Minimizing Energy Consumption in Sensor Networks Using a Wakeup Radio

Matthew J. Miller and Nitin H. Vaidya

IEEE WCNC
March 25, 2004

Problem Statement

- Sensor networks with limited resources
  - Energy
  - Queue size for packets
- Address the power save problem
  - When should a node switch its radio to the sleep state and for how long?

Motivation

- Sleep mode power consumption is much less than idle power consumption
- Sensors have limited queue size
  - Packets may be dropped if wakeups are too far apart

Radio State	Power Consumption (mW)
Transmit	81
Receive/Idle	30
Sleep	0.003

Power Characteristics for a Mica2 Mote Sensor

Power Save Design Alternatives

- Timer-Based
  - When a node enters sleep mode, it sets a timer to wakeup at a pre-determined time
- On-Demand
  - A sleeping node can be woken at any time via out-of-band communication
- Hybrid
  - Timer-Based plus On-Demand

Wakeup Radio

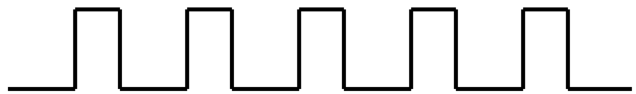
- Add second, low-power radio to wakeup neighbors on-demand
- Low-power could be achieved by:
  - Simpler hardware with a lower bit-rate and/or less decoding capability
  - Periodic listening using a radio with identical physical layer as data radio (e.g., STEM)
    - Used in this work

Directed vs. Broadcast Wakeups

- Directed
  - Encode ID of node to be woken in the wakeup signal
- Broadcast
  - Wakeup signal awakes entire neighborhood (e.g., busy tone)
  - Only have to detect energy on channel rather than decode packet
    - Simple hardware
    - Small detection time

Sleeping Protocol

- Sense wakeup channel periodically



- If wakeup channel sensed busy:
  - Turn on data radio
  - Receive *filter* packet on data channel
  - If filter is for another node, return to sleep

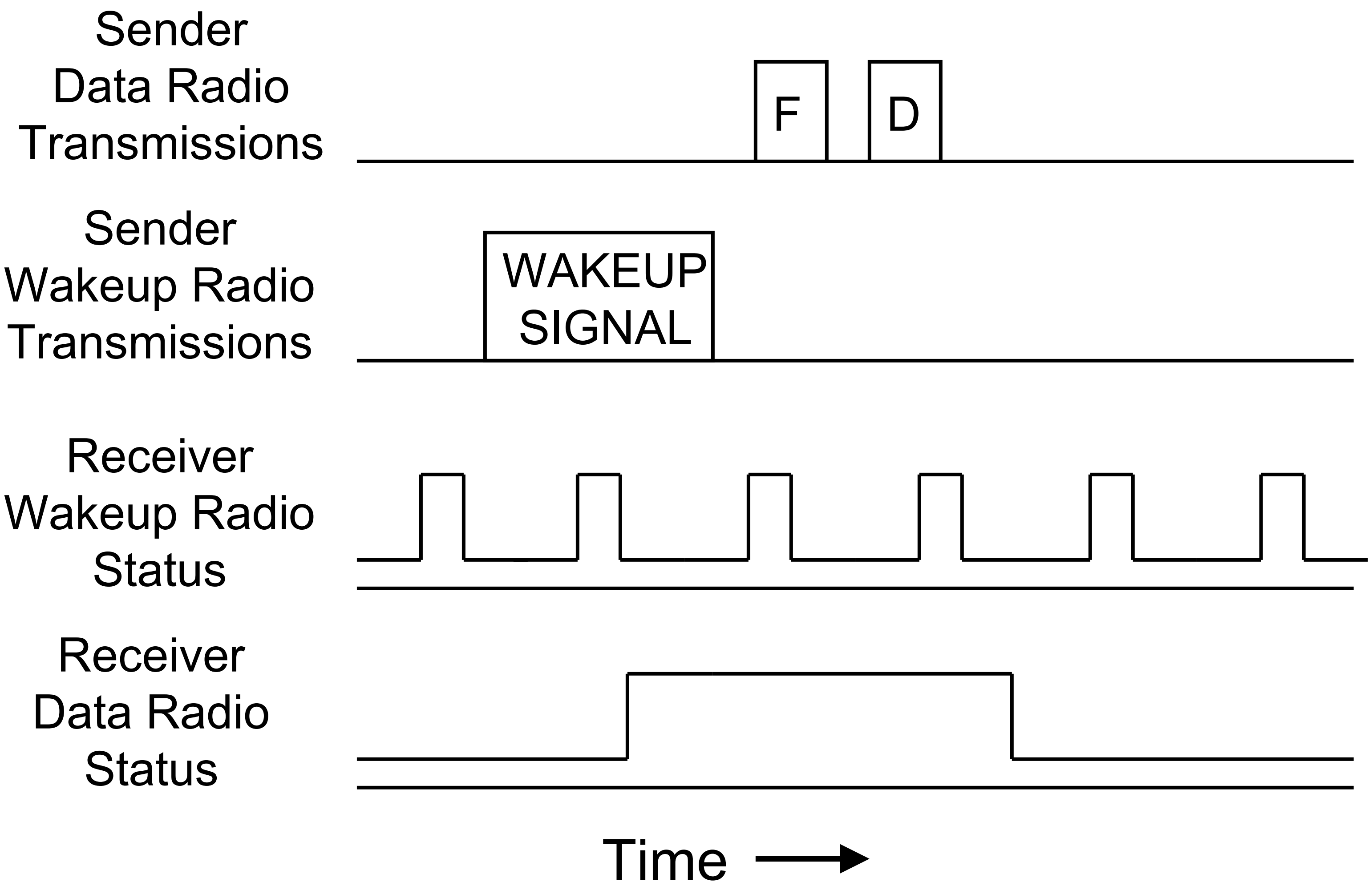
- Filter is like RTS, but can specify multiple receivers

Sending Protocol

- Transmit wakeup signal long enough for all neighbors to hear it
- Transmit filter packets specifying intended receiver(s)
- Transmit data to receiver

- Entire neighborhood wakes up long enough to receive filter
  - Large energy cost
  - Referred to as a *full wakeup*

Full Wakeup Example



Timeout Triggered Wakeups

- Nodes do a timer-based wakeup on the data radio
  - Referred to as *timeout triggered wakeup*
- If the node cannot wait until the timer expires, it does a full wakeup
  - Do a full wakeup when a specified queue threshold (*L*) is reached

- Main contribution: adding timeout triggered wakeups in addition to full wakeups

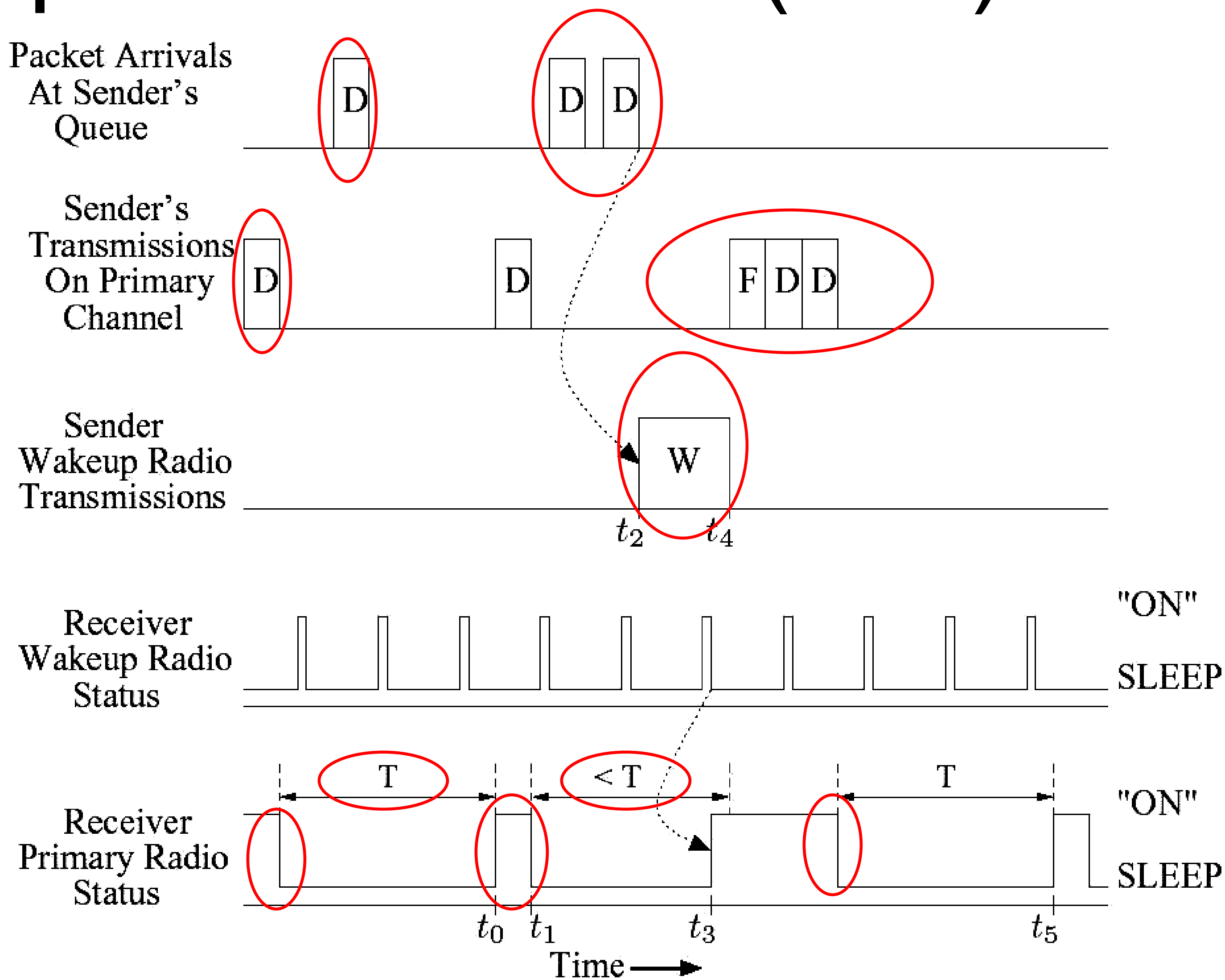
Timeout Triggered Wakeups (cont.)

- Timeout computed based on recent traffic rate
  - Packets interarrival times have exponential distribution
- Sender will compute timeout value and piggyback on data packets
  - No absolute synchronization required

Timeout Tradeoff

- Too small
  - Nodes wakeup when there are no pending packets
- Too large
  - Full wakeups are more likely to occur

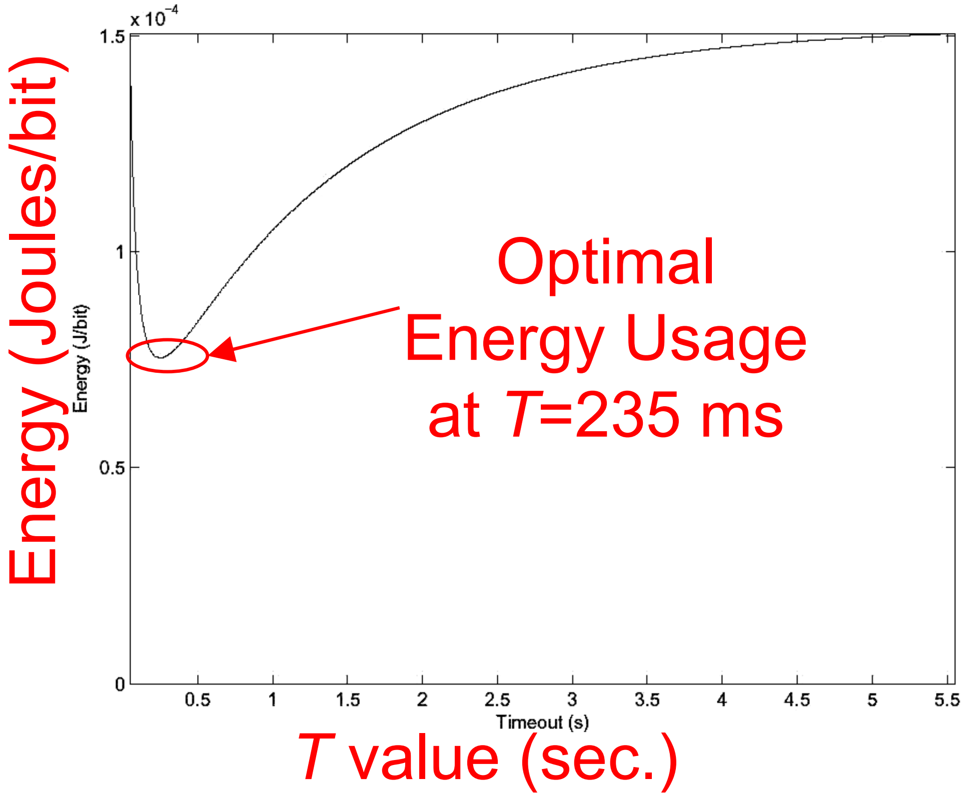
Proposed Protocol (L=2)



Analysis

- **Goal**: Find *T* value that minimizes the energy per bit.

- One sender and one receiver
- Single hop network
- *N*=8, *L*=2, *R*=1.0



Analysis (cont.)

- Based on analysis, we observe that the optimal *T* value (T_{opt}) is:

$$T_{opt} = \gamma \frac{L}{R}$$

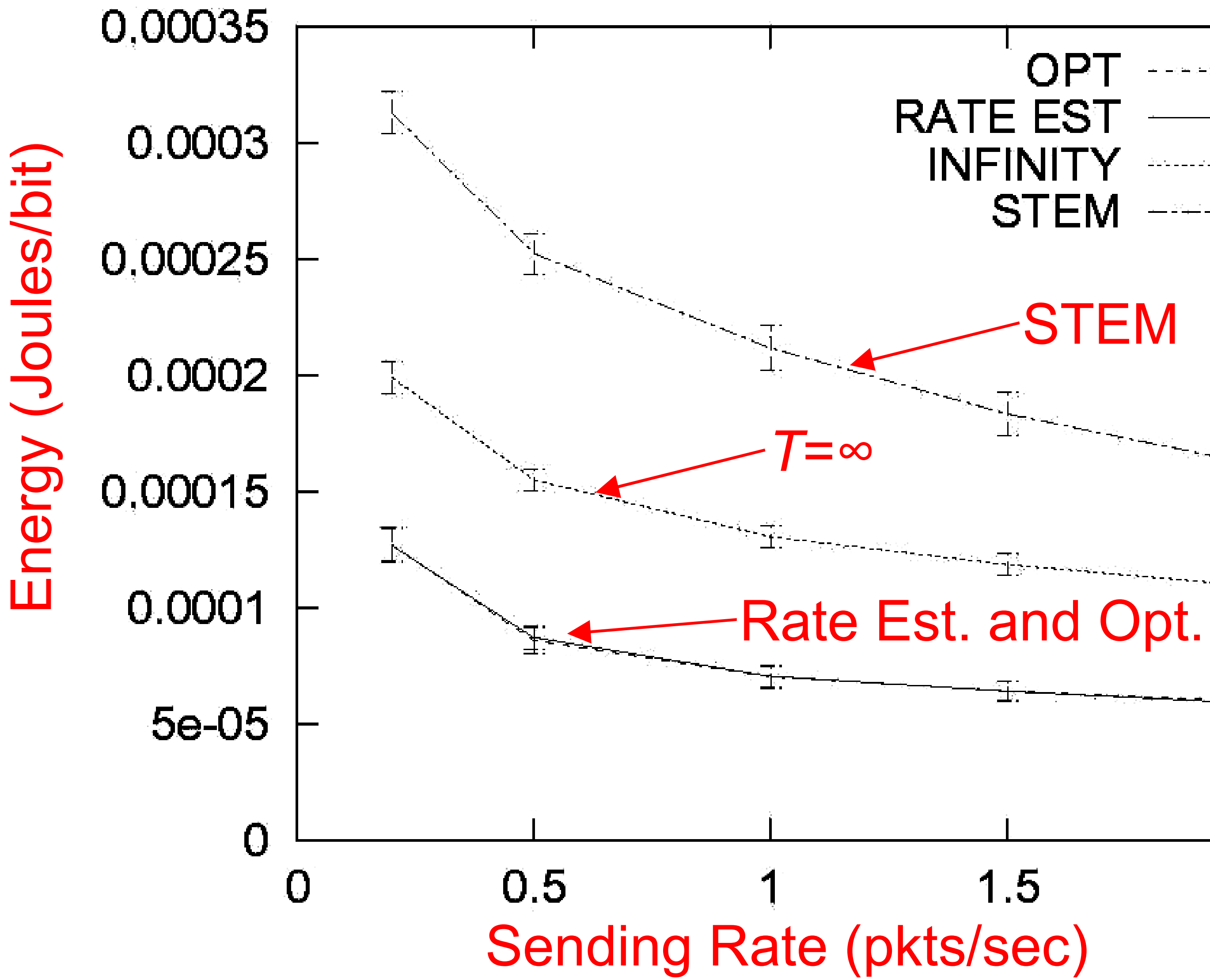
  where γ is a function of *N* and *L*.

- Compute γ offline, given *N* and *L*, and estimate rate based on weighted average of packet interarrival times

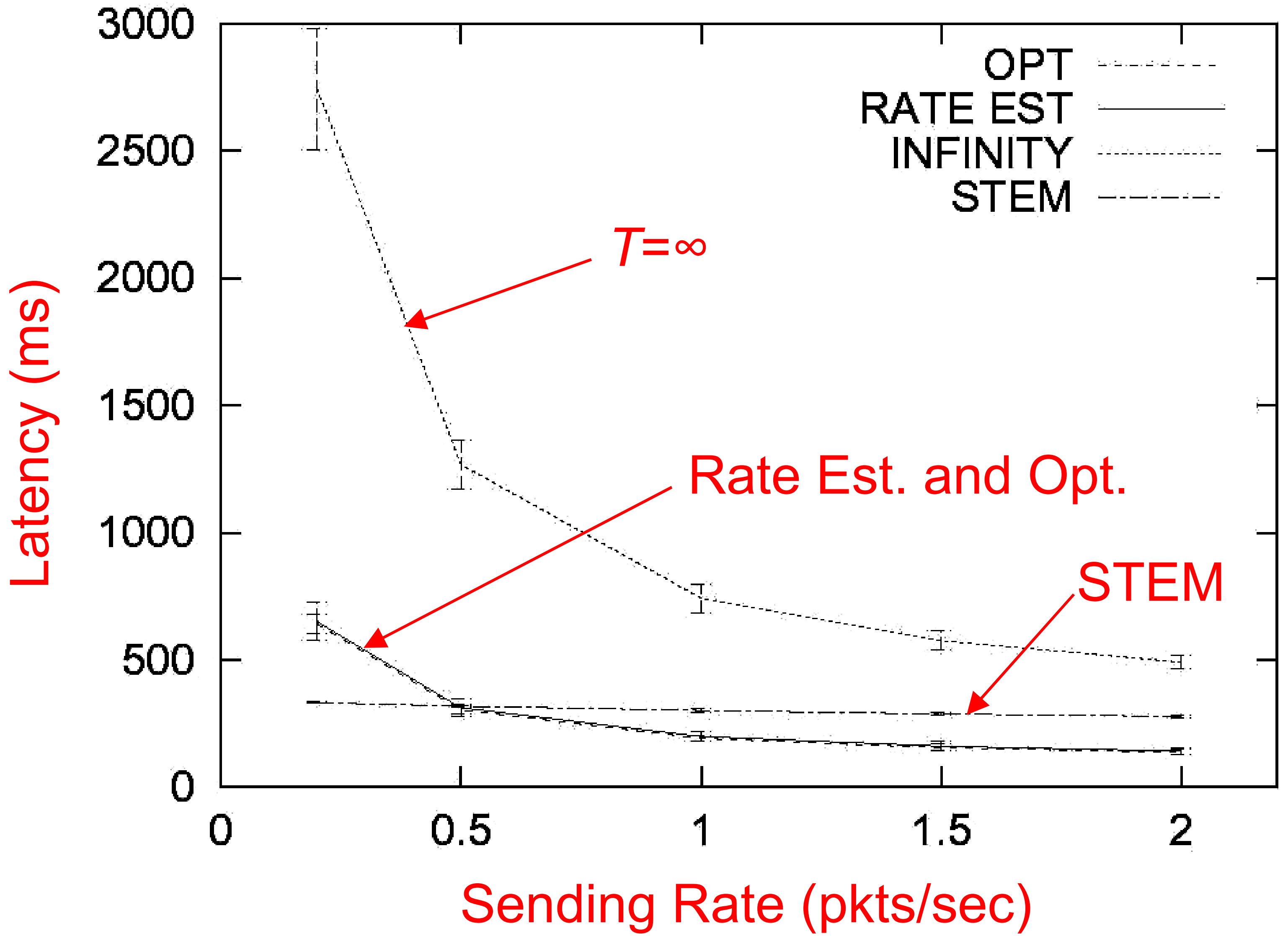
Protocols Tested

- Rate Estimation
  - Proposed protocol. γ is input for L=2 and N=8.
- Static Optimal
  - Static value of T which minimizes energy is input
- $T=\infty$
  - No timeout triggered wakeups. Full wakeups occur when L=2 packets are in the queue.
- STEM
  - Protocol proposed in [Schurgers02Optimizing]. Special case of our protocol with $T=\infty$ and L=1.

Energy Usage



Latency

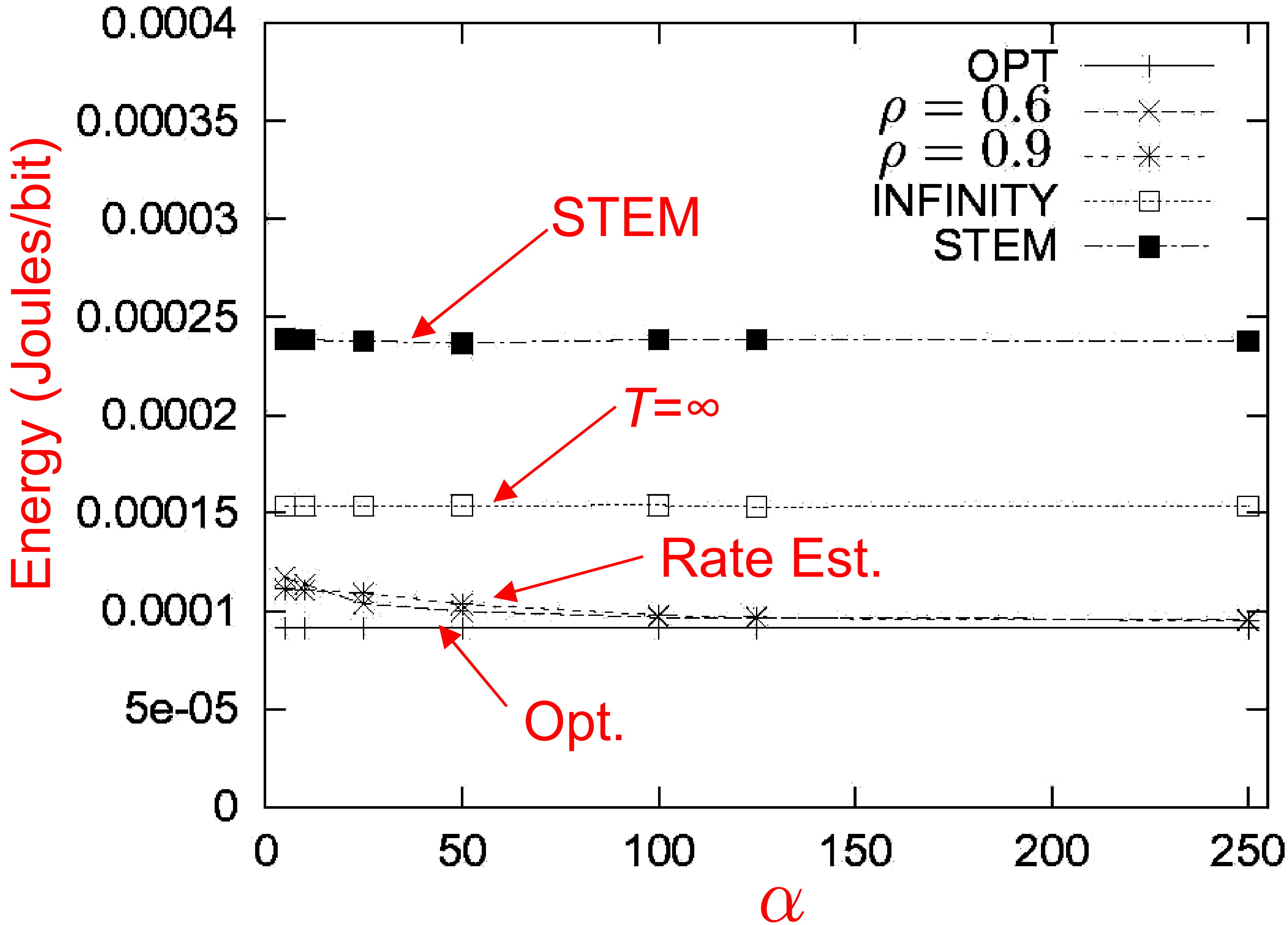

3000
2500
2000
1500
1000
500
0
0
0.5
1
1.5
2
OPT
RATE EST
INFINITY
STEM
T=∞
Rate Est. and Opt.
STEM
Latency (ms)
Sending Rate (pkts/sec)

Time-Variant Traffic

- Rate periodically switches between 0.2 and 2.0

- The α parameter represents how frequently rates are switched
  - Smaller α means more frequent switches

Time-Variant Traffic



Conclusion and Future Work

- Protocol dynamically adjusts timeout based on traffic to minimize energy consumption
  - Performs very close to the static optimal
  - Performs better than protocols which do not use timeout triggered wakeups
- Future work
  - Adapt for multihop and multiflow settings with increased contention
  - Use multiple wakeup channels