Granting Silence to Avoid Wireless Collisions

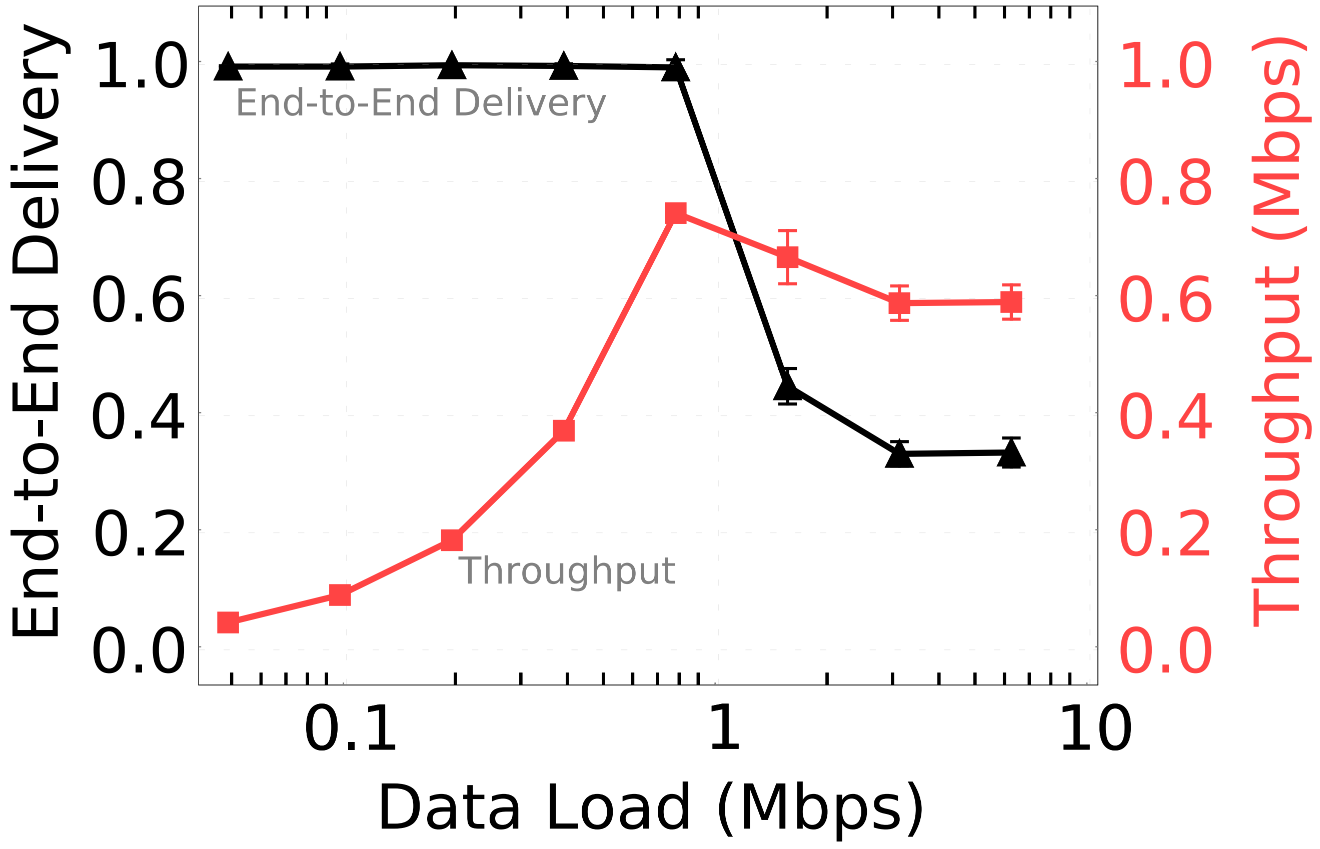
Jung Il Choi, Mayank Jain, Maria A. Kazandjieva, and Philip Levis

October 6, 2010
ICNP 2010

STANFORD
UNIVERSITY

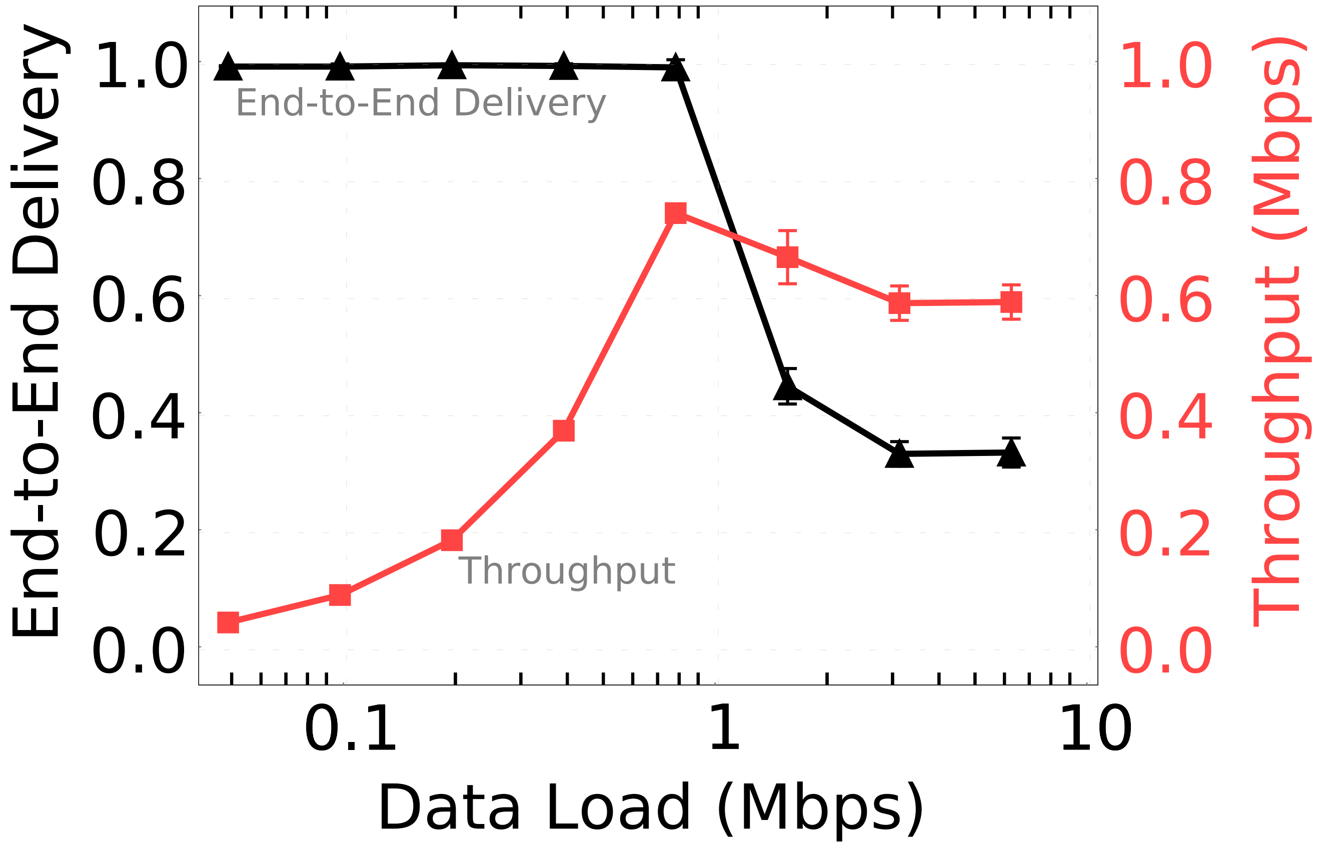
Wireless Mesh and CSMA

- One UDP flow along a static 4-hop route in 802.11b mesh testbed



Wireless Mesh and CSMA

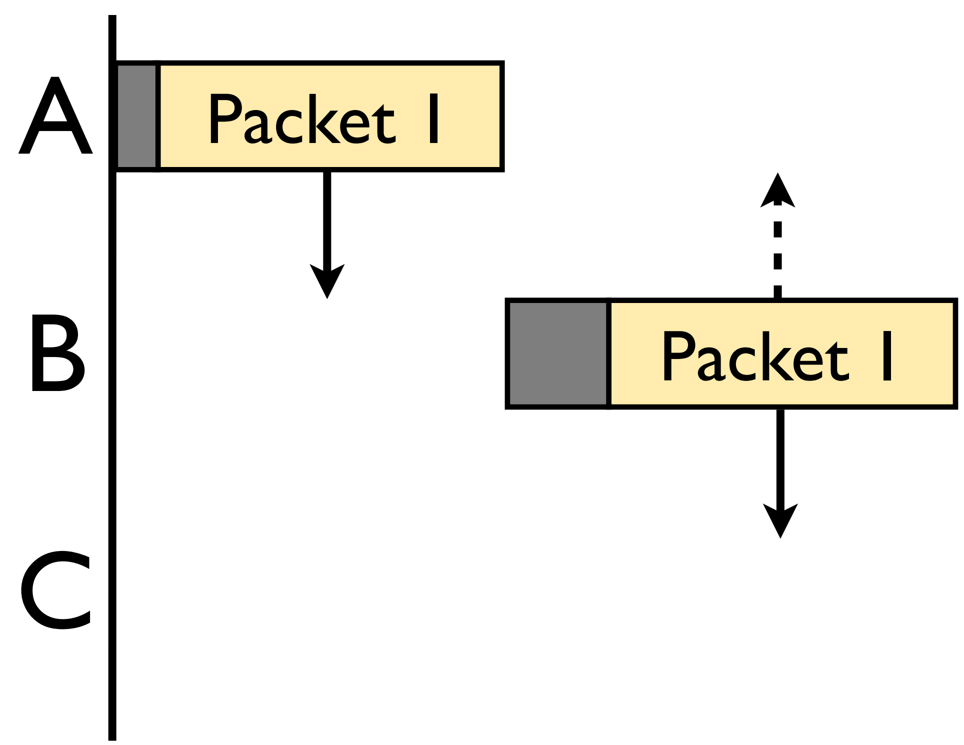
- One UDP flow along a static 4-hop route in 802.11b mesh testbed



Sending more packets causes throughput decrease

Self-Interference

- Packets within a flow collide due to hidden terminals
- Known problem reported by Li et al.[1] and Vyas et al.[2]

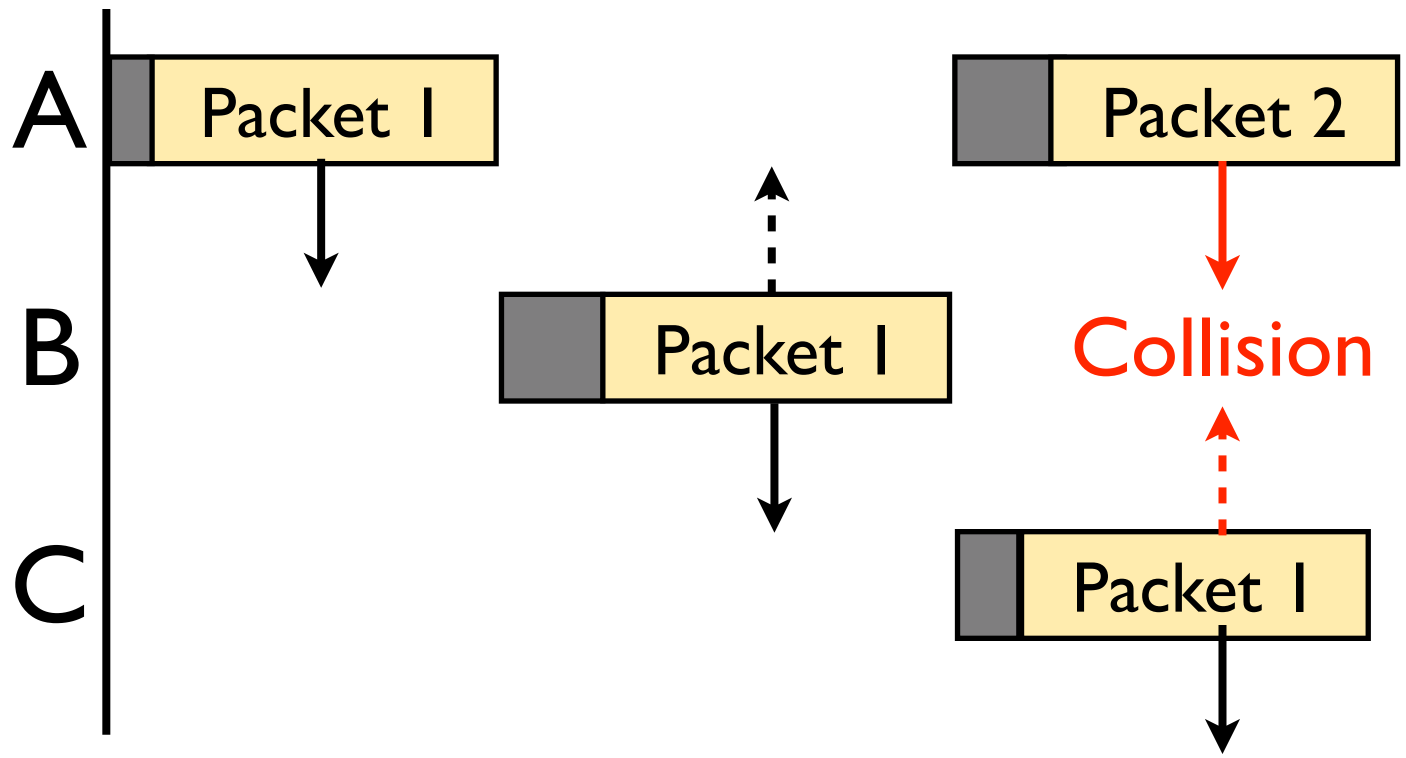


(1) J. Li, C. Blake, D. S. D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. ACM MobiCom, 2001

(2) A. Vyas and F. Tobagi. Impact of interference on the throughput of a multihop path in a wireless network. ICST BROADNETS, 2006

Self-Interference

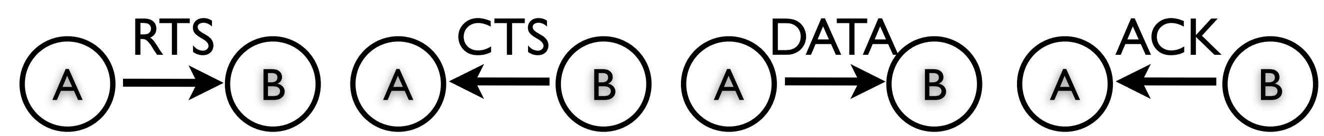
- Packets within a flow collide due to hidden terminals
- Known problem reported by Li et al.[1] and Vyas et al.[2]



(1) J. Li, C. Blake, D. S. D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. ACM MobiCom, 2001

(2) A. Vyas and F. Tobagi. Impact of interference on the throughput of a multihop path in a wireless network. ICST BROADNETS, 2006

Practical Solution?

- Can we fix this problem with existing hardware?
- One candidate: RTS/CTS 

  - Can help avoid collisions due to hidden terminals
  - Incurs heavy overhead: Control packets are sent at 1 or 2 Mbps

Bitrate	CSMA	RTS/CTS	Overhead
1 Mbps	0.79	0.76	4.0%
2 Mbps	1.44	1.35	6.6%
5.5 Mbps	3.36	2.89	14.1%
11 Mbps	5.89	4.42	25.1%

Grant-To-Send (GTS)

- A novel collision avoidance mechanism for CSMA based wireless mesh networks
- Instead of avoiding collisions for packets a node would transmit, GTS avoids collisions with packets the node expects to hear
  - A transmitting node grants a clear wireless channel to the receiver
- Generic: Works for both 802.11 and 802.15.4
- No control packets, low overhead, compatible with existing hardware.

In a Nutshell

- Present Grant-to-Send (GTS). Analyze and evaluate GTS through simulations and experiments
- GTS outperforms CSMA and RTS/CTS

  4-hop UDP throughput increases by 23%, 96% of the maximum possible

- GTS can replace existing per-protocol collision avoidance mechanisms in sensor networks

  Can prevent inter-protocol interactions

Talk Outline

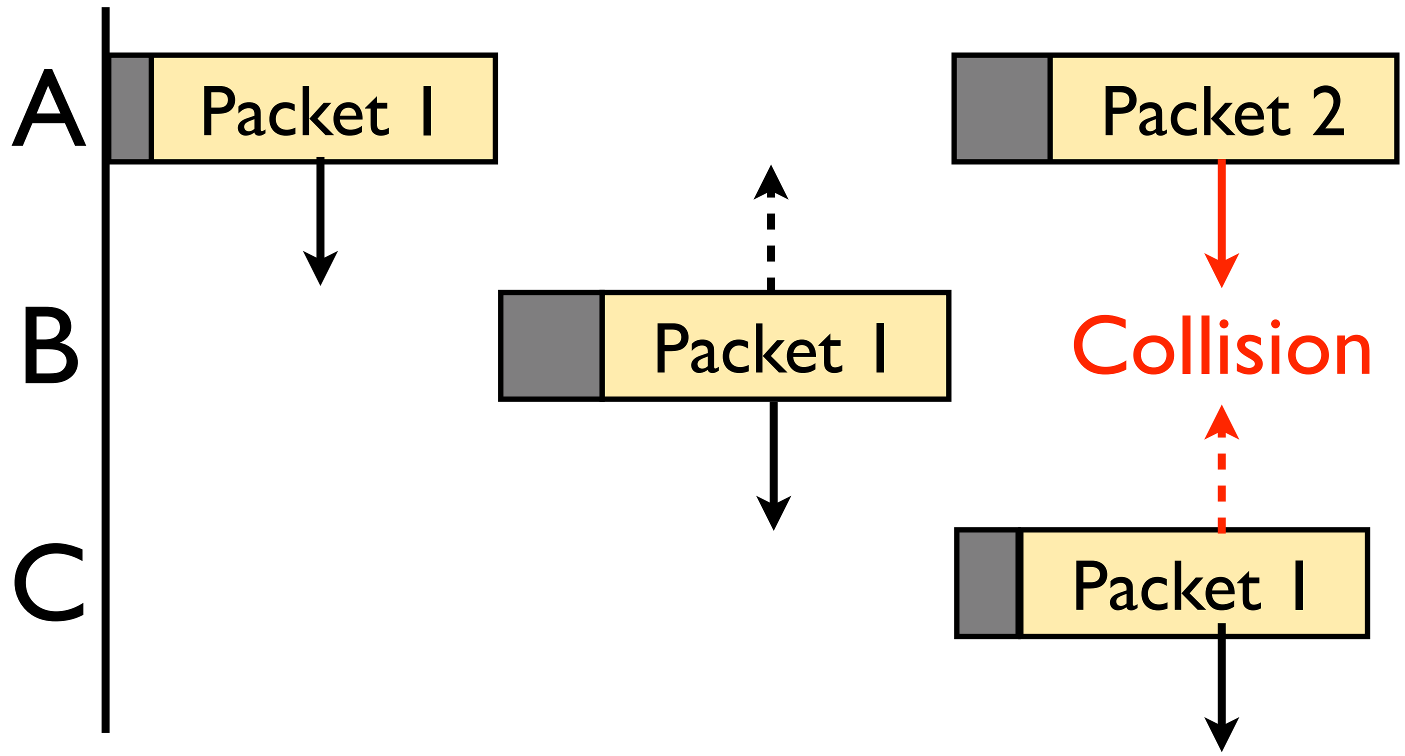
- Grant-To-Send Mechanism
- Optimal Grant Duration
- GTS in 802.11 : UDP
- GTS in 802.15.4 : CTP and Deluge
- Limitations of GTS

Mechanism

- Every data transmission contains a "grant duration"
- The transmitter and nodes that overhear this transmission must be silent for the duration after the transmission
- Only the receiver can transmit for the grant duration
    - i.e. the transmitter "grants" the receiver to send

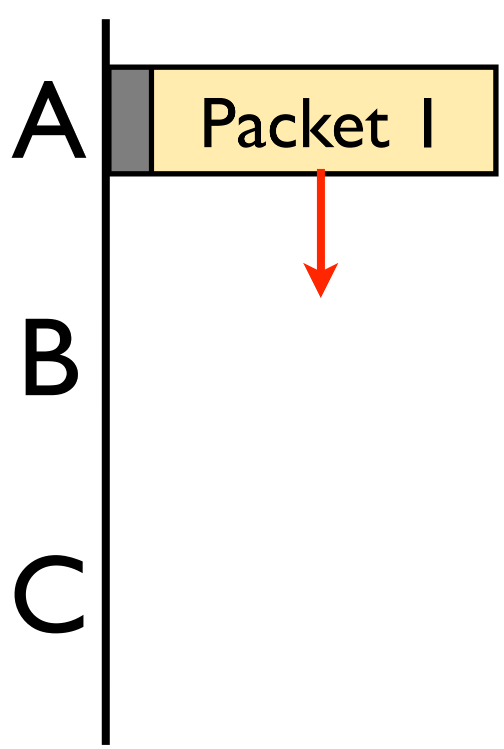
Mechanism

- Every data transmission contains a "grant duration"
- The transmitter and nodes that overhear this transmission must be silent for the duration after the transmission
- Only the receiver can transmit for the grant duration
  - i.e. the transmitter "grants" the receiver to send



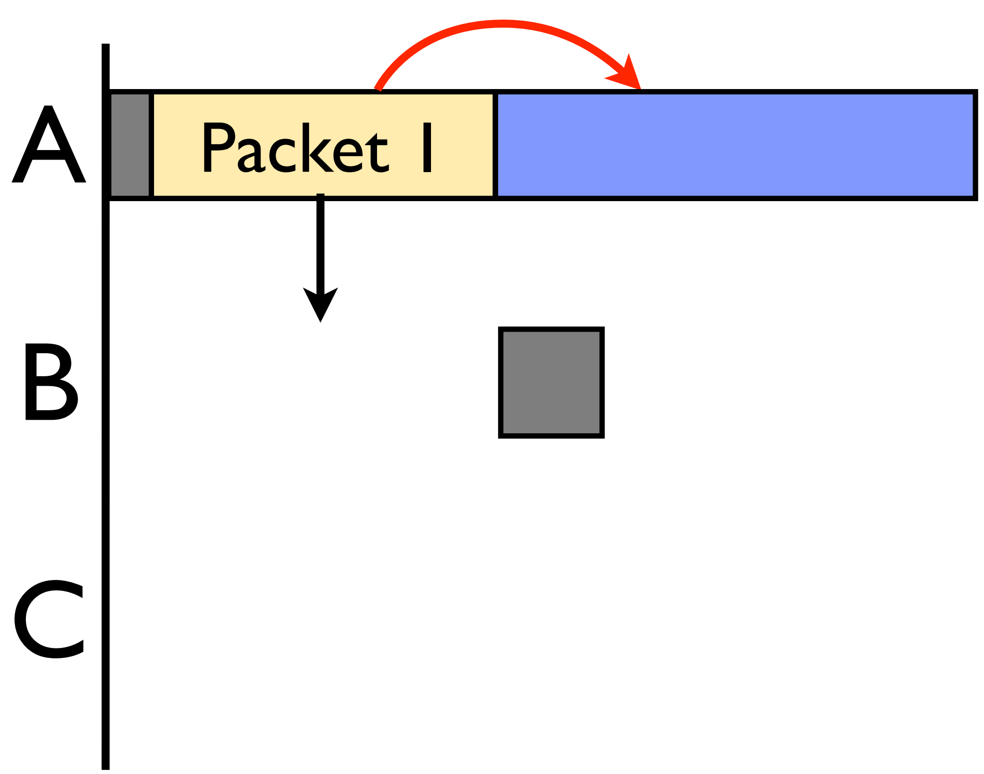
Mechanism

- Every data transmission contains a “grant duration”
- The transmitter and nodes that overhear this transmission must be silent for the duration after the transmission
- Only the receiver can transmit for the grant duration
  - i.e. the transmitter “grants” the receiver to send



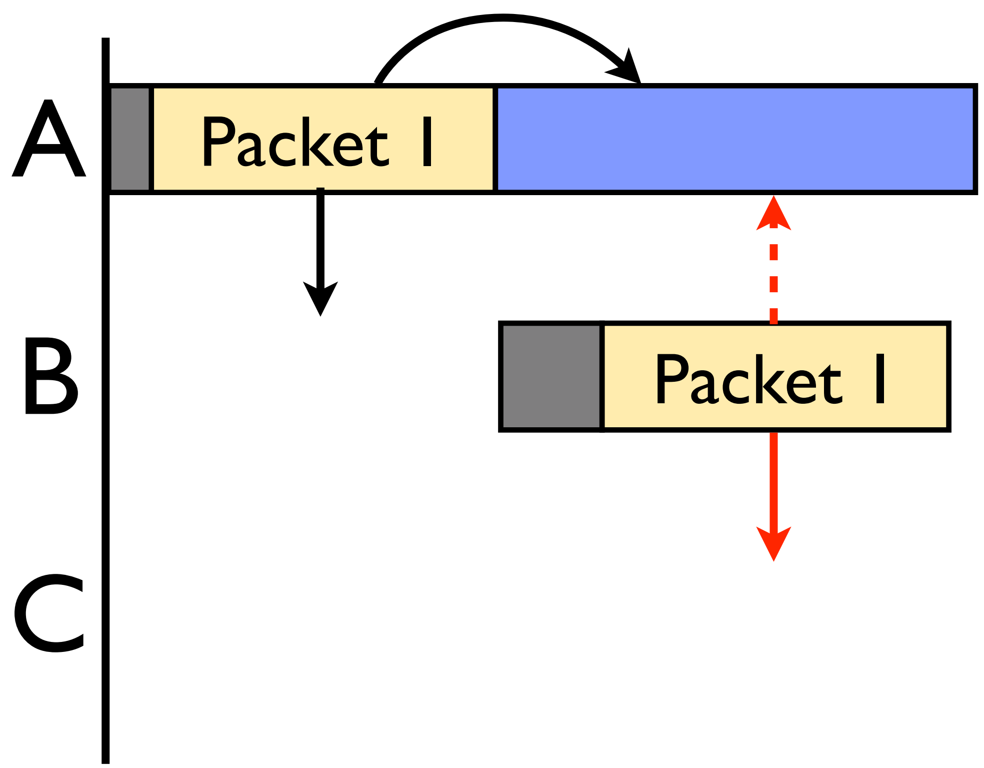
Mechanism

- Every data transmission contains a "grant duration"
- The transmitter and nodes that overhear this transmission must be silent for the duration after the transmission
- Only the receiver can transmit for the grant duration
  - i.e. the transmitter "grants" the receiver to send



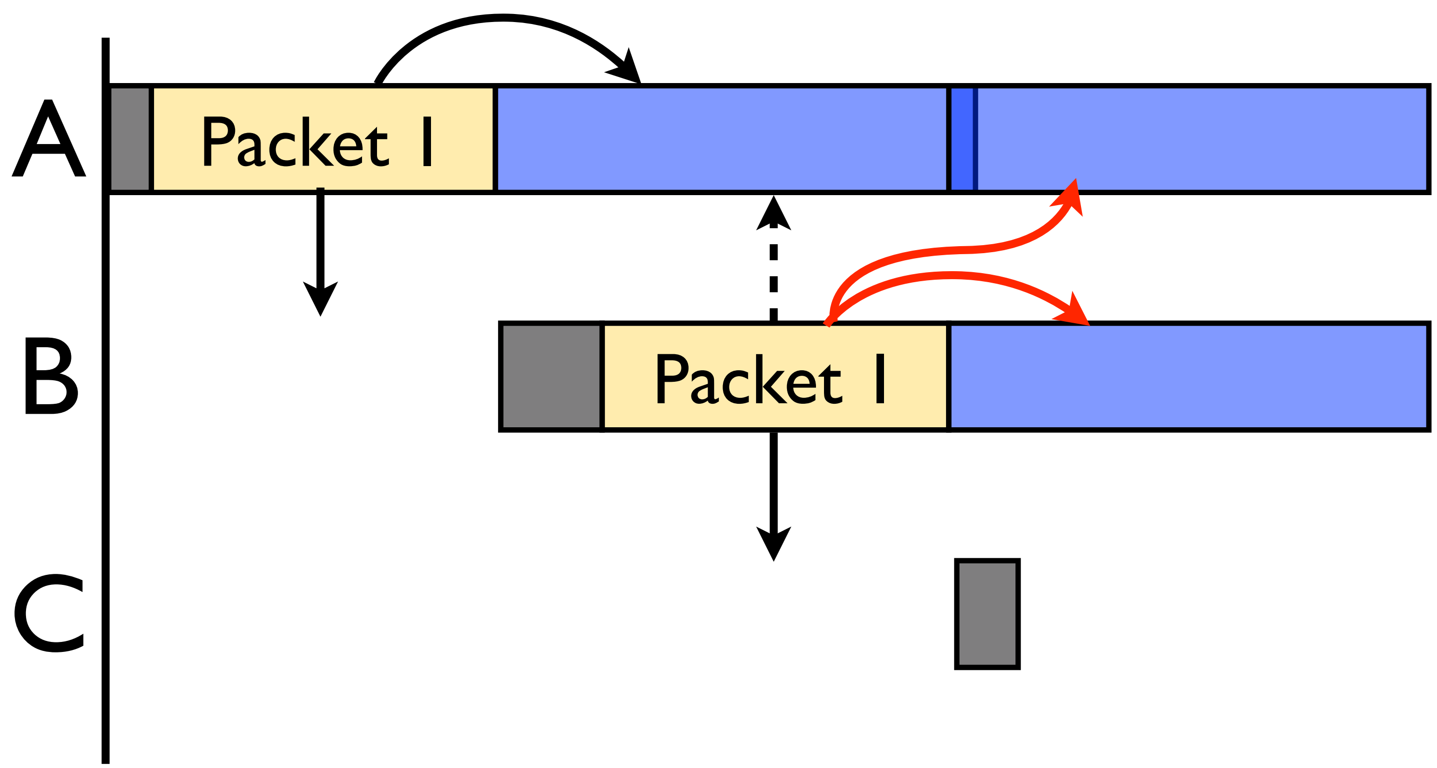
Mechanism

- Every data transmission contains a “grant duration”
- The transmitter and nodes that overhear this transmission must be silent for the duration after the transmission
- Only the receiver can transmit for the grant duration
    - i.e. the transmitter “grants” the receiver to send



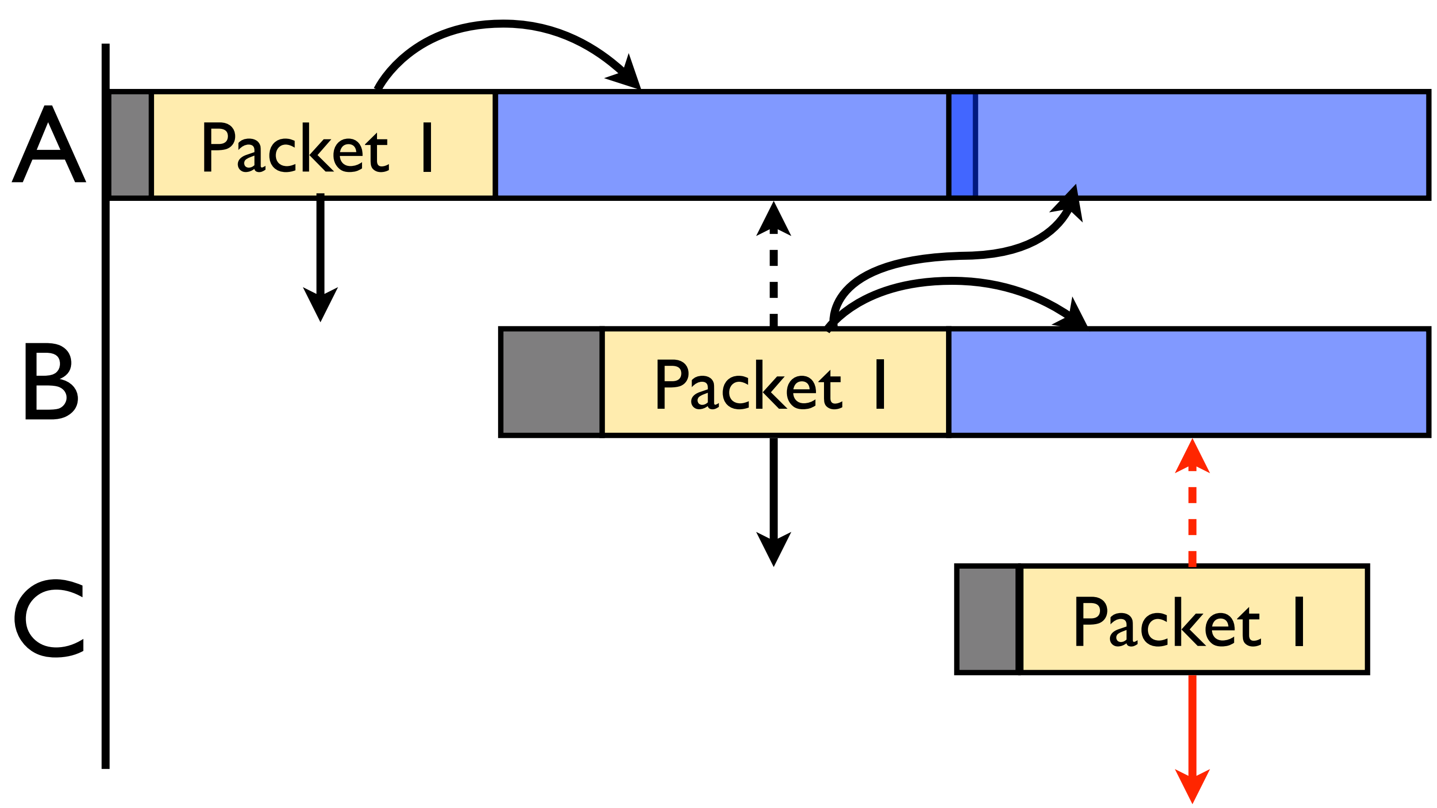
Mechanism

- Every data transmission contains a "grant duration"
- The transmitter and nodes that overhear this transmission must be silent for the duration after the transmission
- Only the receiver can transmit for the grant duration
  - i.e. the transmitter "grants" the receiver to send



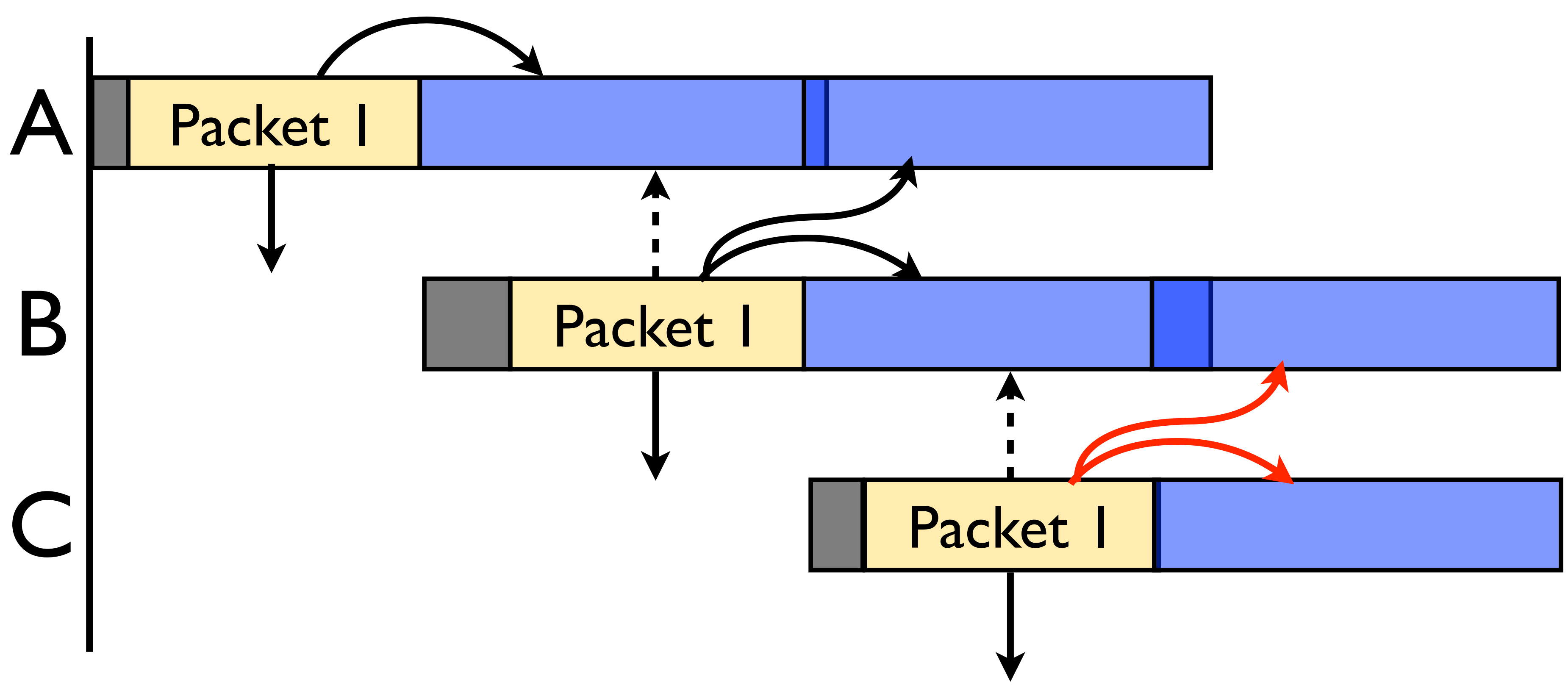
Mechanism

- Every data transmission contains a "grant duration"
- The transmitter and nodes that overhear this transmission must be silent for the duration after the transmission
- Only the receiver can transmit for the grant duration
  - i.e. the transmitter "grants" the receiver to send



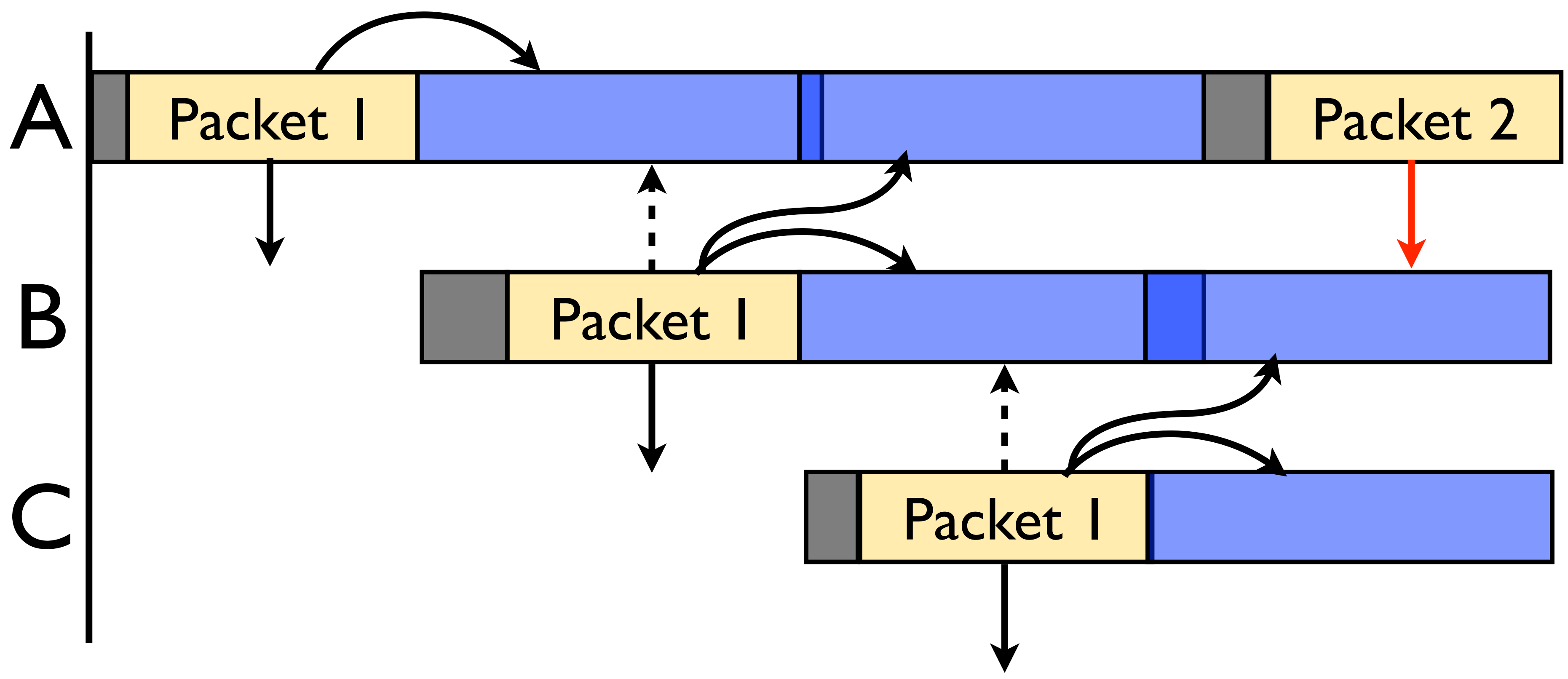
Mechanism

- Every data transmission contains a “grant duration”
- The transmitter and nodes that overhear this transmission must be silent for the duration after the transmission
- Only the receiver can transmit for the grant duration
    - i.e. the transmitter “grants” the receiver to send



Mechanism

- Every data transmission contains a "grant duration"
- The transmitter and nodes that overhear this transmission must be silent for the duration after the transmission
- Only the receiver can transmit for the grant duration
  - i.e. the transmitter "grants" the receiver to send



Implementation for 802.11

- Reuse the Network Allocation Vector field (NAV)
  - Originally, NAV is used to protect the current packet exchange: RTS sets NAV duration CTS+DATA+ACK

	NAV duration	Suppressed nodes
Original 802.11	Protects current packet exchange	Overhearing nodes
Grant-to-Send	Protects expected response from receiver	Overhearing nodes and transmitter

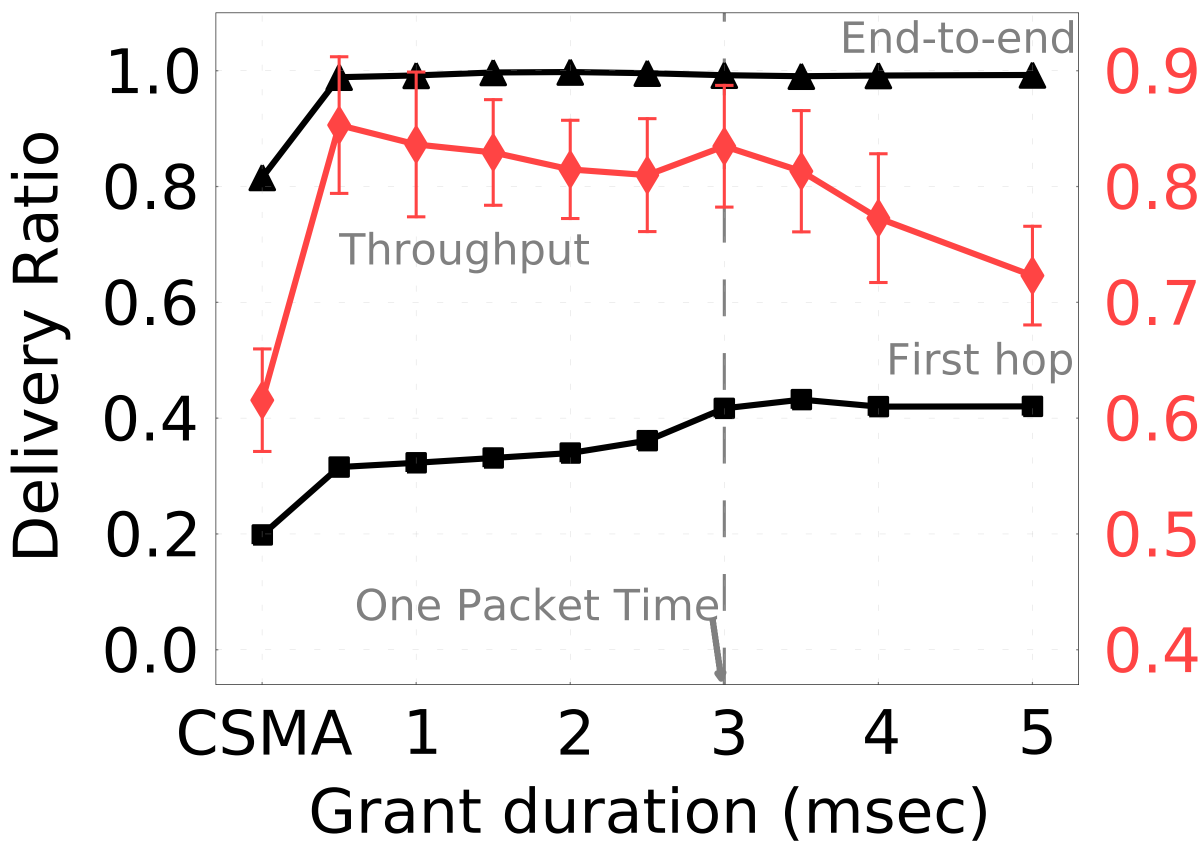
Implementation

- 802.11
  - 11 lines of driver code
  - No overhead in data packets
  - Works with MadWiFi and ath9k drivers with Atheros cards
- 802.15.4
  - 50 lines of TinyOS code
  - 9B RAM
- Both implementations work with existing hardware

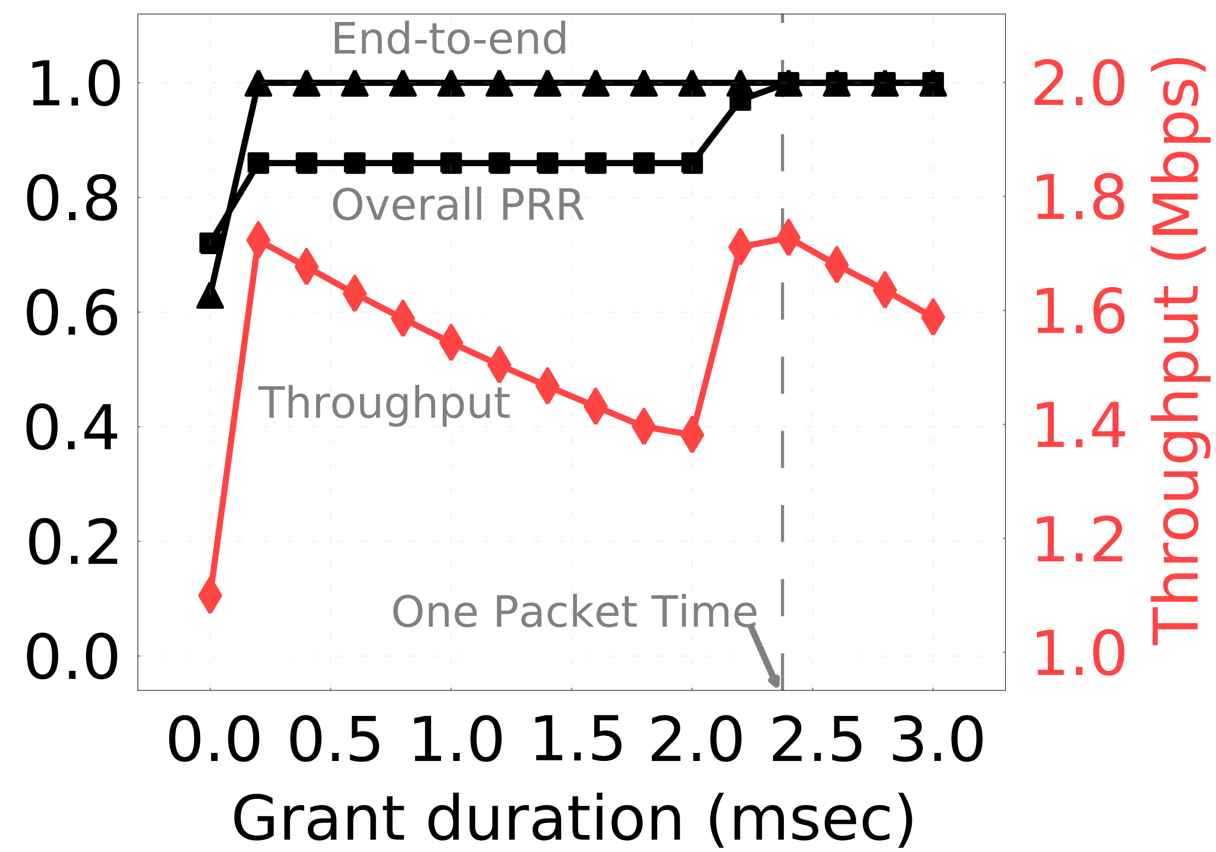
Talk Outline

- Grant-To-Send Mechanism
- Optimal Grant Duration
- GTS in 802.11 : UDP
- GTS in 802.15.4 : CTP and Deluge
- Limitations of GTS

Optimal Grant Duration



4-hop experiment

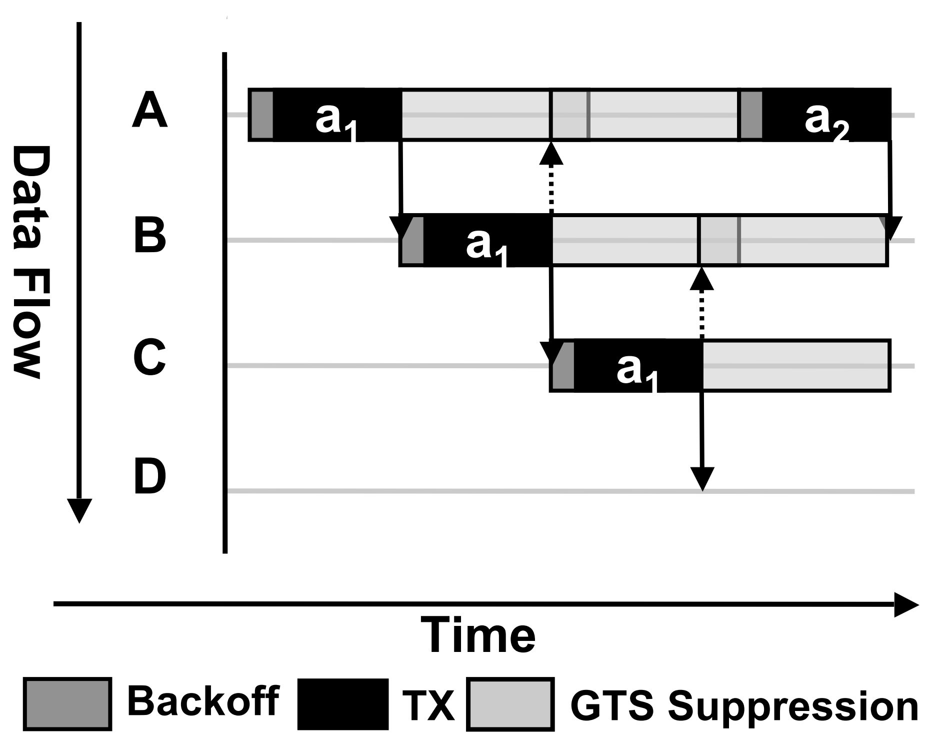


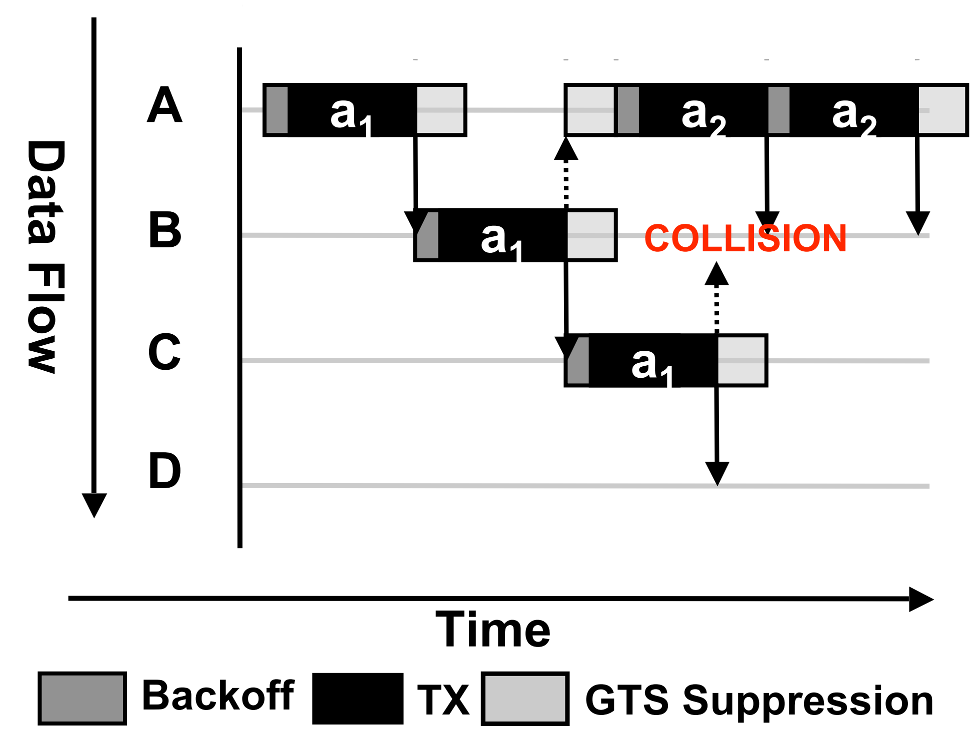
6-hop simulation

- One packet time seems to be the optimal
- Intuition: the transmitter and its neighbors wait for the recipient to forward one packet

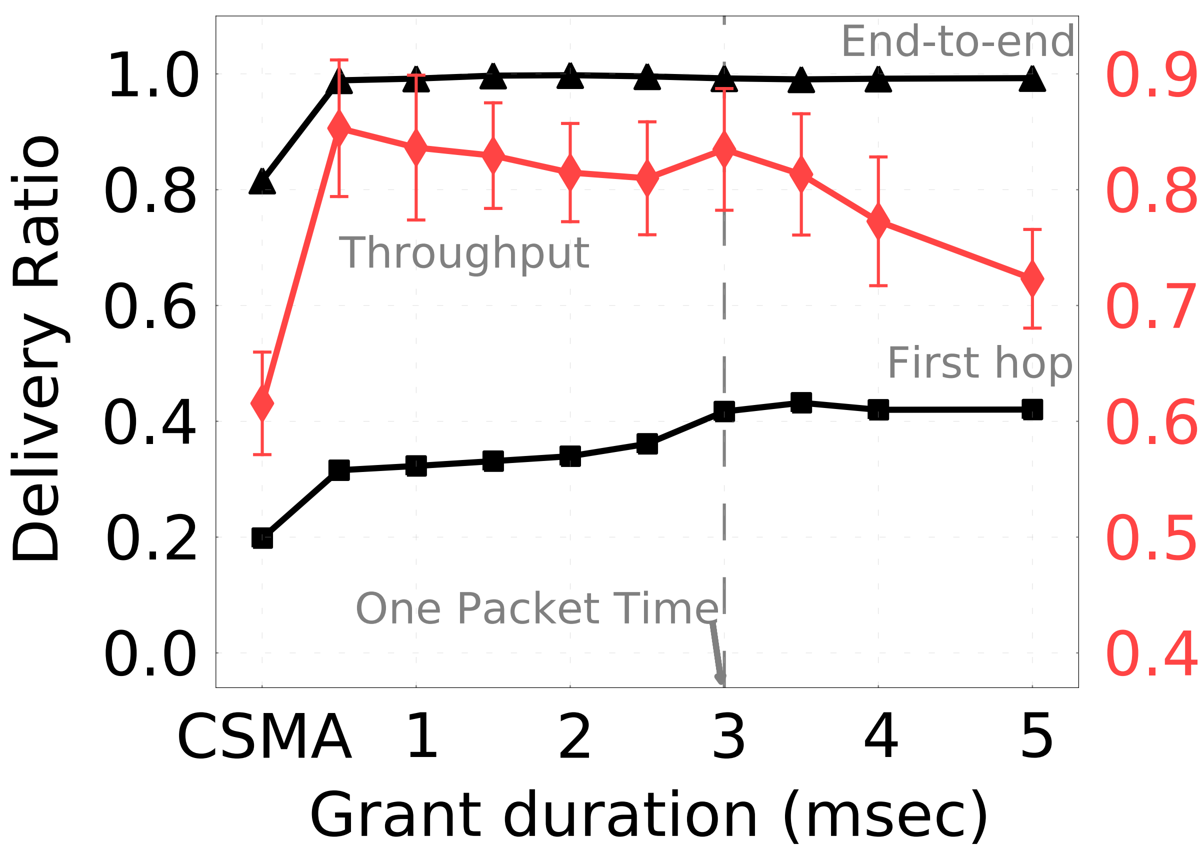
Long and Short Grants

- Long grants
  - avoid more collisions
  - may cause unnecessary idle times
- Short grants
  - prioritize forwarders
  - waste more channel time due to collisions

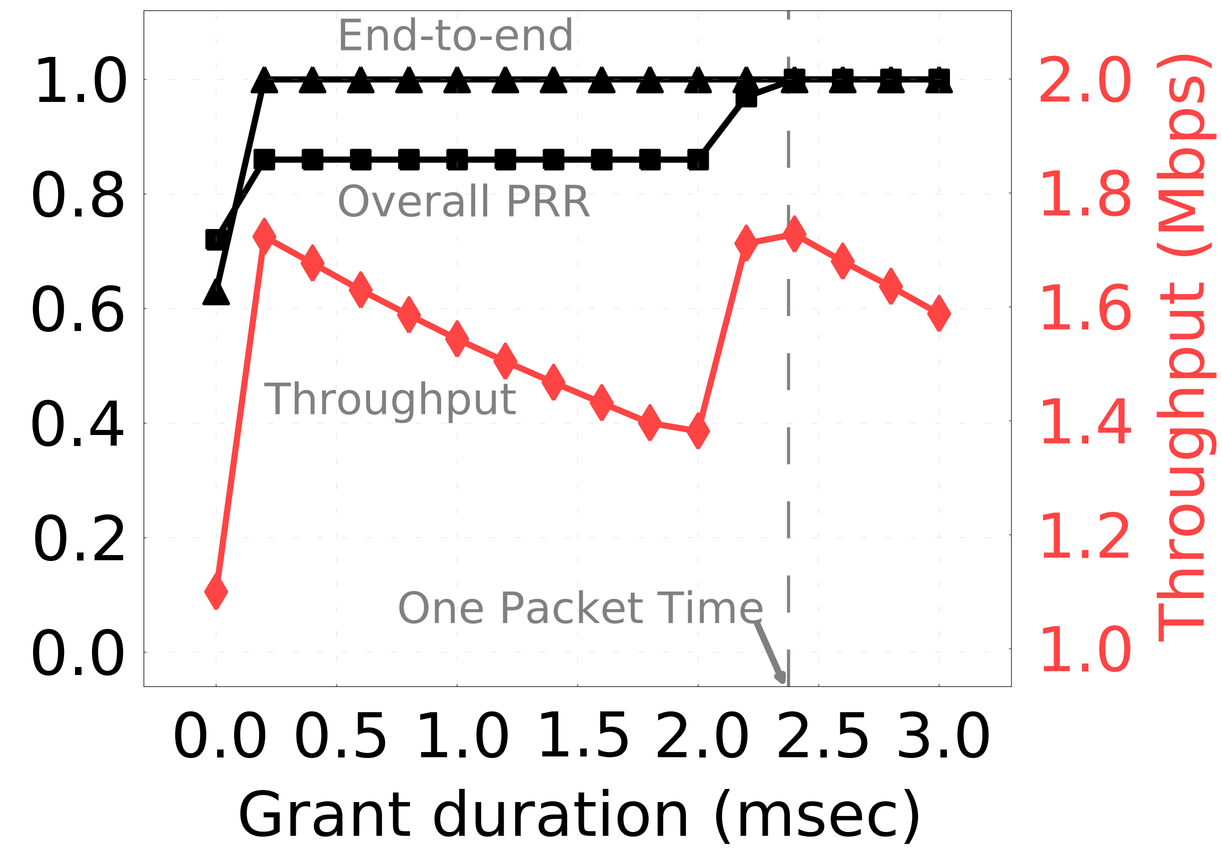




Analysis



4-hop experiment



6-hop simulation

From analysis, throughput $T(g) = \begin{cases} \frac{B}{3+k} & \text{if } g = 0 \\ \frac{B}{3+\frac{g}{p}} & \text{if } g < p \\ \frac{B}{2+\frac{g}{p}} & \text{if } g \geq p \end{cases}$ (k : 0.3~3)[1]

g : grant duration, p : packet time, B : link capacity

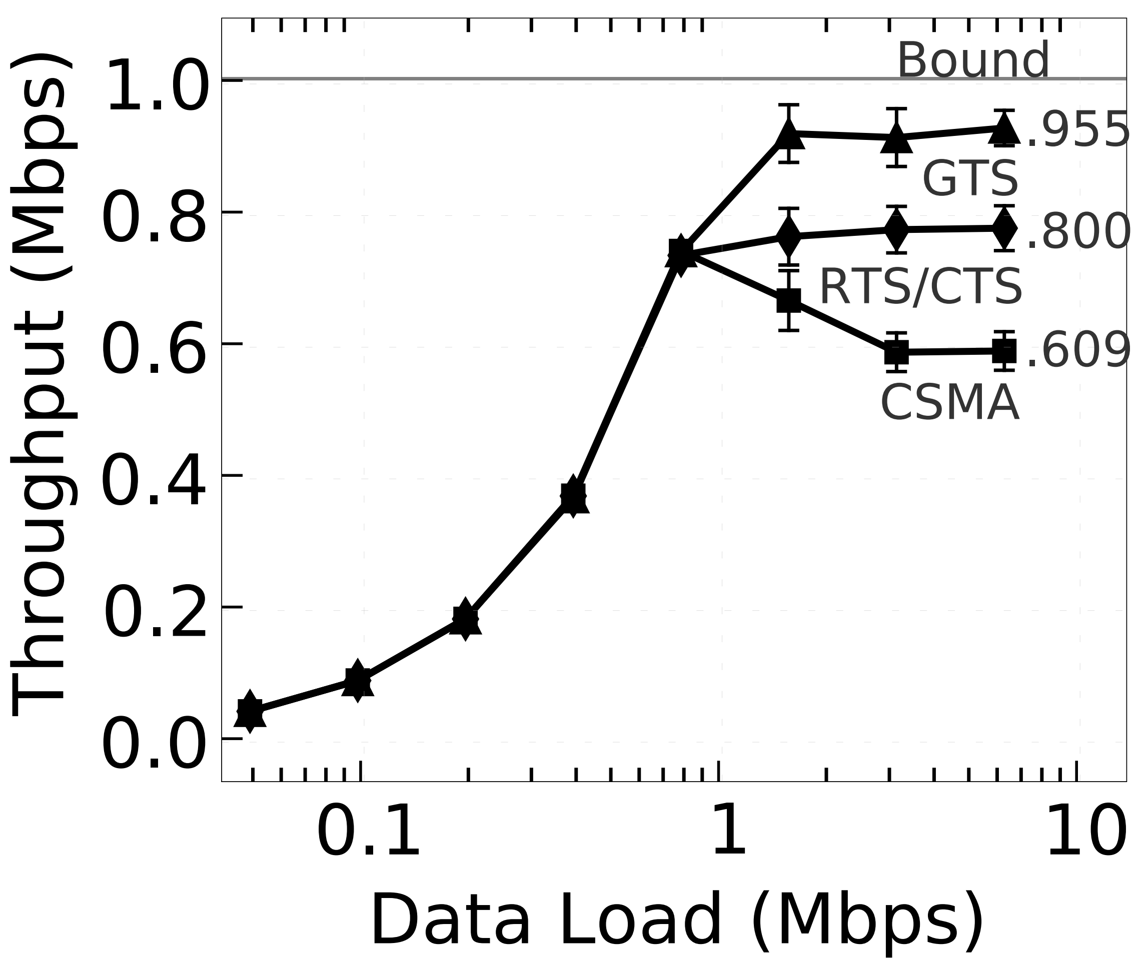
(1) A. Vyas and F. Tobagi. Impact of interference on the throughput of a multihop path in a wireless network. ICST BROADNETS, 2006

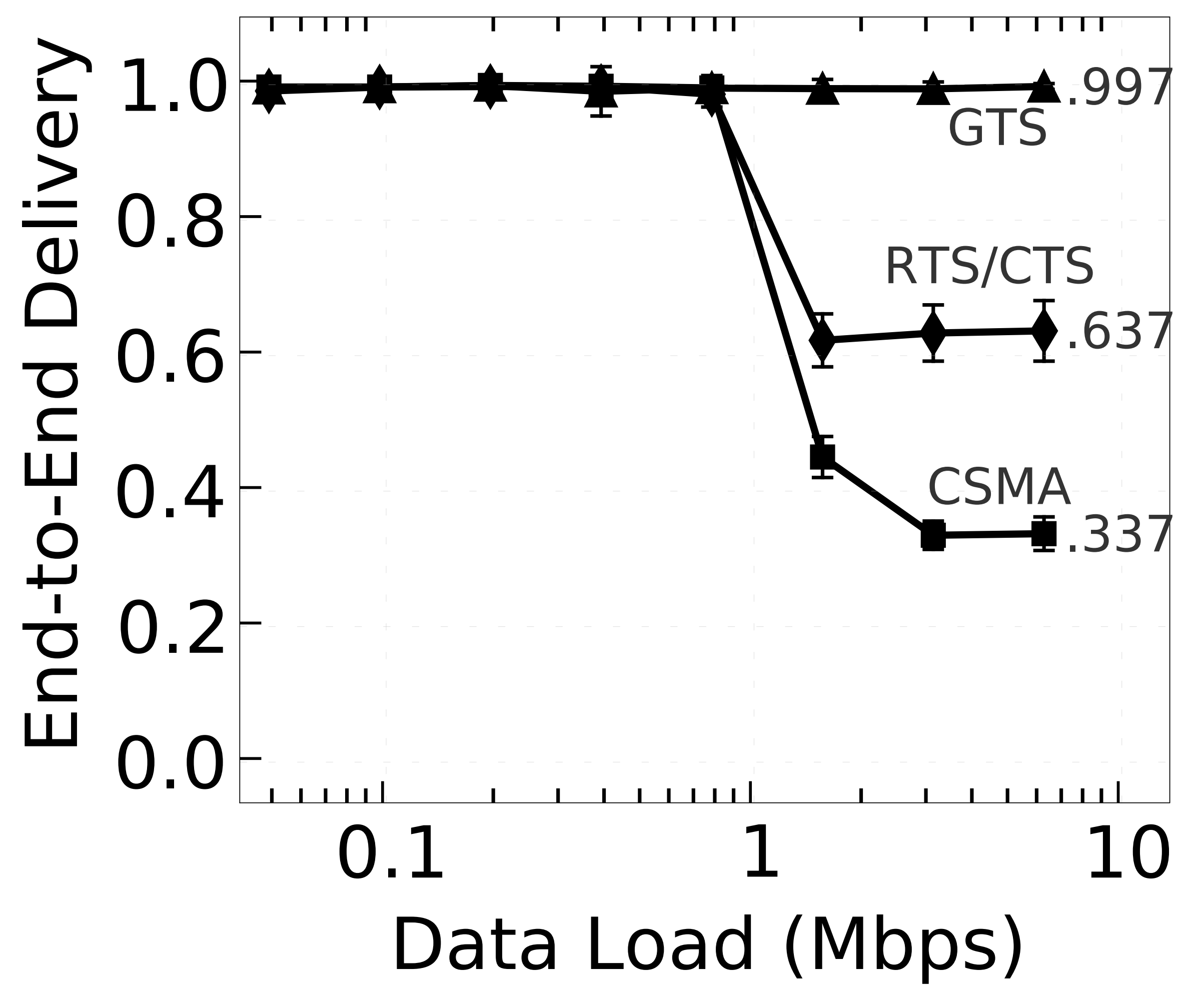
Talk Outline

- Grant-To-Send Mechanism
- Optimal Grant Duration
- GTS in 802.11 : UDP
- GTS in 802.15.4 : CTP and Deluge
- Limitations of GTS

CSMA, RTS/CTS, and GTS

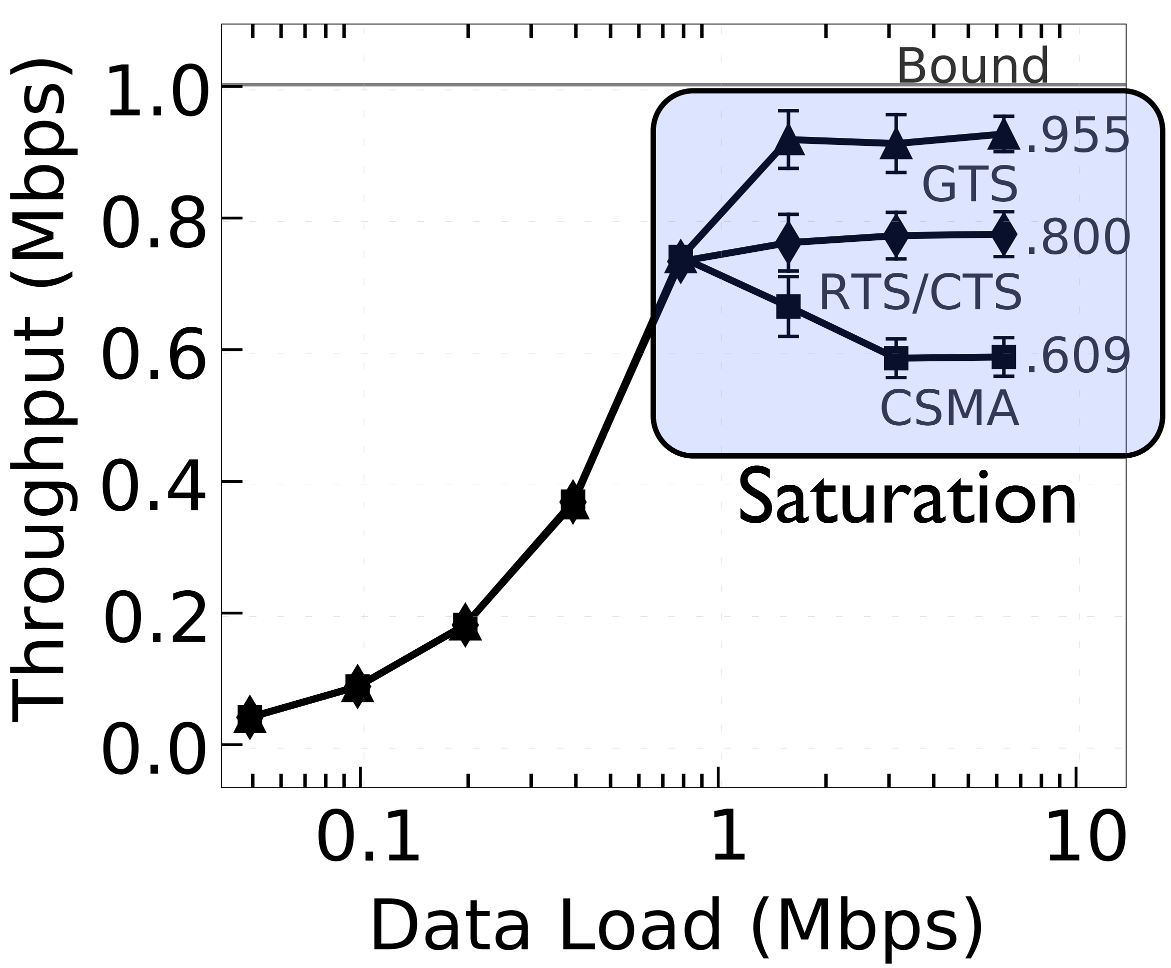
- 4-hop static route testbed experiment with 5.5Mbps bitrate
- GTS achieves 96% of the throughput upper bound

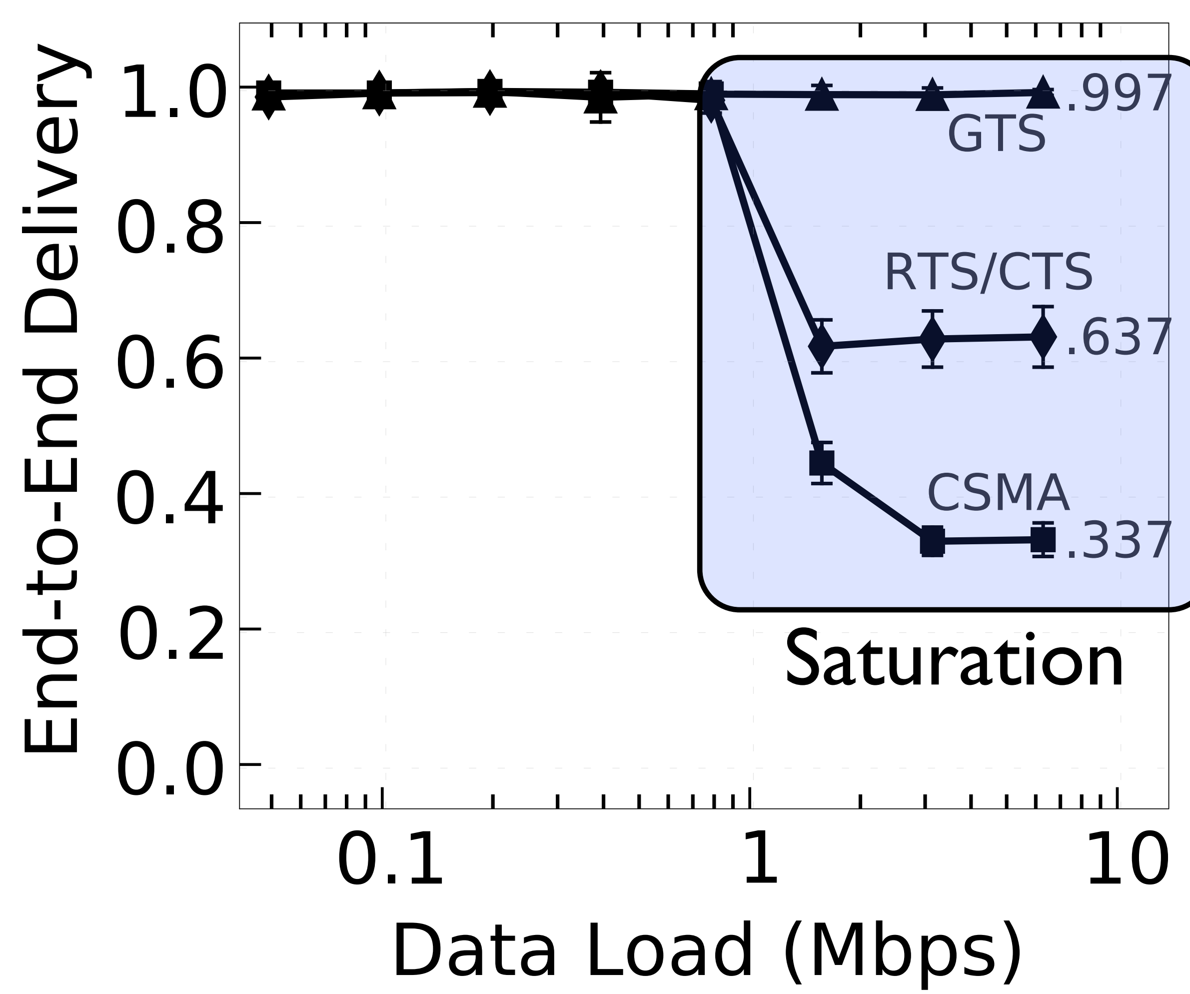




CSMA, RTS/CTS, and GTS

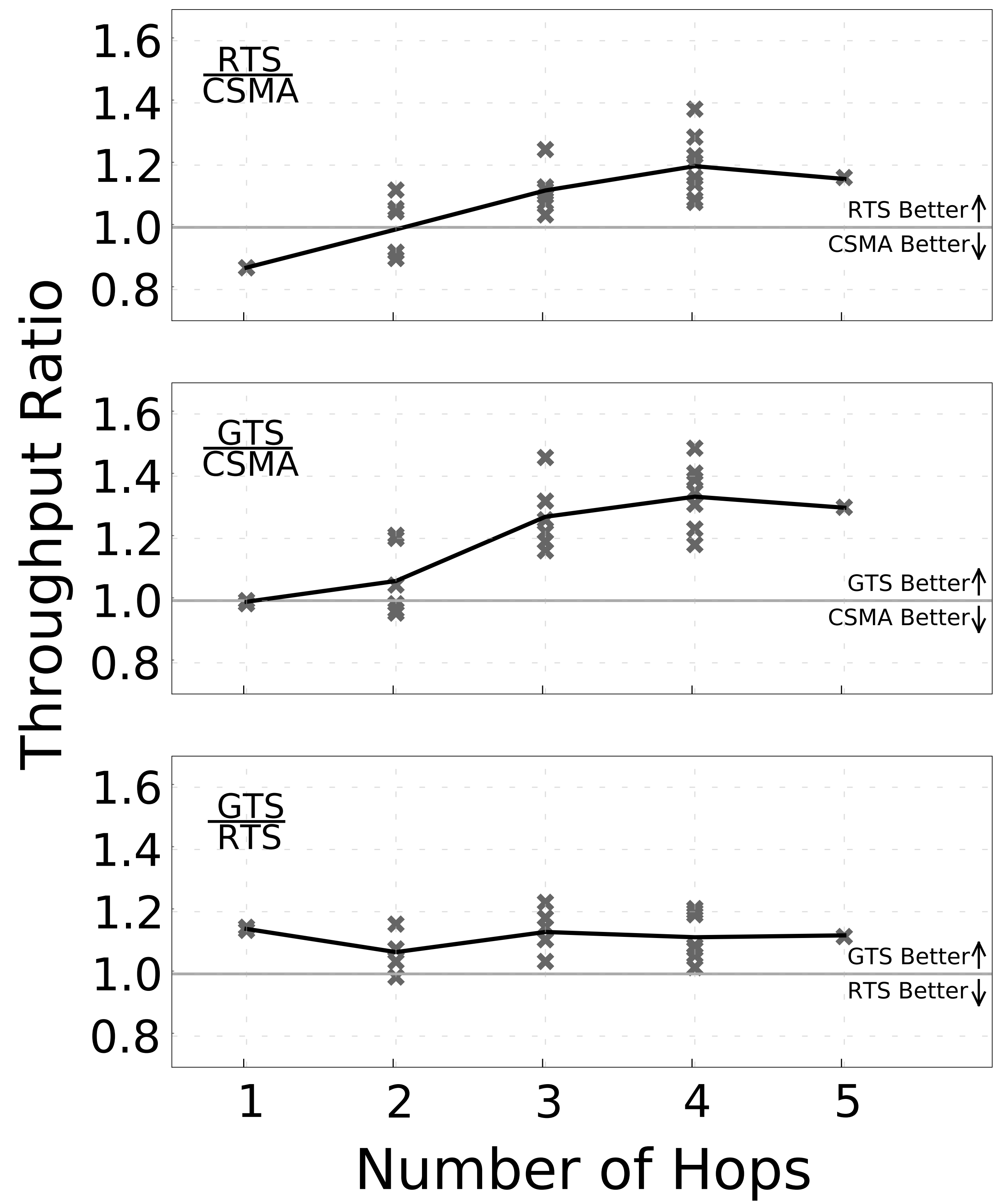
- 4-hop static route testbed experiment with 5.5Mbps bitrate
- GTS achieves 96% of the throughput upper bound





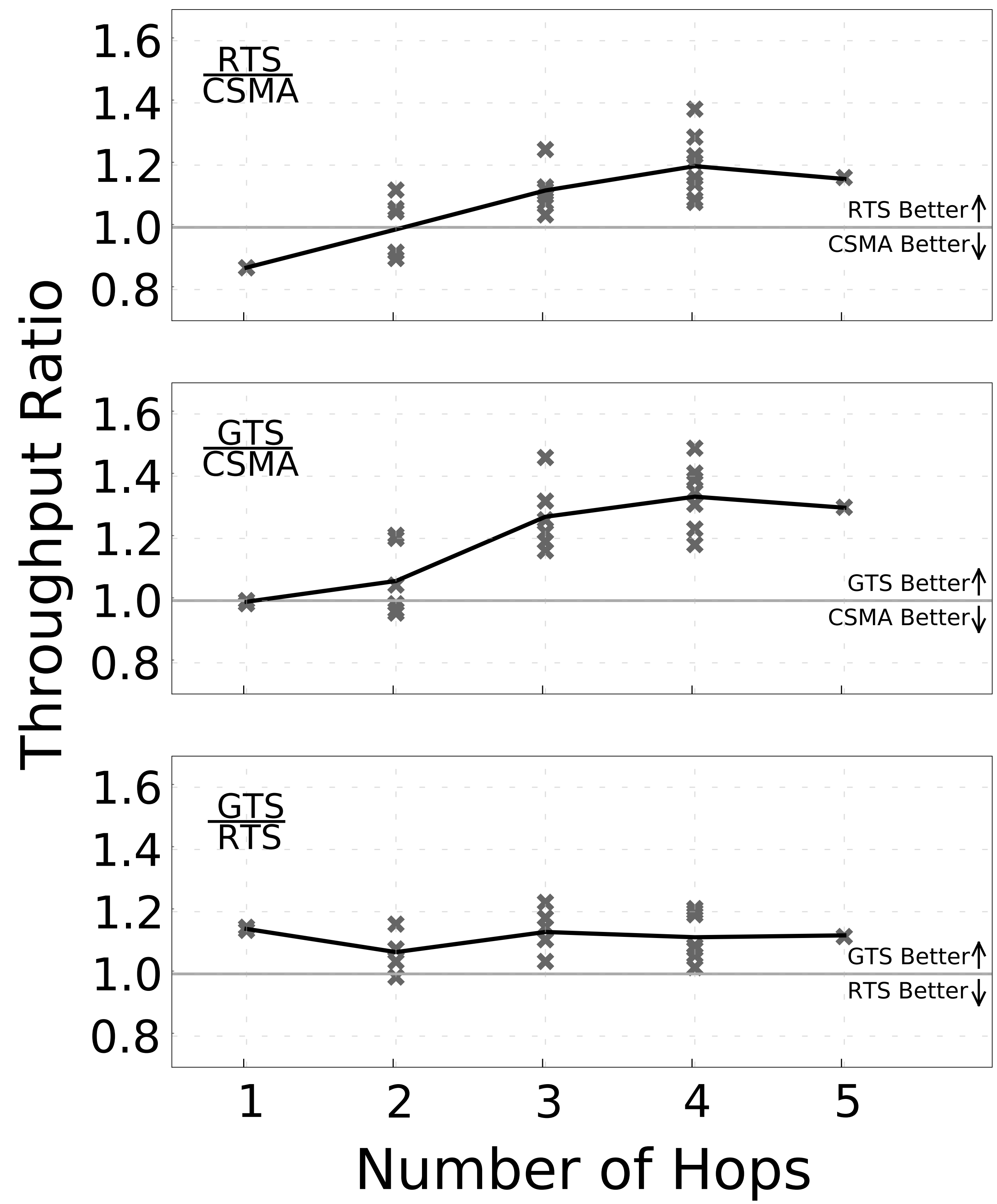
Effect of Hop Count

- 24-node large testbed
  - Spread across 6 floors in our CS building
  - 802.11 Channel 1
- iperf measures the throughput of 23 pairs



Effect of Hop Count

- Shorter paths ➜ fewer collisions
  - CSMA outperforms RTS/CTS due to no overhead
  - GTS matches CSMA's performance
- Longer paths ➜ more collisions
  - RTS/CTS outperforms CSMA due to better collision avoidance
  - GTS outperforms both RTS/CTS and CSMA
- GTS matches/outperforms both in any case

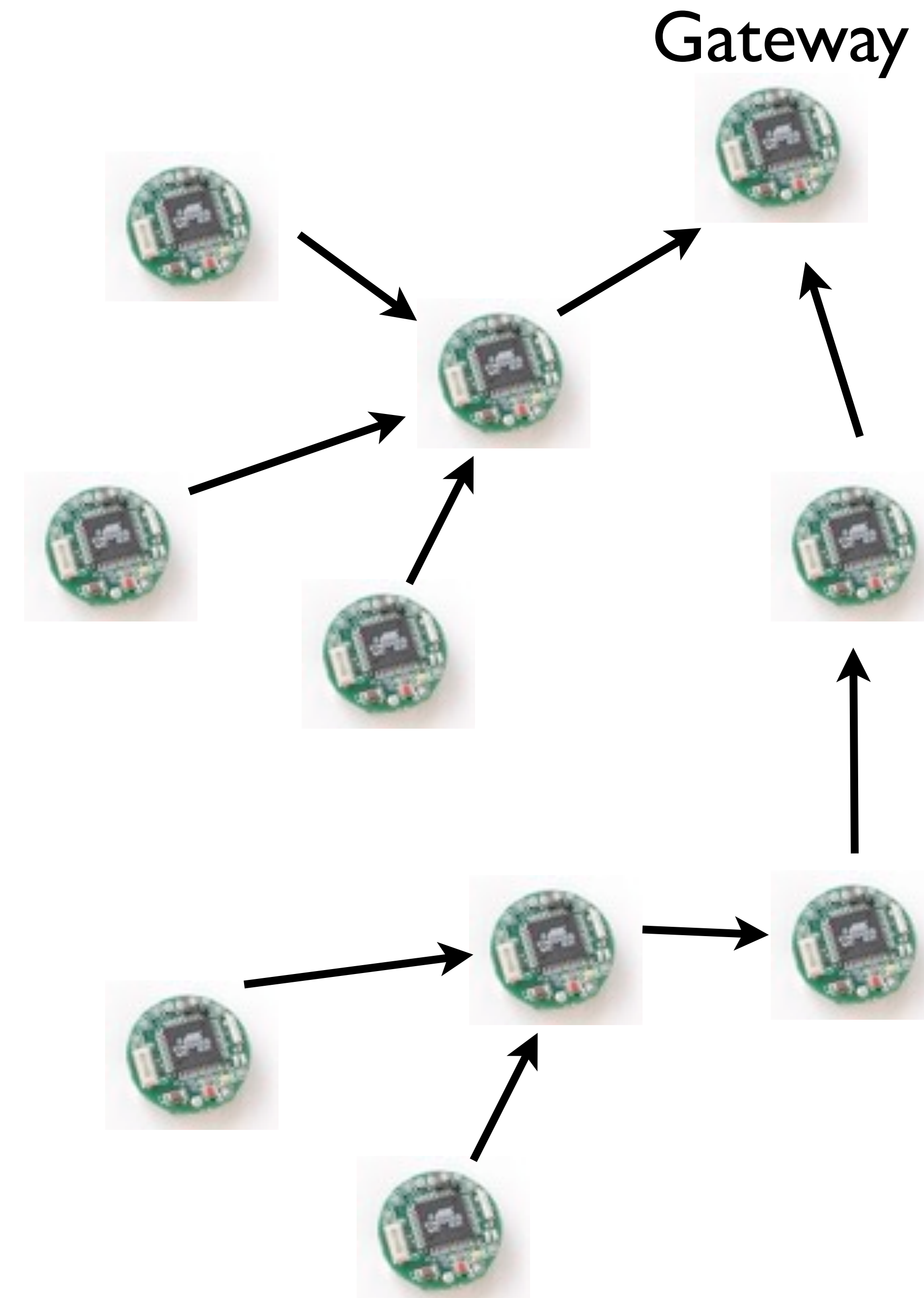


Talk Outline

- Grant-To-Send Mechanism
- Optimal Grant Duration
- GTS in 802.11 : UDP
- GTS in 802.15.4 : CTP and Deluge
- Limitations of GTS

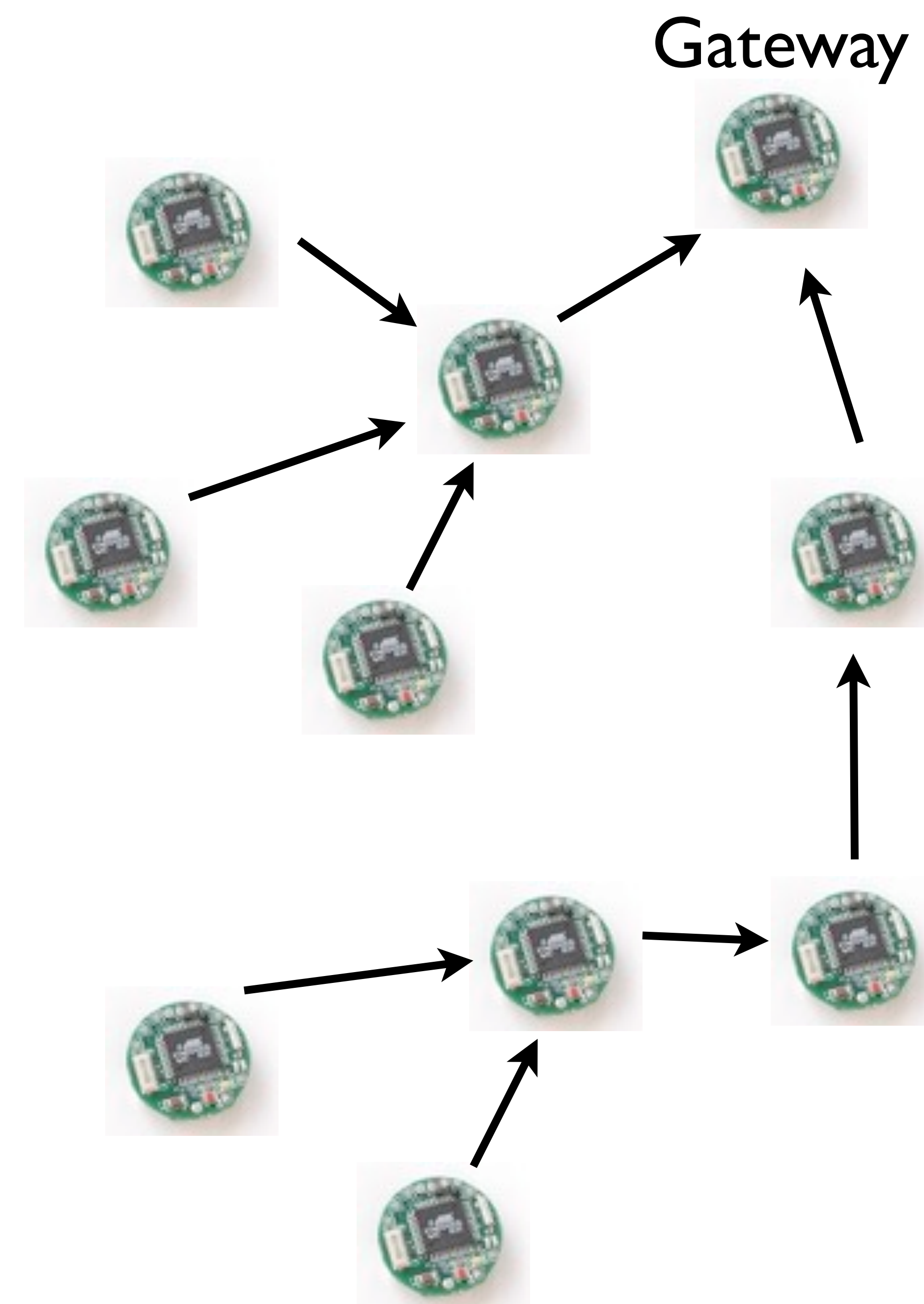
Collection Tree Protocol

- Collects sensor data to gateway by constructing a minimum-cost tree
- Multiple converging UDP-like flows: susceptible to intra-flow collisions



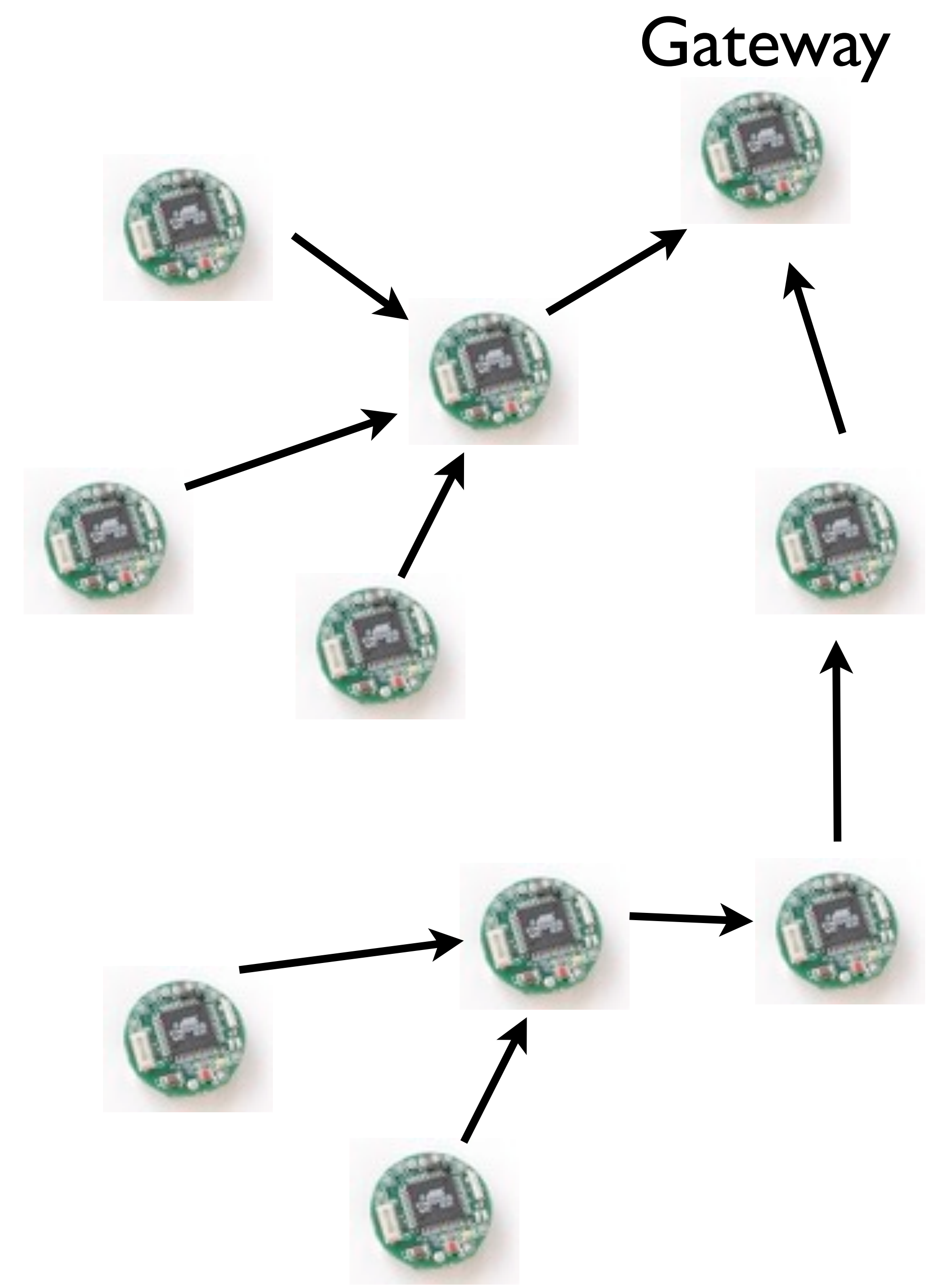
Collection Tree Protocol

- Collects sensor data to gateway by constructing a minimum-cost tree
- Multiple converging UDP-like flows: susceptible to intra-flow collisions
- Has built-in collision avoidance mechanism
  - Delays back-to-back transmission by ~2 pkt times



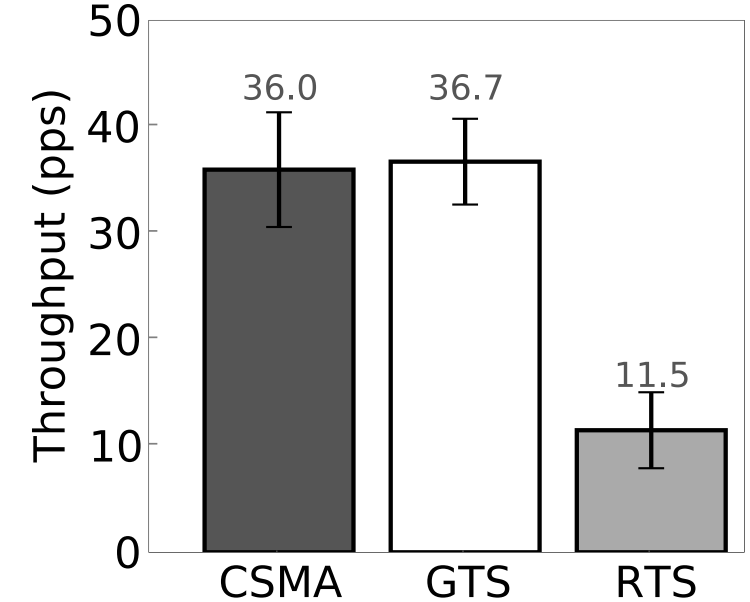
Collection Tree Protocol

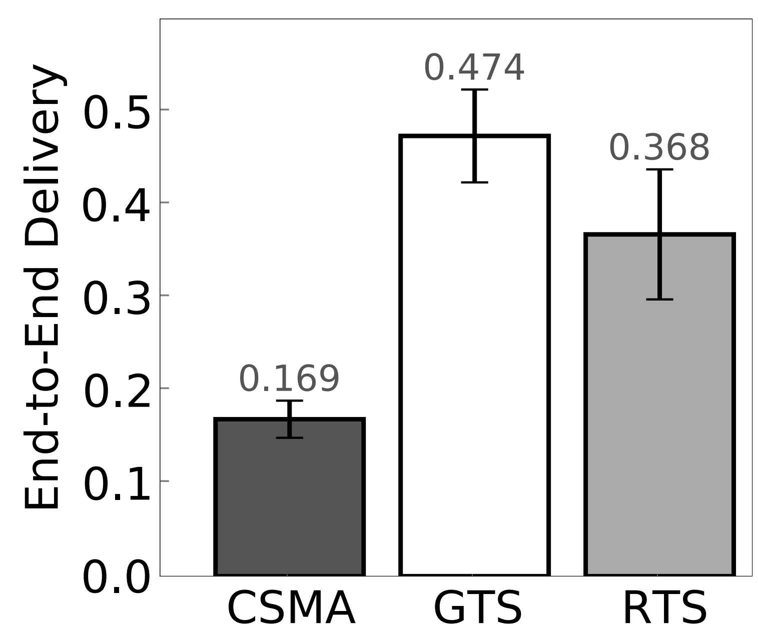
- Collects sensor data to gateway by constructing a minimum-cost tree
- Multiple converging UDP-like flows: susceptible to intra-flow collisions
- Has built-in collision avoidance mechanism
    - Delays back-to-back transmission by ~2 pkt times
- GTS can substitute the layer 3 mechanism



Evaluation on CTP

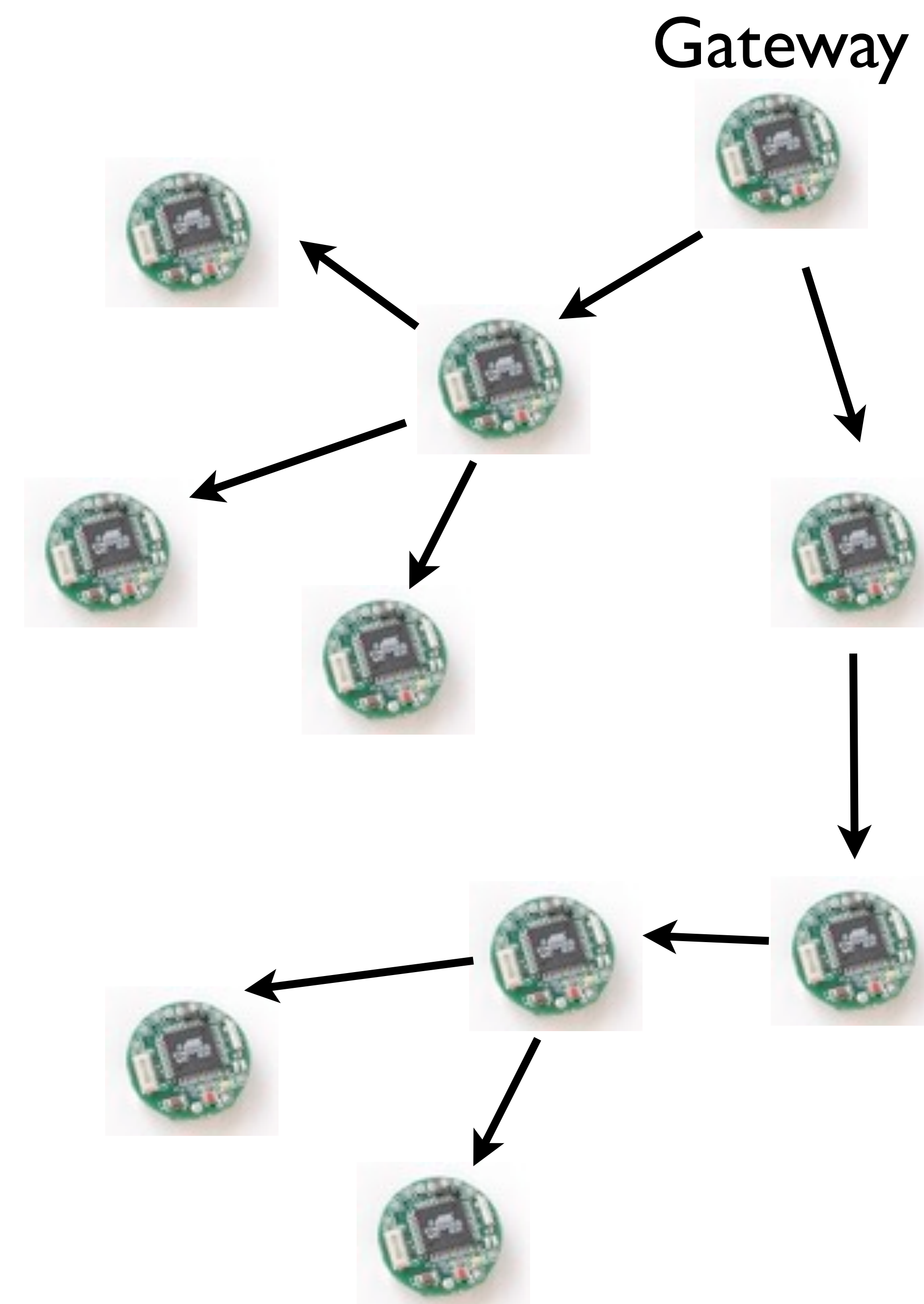
- 64-node Mirage testbed
- Event-triggered collection scenario
- GTS maintains the throughput while improving end-to-end delivery
- GTS provides the natural per-region rate limitation





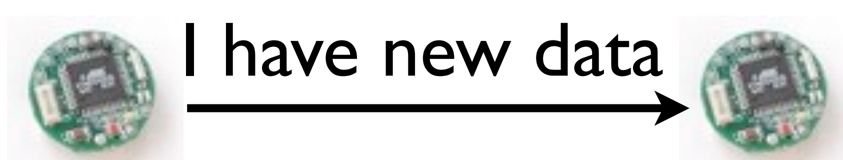
Dissemination: Deluge

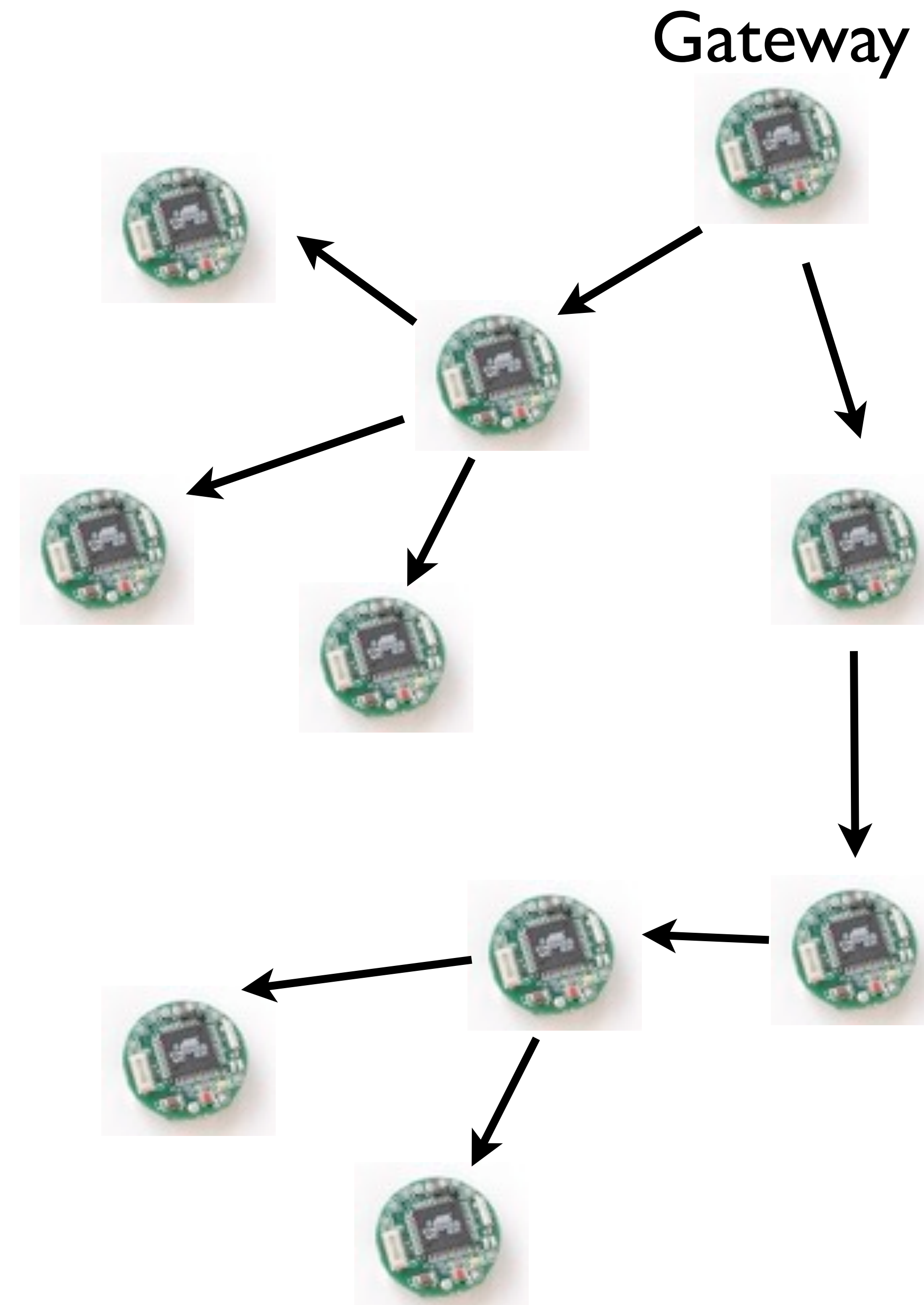
- Distributes a large piece of data from a Gateway or source to each node in network
    - eg: distributing new binary



Dissemination: Deluge

- Distributes a large piece of data from a Gateway or source to each node in network
  - eg: distributing new binary
- Steps:
  - Advertisement

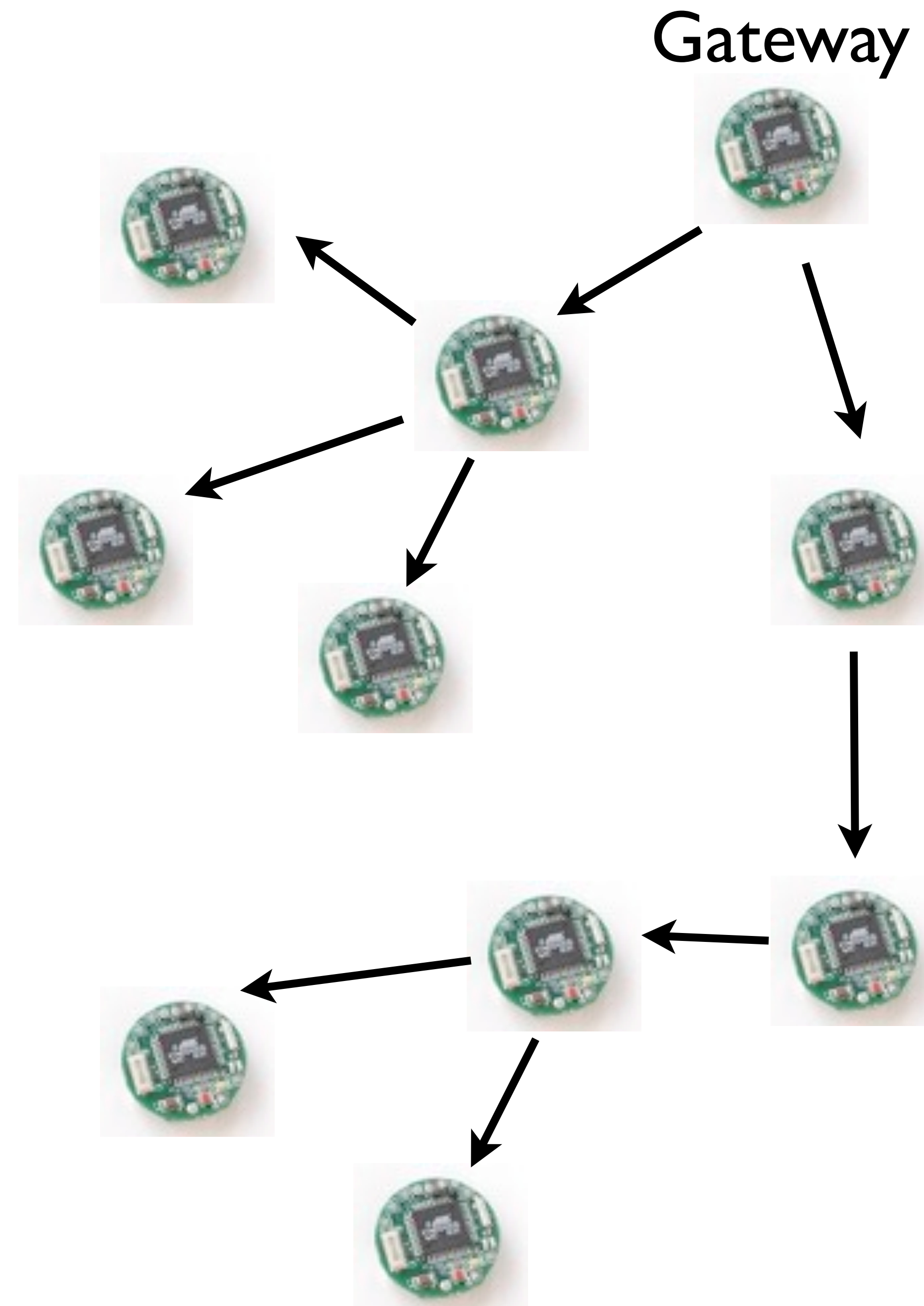




Dissemination: Deluge

- Distributes a large piece of data from a Gateway or source to each node in network
  - eg: distributing new binary
- Steps:
  - Advertisement

  - Request

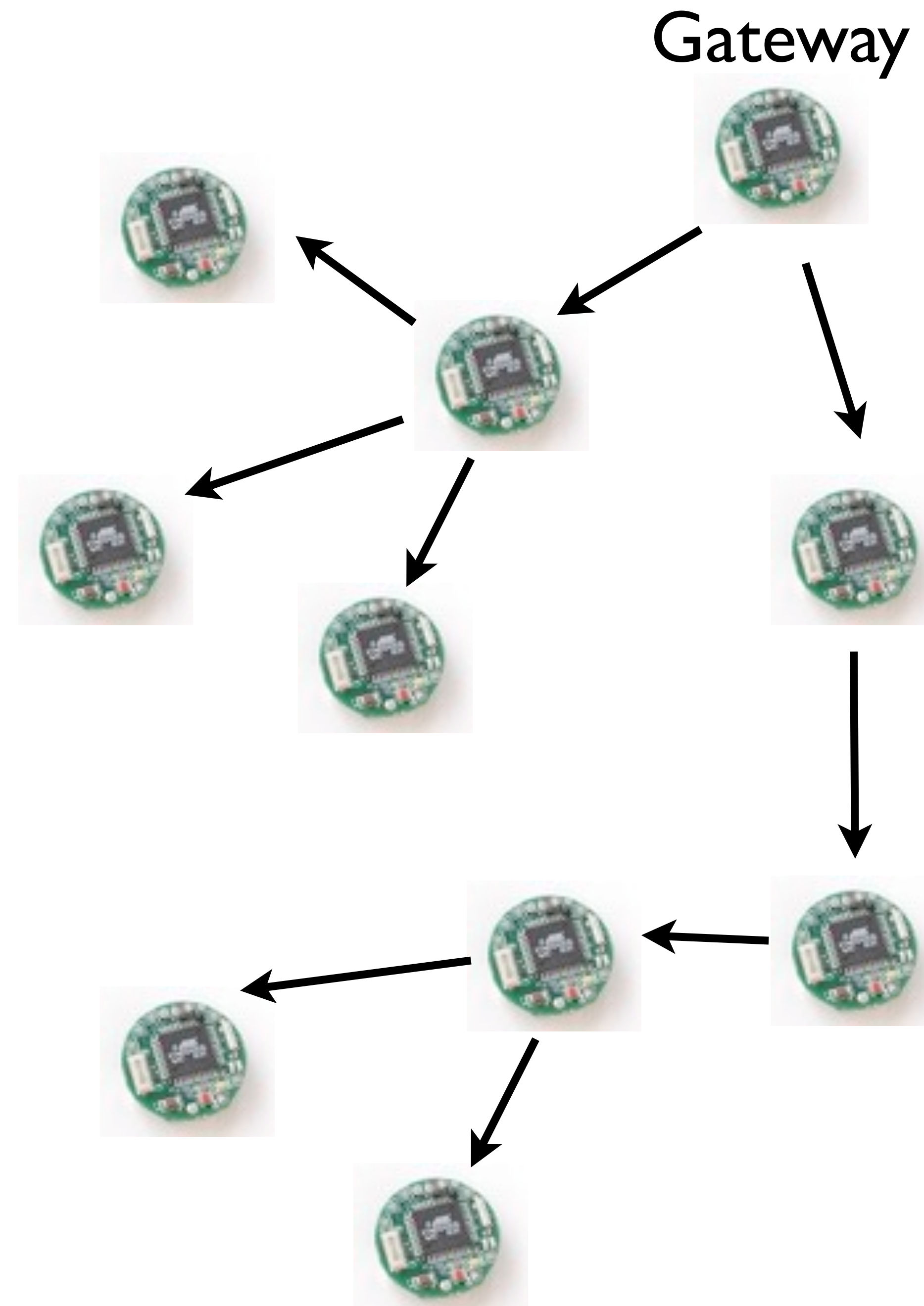


Dissemination: Deluge

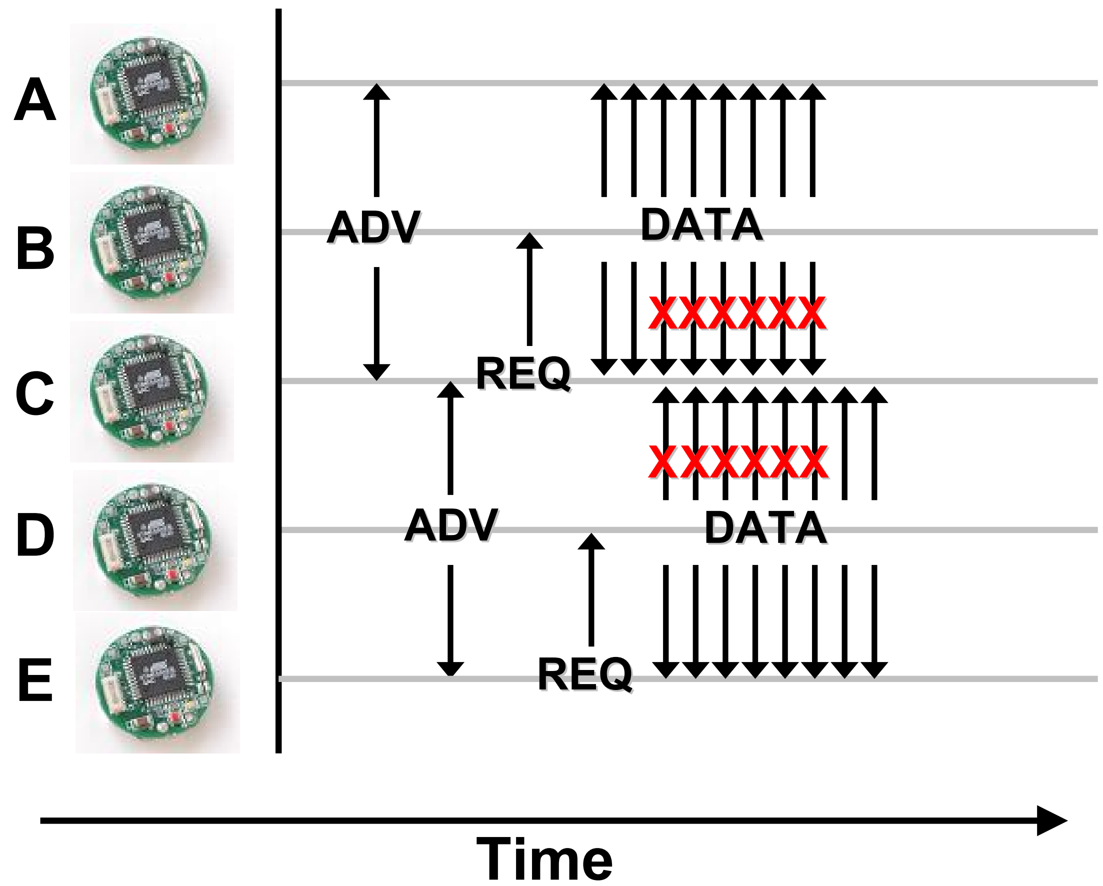
- Distributes a large piece of data from a Gateway or source to each node in network
  - eg: distributing new binary
- Steps:
  - Advertisement

  - Request

  - Dissemination



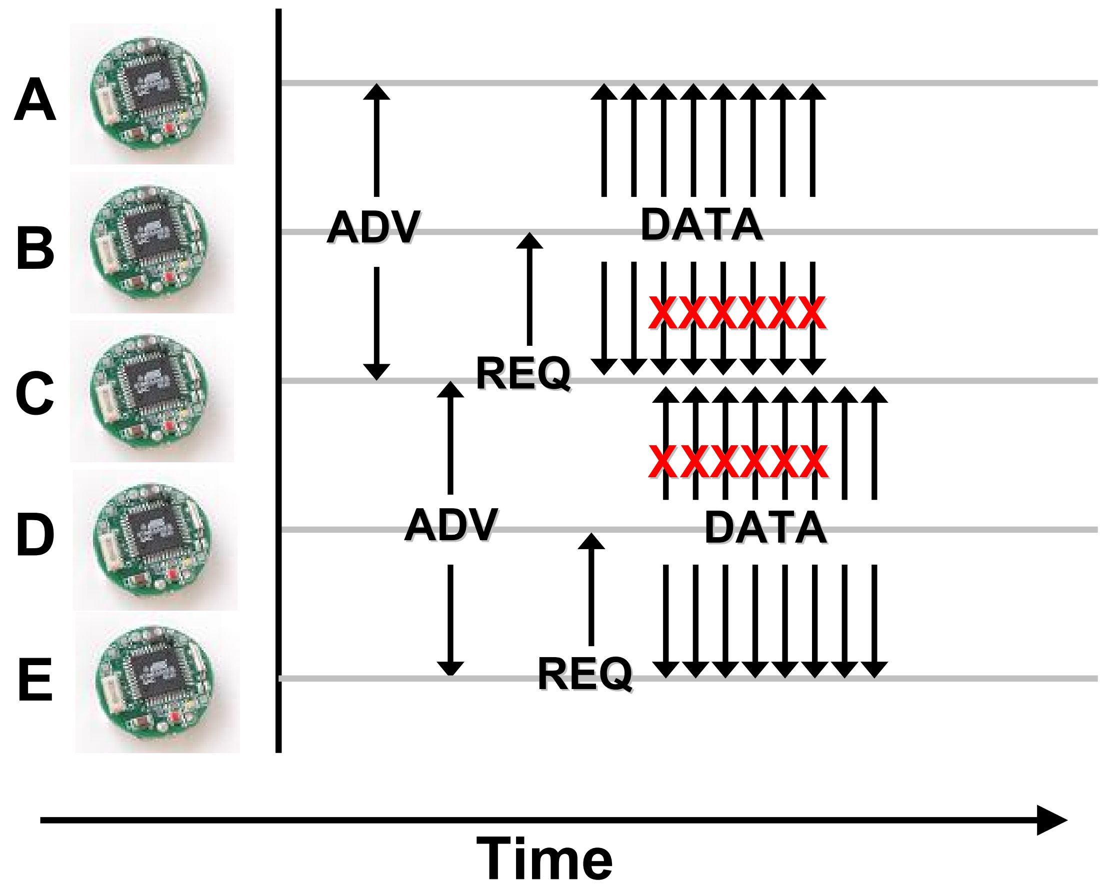
Deluge and GTS



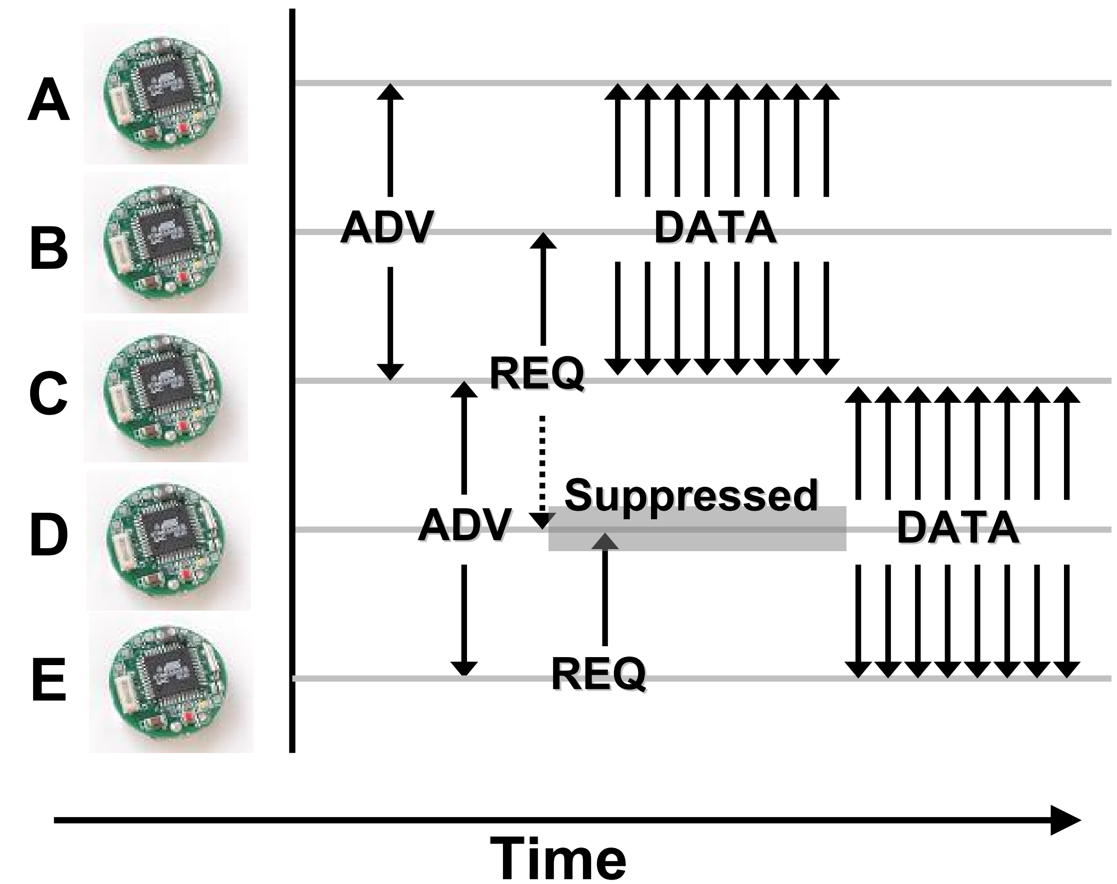
Plain Deluge

- Deluge requests can lead to a flurry of losses due to hidden terminal

Deluge and GTS



Plain Deluge



GTS-Deluge

- Deluge requests can lead to a flurry of losses due to hidden terminal

- GTS: Embed grant for the whole data in requesting packets

- A non-forwarding example of GTS

Talk Outline

- Grant-To-Send Mechanism
- Optimal Grant Duration
- GTS in 802.11 : UDP
- GTS in 802.15.4 : CTP and Deluge
- Limitations of GTS

Imperfect Grants

- Granter must guess how long the channel will be used by the grantee
- Not obvious: variable bit-rate, different packet sizes, retransmissions
- Can be estimated: e.g. nodes can learn the bit-rate used at the next hop
- Small grants are better than no grant

Inter-Flow Collisions

- GTS does not address inter-flow collisions
- Can still benefit when multiple flows are in the same direction (e.g. CTP)
- Generally hard to address with link-layer mechanisms
  - 2-hop reservation incurs overhead
- (GTS + network coding) can be an answer

Is Collision a Problem?

- Collision can be recovered using various PHY-layer techniques
  - E.g. ZigZag, ANC, SIC, etc.
- Require a new hardware
- Does not mean *any* collision can be recovered
  - Hard to recover collisions with more than 2~3 concurrent packets
- Can work together with GTS

Conclusions

- A simple and inexpensive collision avoidance mechanism for wireless mesh
- Backwards-compatible with existing 802.11
  - 802.11 respects grants, GTS respects 802.11
  - Nodes talking to AP behaves like normal CSMA
- GTS outperforms CSMA and RTS/CTS without incurring overhead

Thank You!

Grant-To-Send (GTS)

- A novel collision avoidance mechanism for CSMA based wireless mesh networks
- Instead of avoiding collisions with packets to be sent, GTS avoids collisions with packets the node expects to hear
  - Eg: Grant forwarding node channel access to forward data packet out of interference range
- No control packets. 0-2 bytes overhead in data packets
- Simple and general
  - 802.11: completely backwards compatible with 11 lines of driver code change w/ existing hardware
  - 802.15.4: 50 lines of TinyOS codes with 9B RAM

Backup: TCP Performance

- Similar performance gain for GTS as UDP
- RTS/CTS shows poor performance
  - Larger overhead for short packets
- GTS and CSMA achieves only ~2/3 of the UDP throughput

Hops	**# Pairs**	**GTS**	**CSMA**	**RTS/CTS**
1	2	2.25	2.21 (2%)	1.91 (18%)
2	6	0.77	0.72 (7%)	0.61 (26%)
3	6	0.51	0.44 (16%)	0.24 (113%)
4	8	0.46	0.31 (48%)	0.18 (156%)
5	1	0.50	0.39 (28%)	0.28 (79%)
Total	23	0.71	0.62 (15%)	0.46 (53%)

Backup: Sending Fewer Acks

- TCP performance bottleneck in wireless mesh may be DATA-ACK collisions
  - Filtering ACK packets gives higher GTS performance