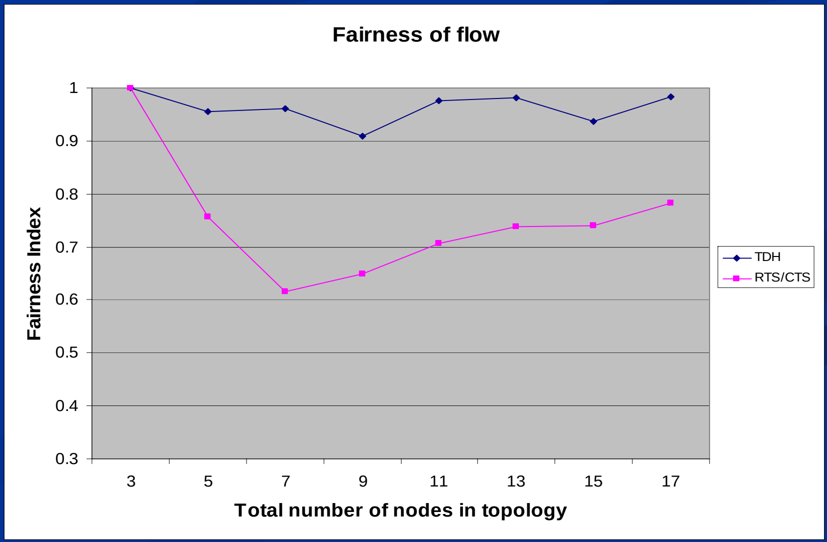




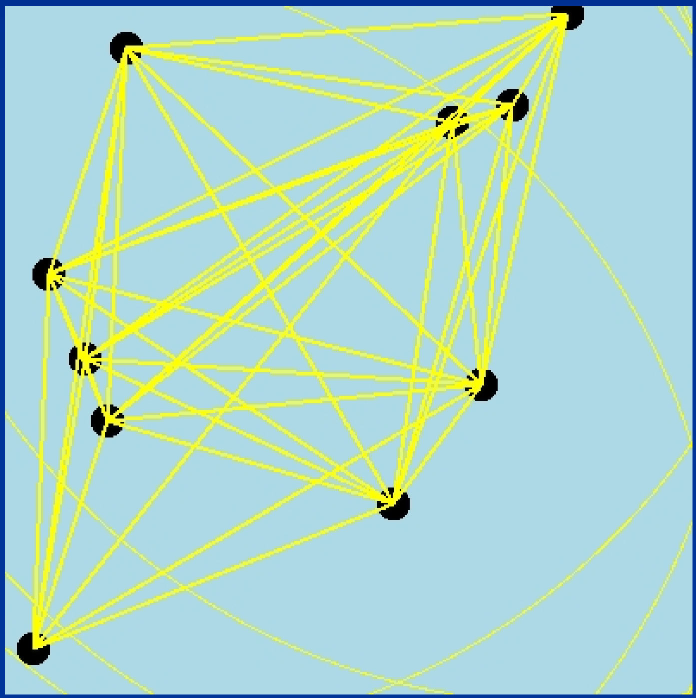
Access Point Topology

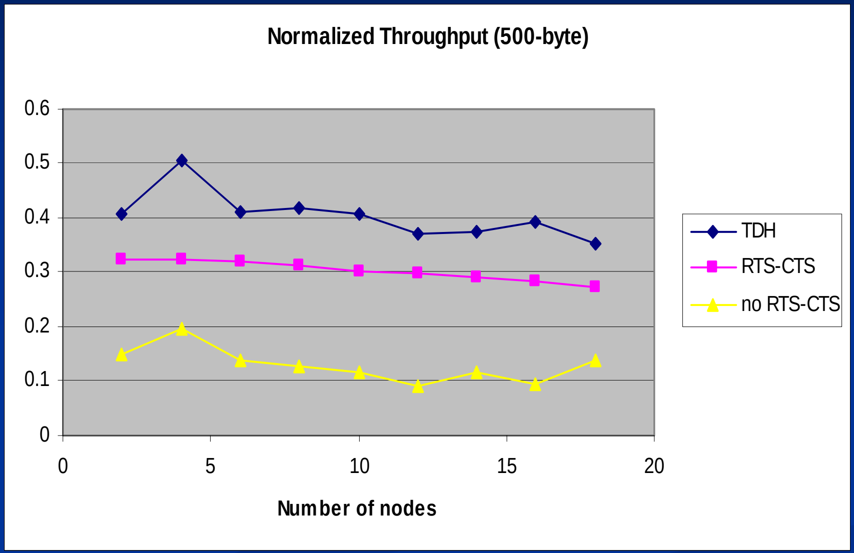


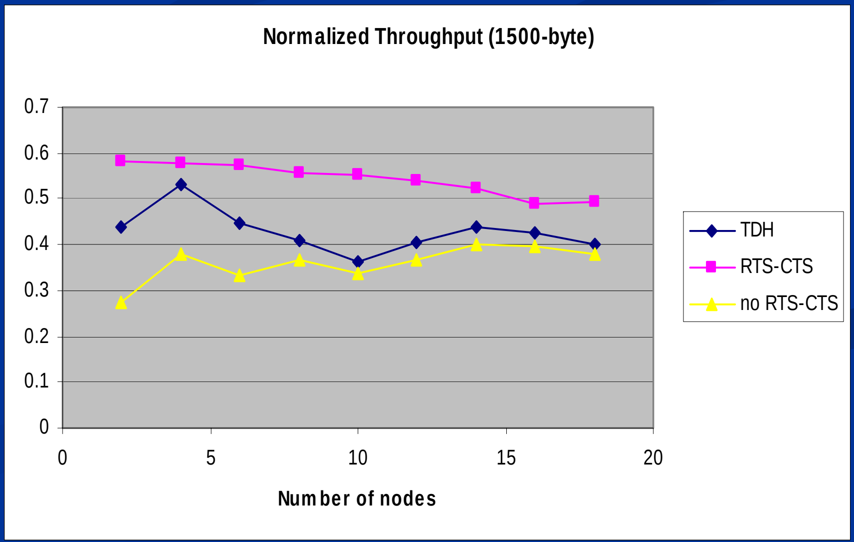




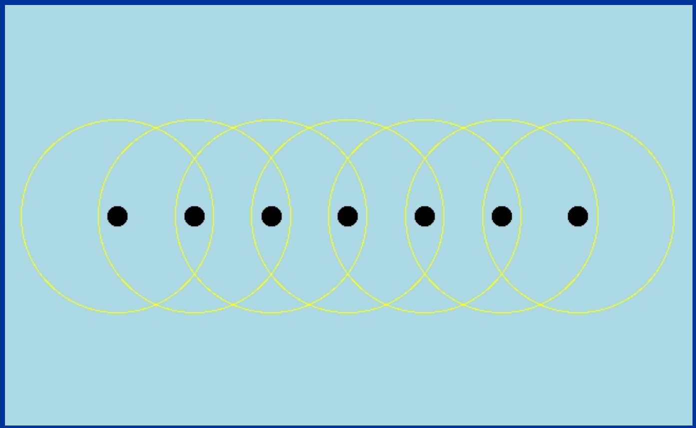
Clique Topology

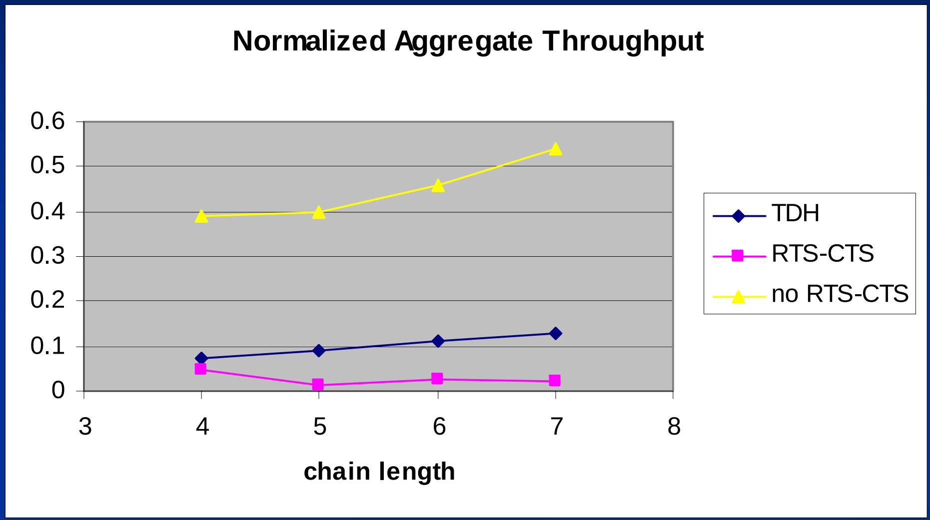


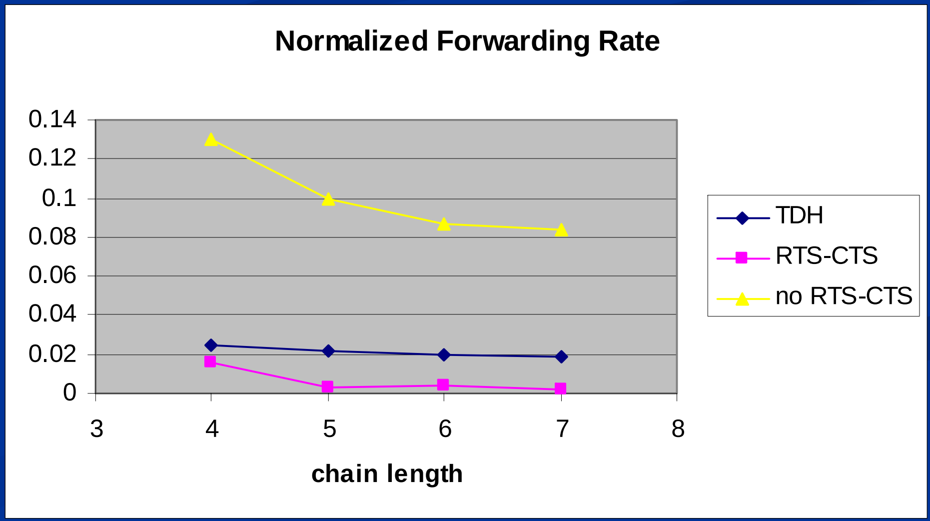




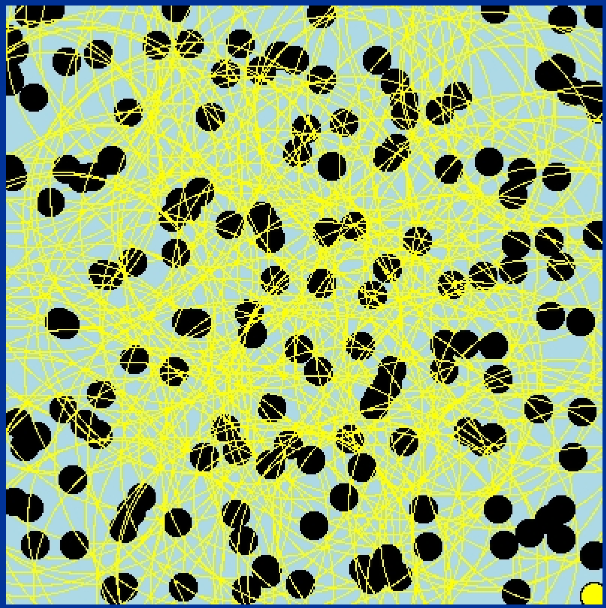
Chain Topology

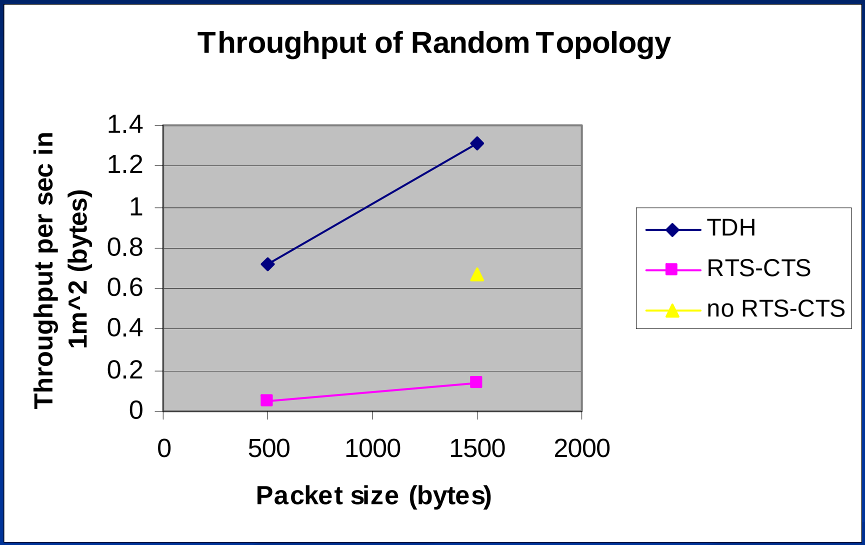


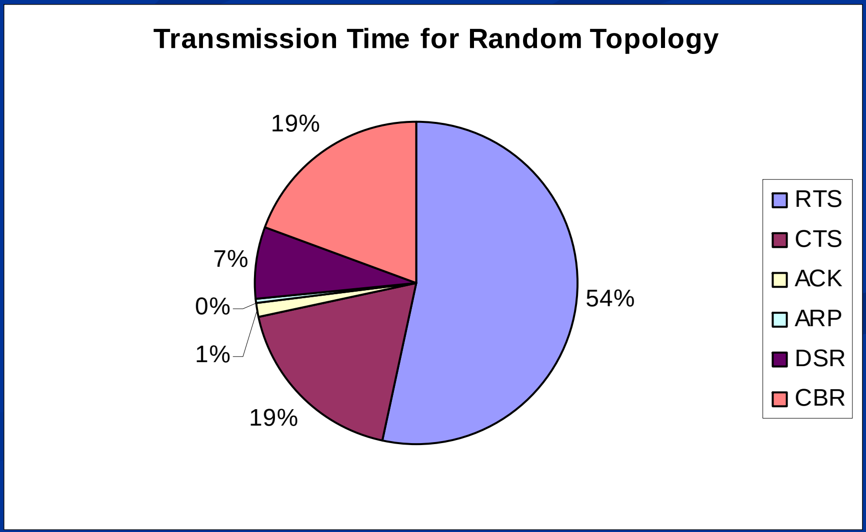




Random Topology







Summary of Results

Normalized Throughput Improvement with TDH

Scenario	TDH	802.11	Improvement
Single Sender	0.716	0.573	125 %
Access Point	0.742	0.572	130 %
Clique	0.507	0.325	156 %
Chain	0.128	0.020	640 %
Random*	1.310	0.138	949 %

* Throughput per m^2

Outline

- Overview
- Proposed Algorithm
- Implementation
- Simulation Results
- Conclusions and Future Work

What We Gained from TDH

- 802.11 performance was significantly degraded by its backoff mechanism and RTS-CTS resulted in more harm than good for congested networks
- TDH is a simple scheme that avoids some of these problems and gives better performance
- TDH treats each Tx-Rx independently, resulting in better fairness

Some Weaknesses of TDH

- High drop rates due to collisions
- Performance degradation in very sparse networks
- Assumption of static p and fixed slot size

Future Work

- Analysis of impact of collisions on transport layer
- Better slot management using subslots
- Dynamic adjustment of p
- Investigate with higher bit rates like 54Mbps

Questions

Thank you!

wwcheng@mit.edu

References

[spec802.11] ANSI/IEEE Std 802.11 "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications." 1999.

[Xu03] K. Xu, M. Gerla. "Effectiveness of RTS/CTS Handshake in IEEE 802.11 based Ad Hoc Networks" Mobicom 2003

[SXu02] S. Xu, T. Saadawi "Does the IEEE 802.11 MAC protocol Work Well in Multihop Wireless Ad Hoc Networks?" IEEE Communication Magazine, Vol 39, N. 6, June 2001, pp 130-137.

[Shepard95] T. Shepard. "Decentralized Channel Management in Scalable Multihop Spread Spectrum Packet Radio Networks" MIT PhD Thesis. 1995.

[Li01] J. Li, C. Blake, D. De Couto, H. Lee, R. Morris "Capacity of Ad Hoc Wireless Networks"

[Bharghavan94] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. "MACAW: A Media-Access Protocol for Packet Radio". In Proceedings of ACM SIGCOMM, 1994.

[Monarch] D. Johnson, Wireless and Mobility Extensions to ns-2, http://www.monarch.cs.rice.edu/cmu-ns.html