The Trickle Algorithm

Analysis, Use, and Implementation

Philip Levis
Computer Systems Lab
Stanford University

Trickle Summary

- An algorithm for establishing eventual consistency in a wireless network
- Establishes consistency quickly
- Imposes low overhead when consistent
- Cost scales logarithmically with density
- Requires very little RAM or code
- Makes no topology assumptions

Consistency

- Powerful primitive with many uses
- Routing tree maintenance
  - Invariant: next hop has lower cost
- Network configuration
  - Invariant: all have the most recent config
- Neighbor discovery
  - Invariant: node is in all neighbor's lists

Overview

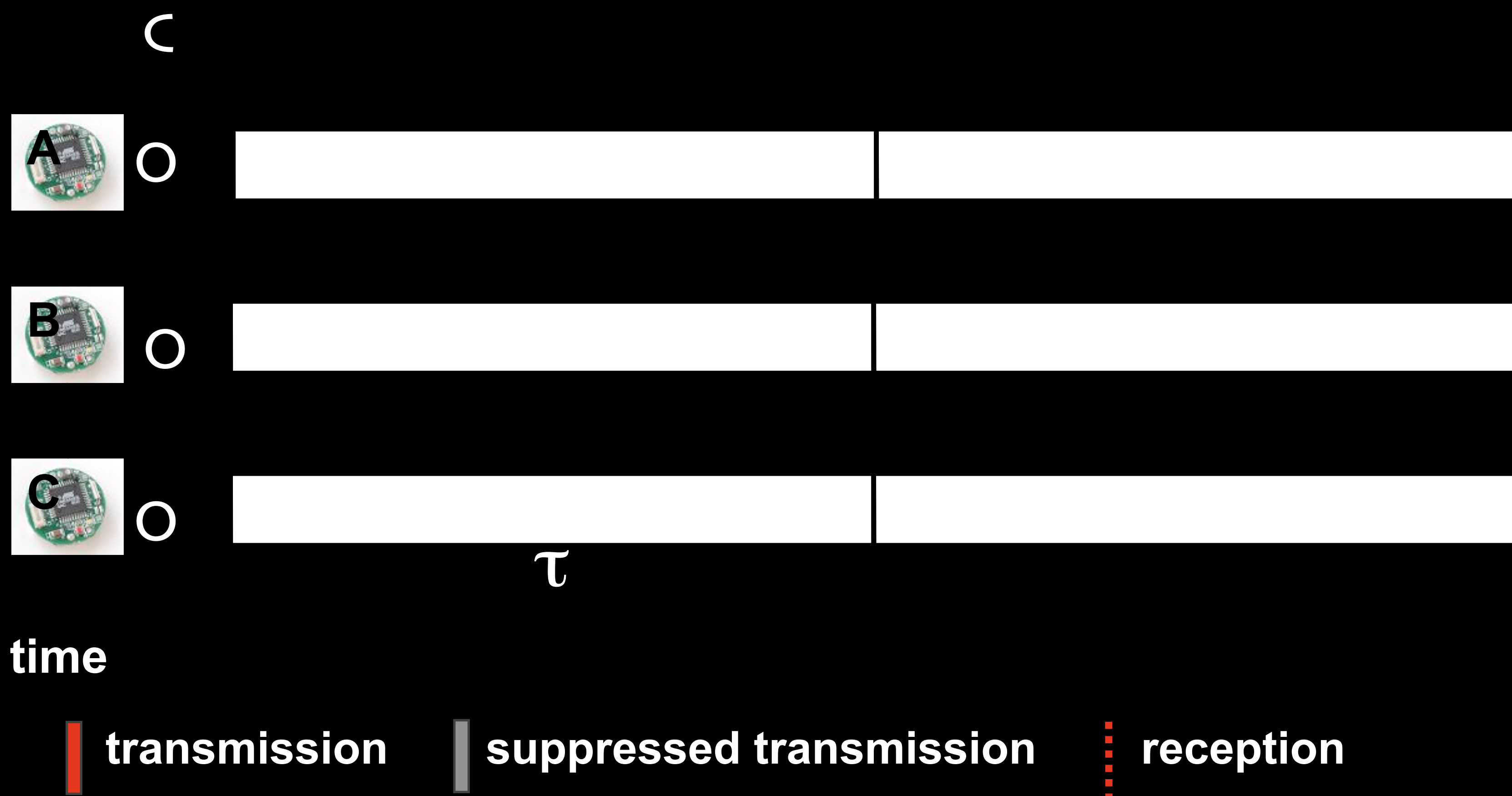
- Trickle operates over time intervals
  - No synchronization needed between nodes
- In each interval, node optionally transmits
  - Transmits if it hasn't heard transmissions that are consistent with its own
- Dynamically scales interval lengths to have fast updates yet low cost when consistent

Suppression

- Motivation: don't waste messages (energy and channel) if all nodes agrees
- Interval of length τ
  - At beginning of interval, counter c=0
  - On consistent transmission, c++
- Node picks a time t in range $[\tau/2,\tau]$
  - At t, transmit if $c < k$ (redundancy constant)

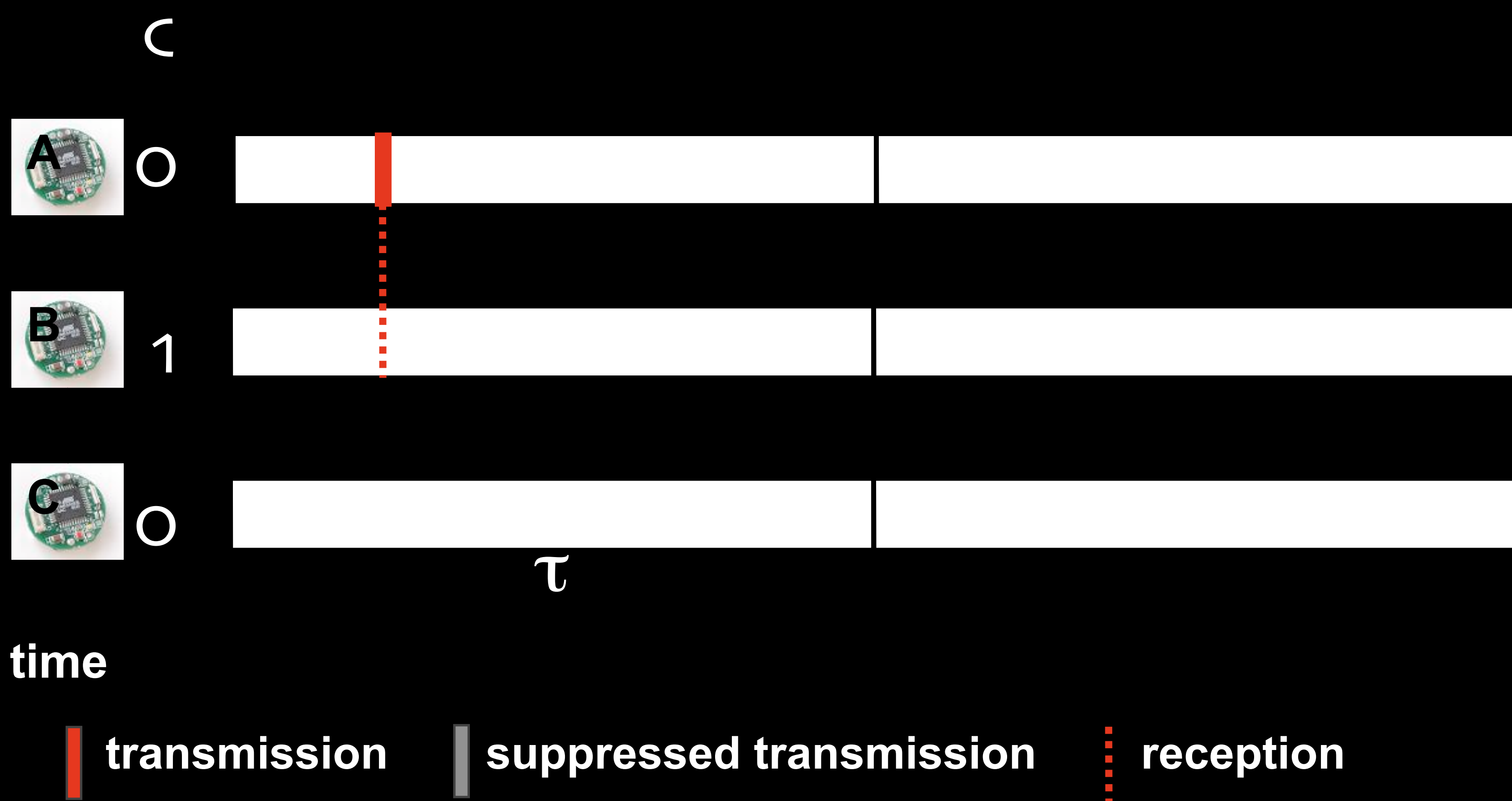
Example Execution

k=1



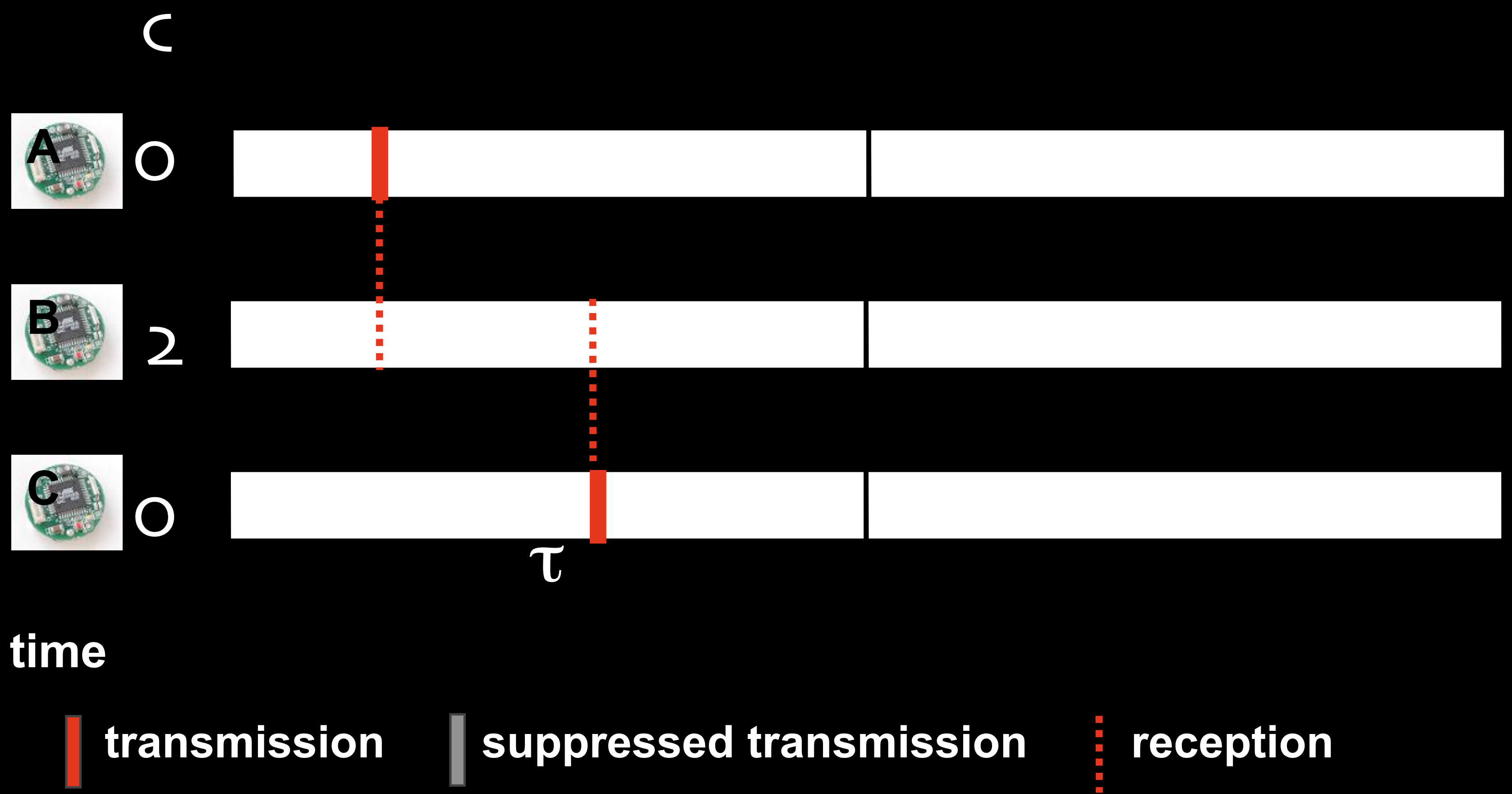
Example Execution

k=1



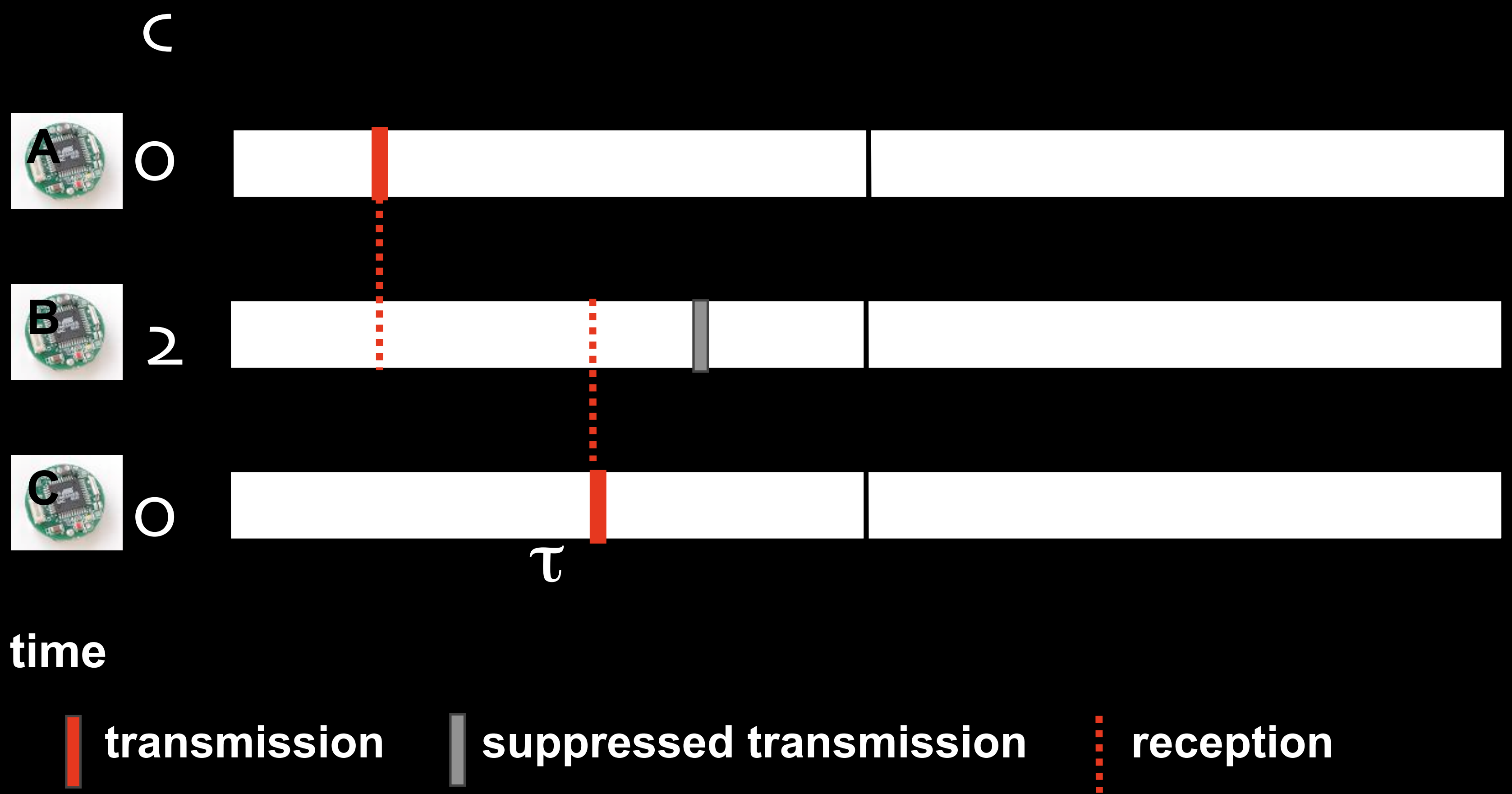
Example Execution

k=1



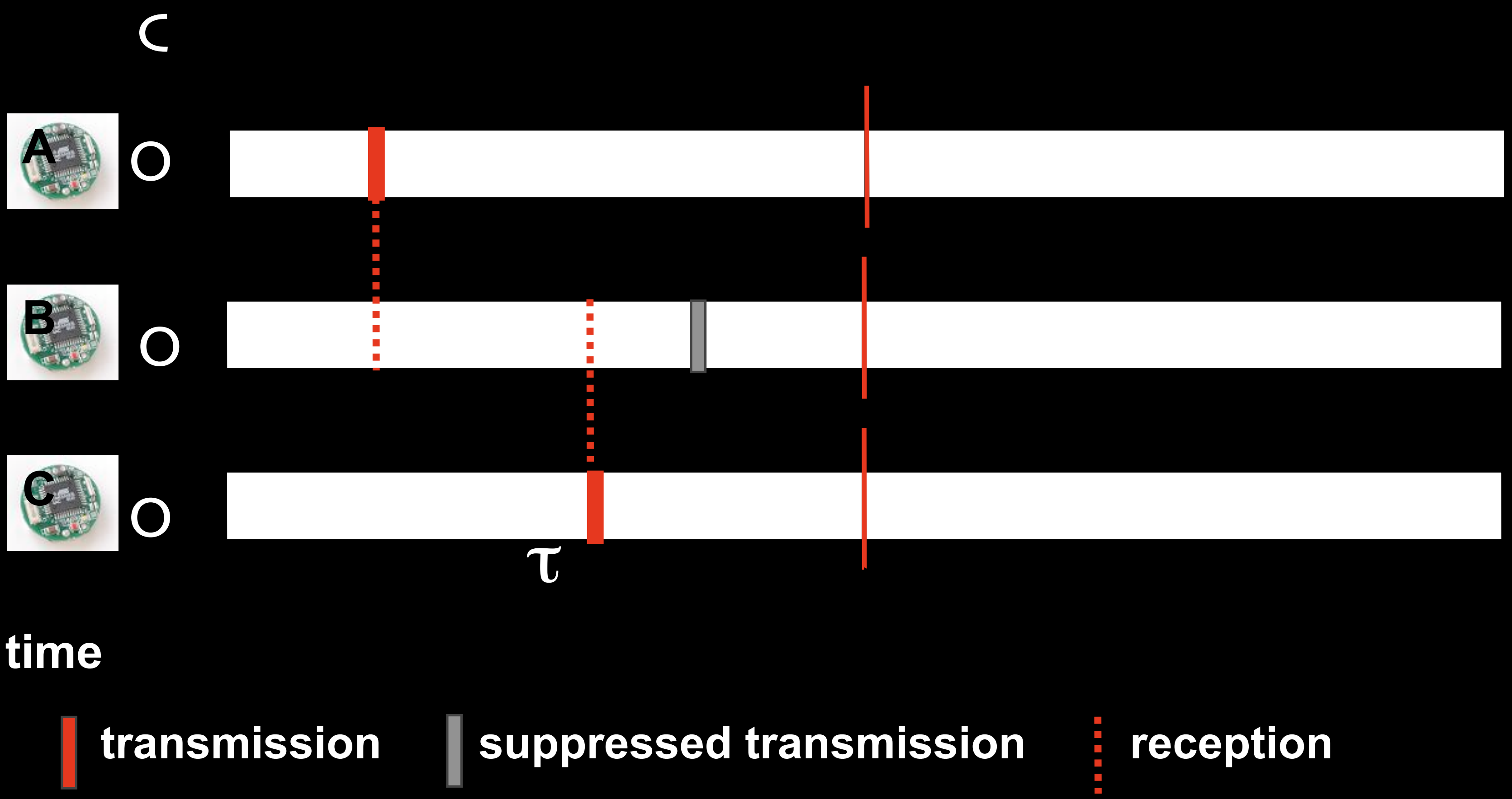
Example Execution

k=1



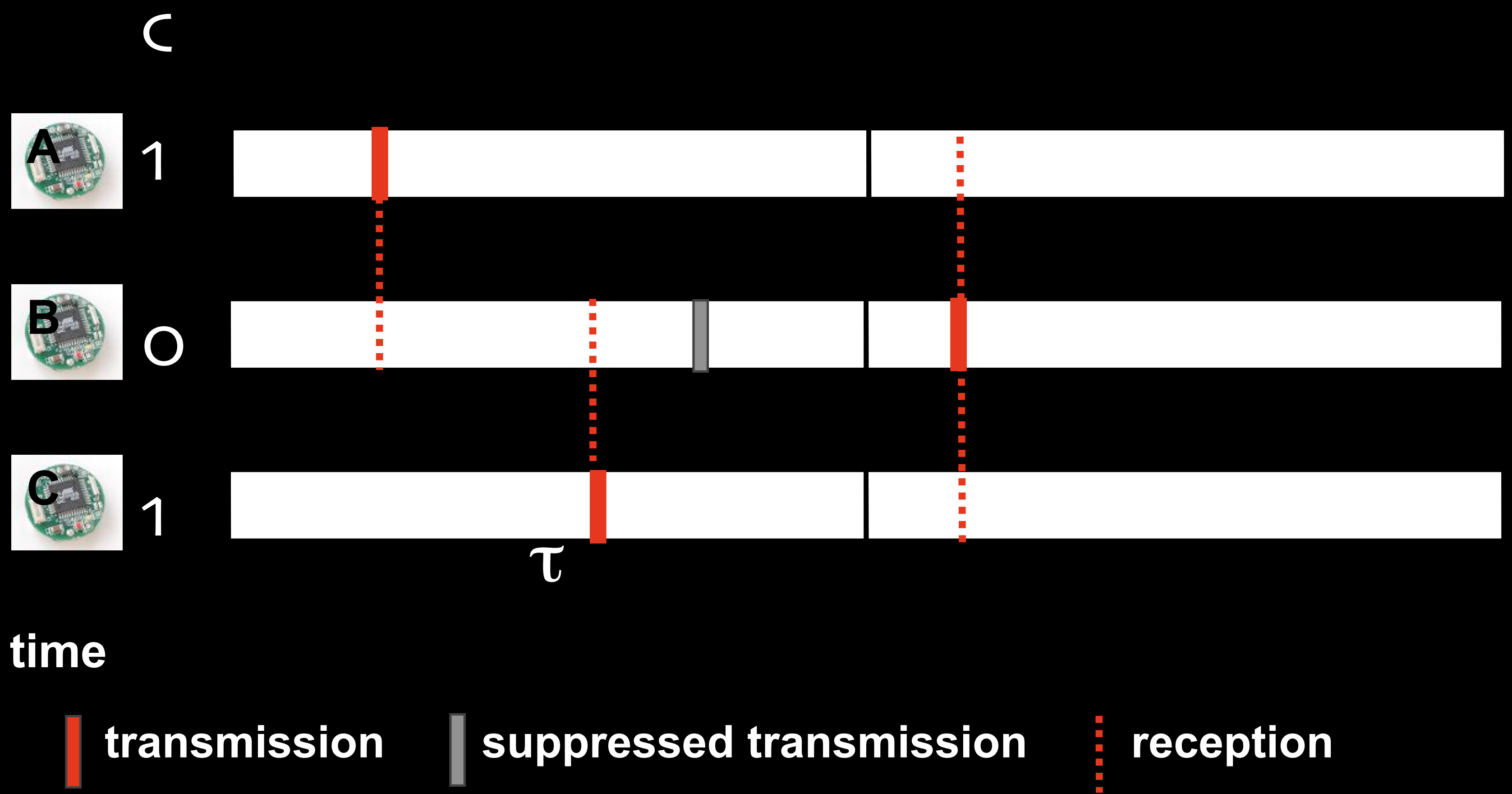
Example Execution

k=1



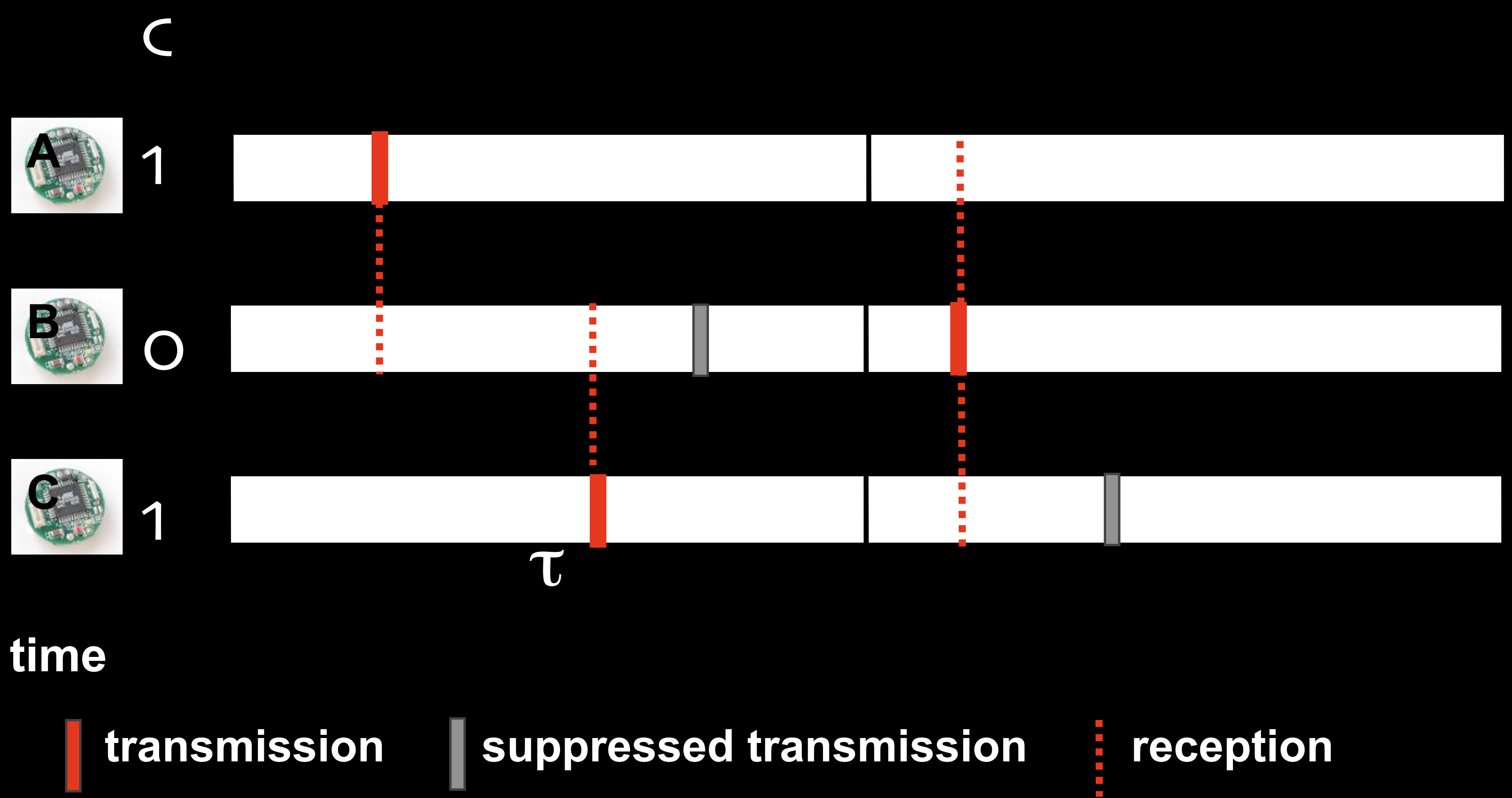
Example Execution

k=1



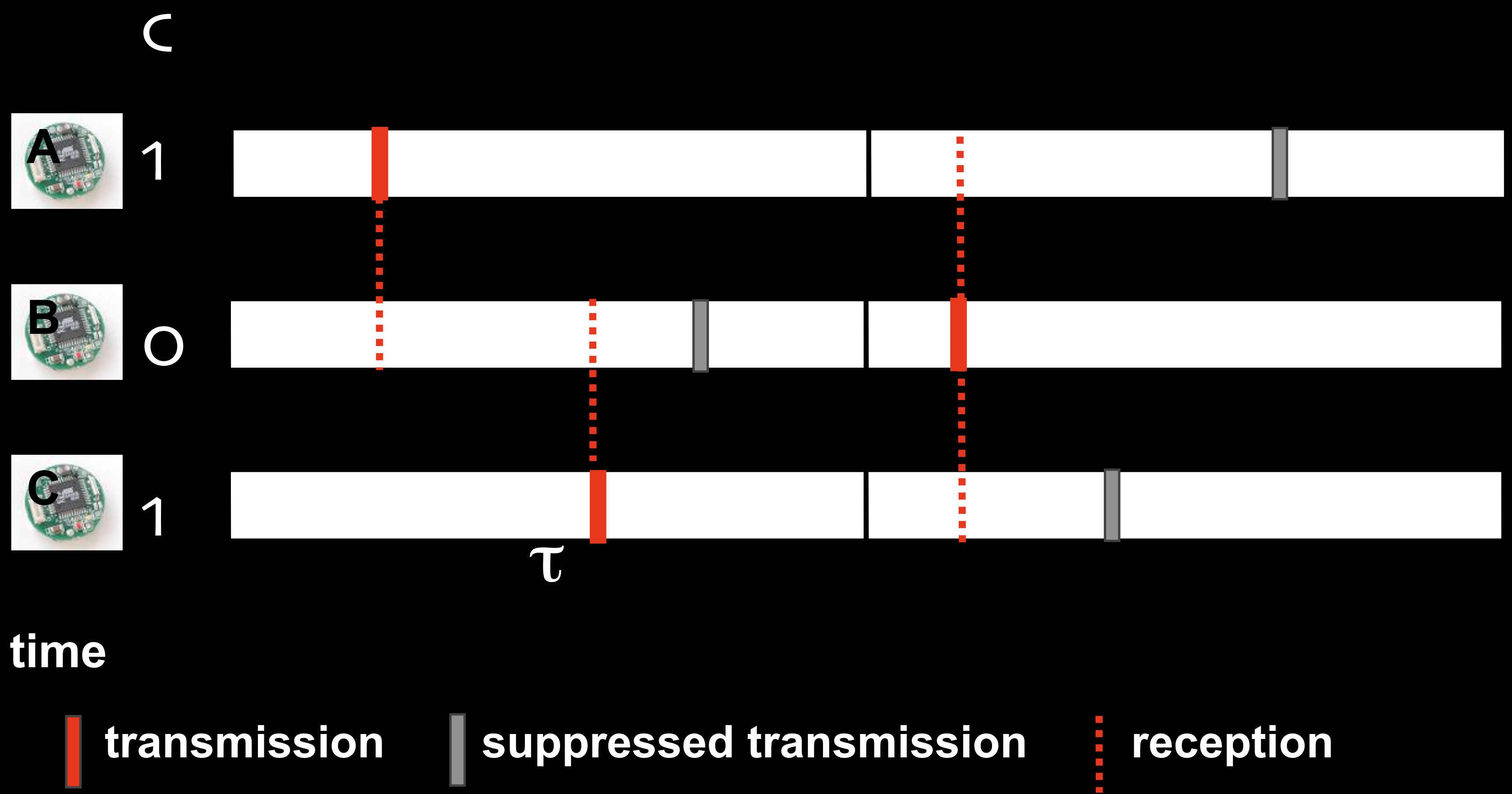
Example Execution

k=1

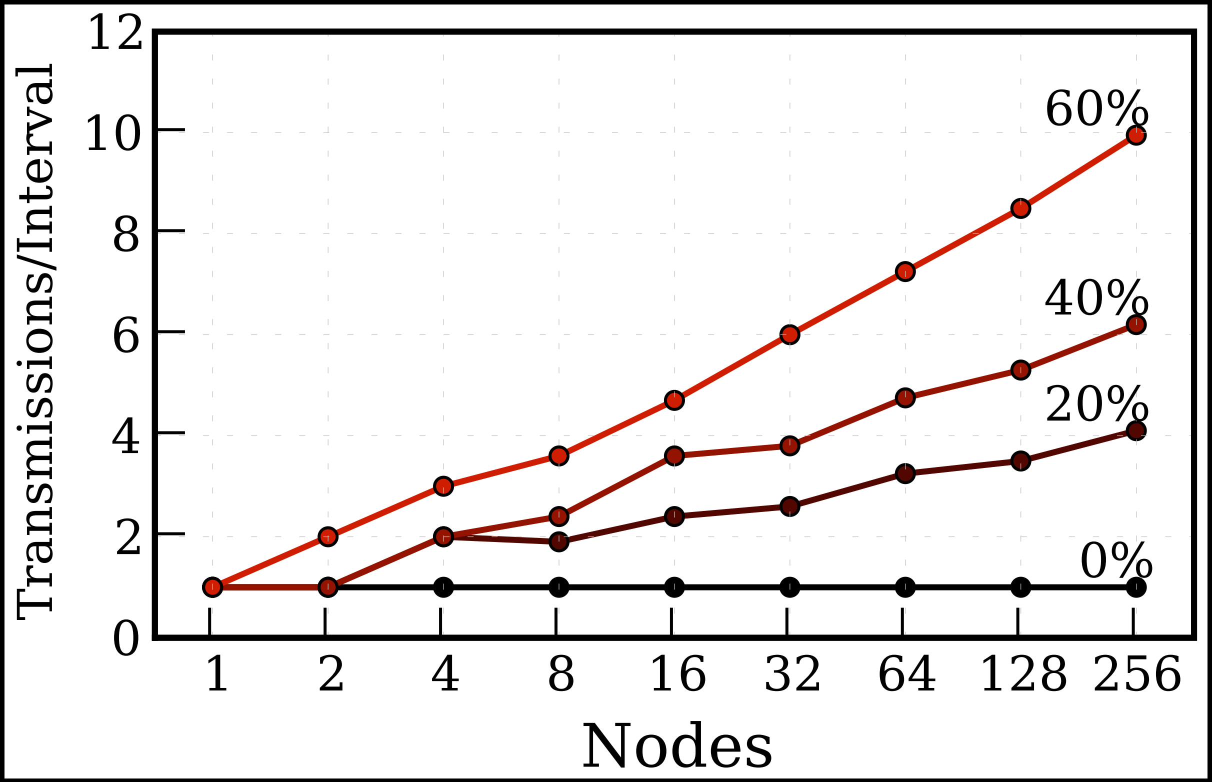


Example Execution

k=1



log(L)
(k=1)



Transmissions/Interval
0
2
4
6
8
10
12
1
2
4
8
16
32
64
128
256
Nodes
60%
40%
20%
0%

Logarithmic Behavior

- Transmission increase is due to the probability that one node has not heard n transmissions
- Example: 10% loss
  - 1 in 10 nodes will not hear one transmission
  - 1 in 100 nodes will not hear two transmissions
  - 1 in 1000 nodes will not hear three, etc.
- Fundamental bound to maintaining a per-node communication rate

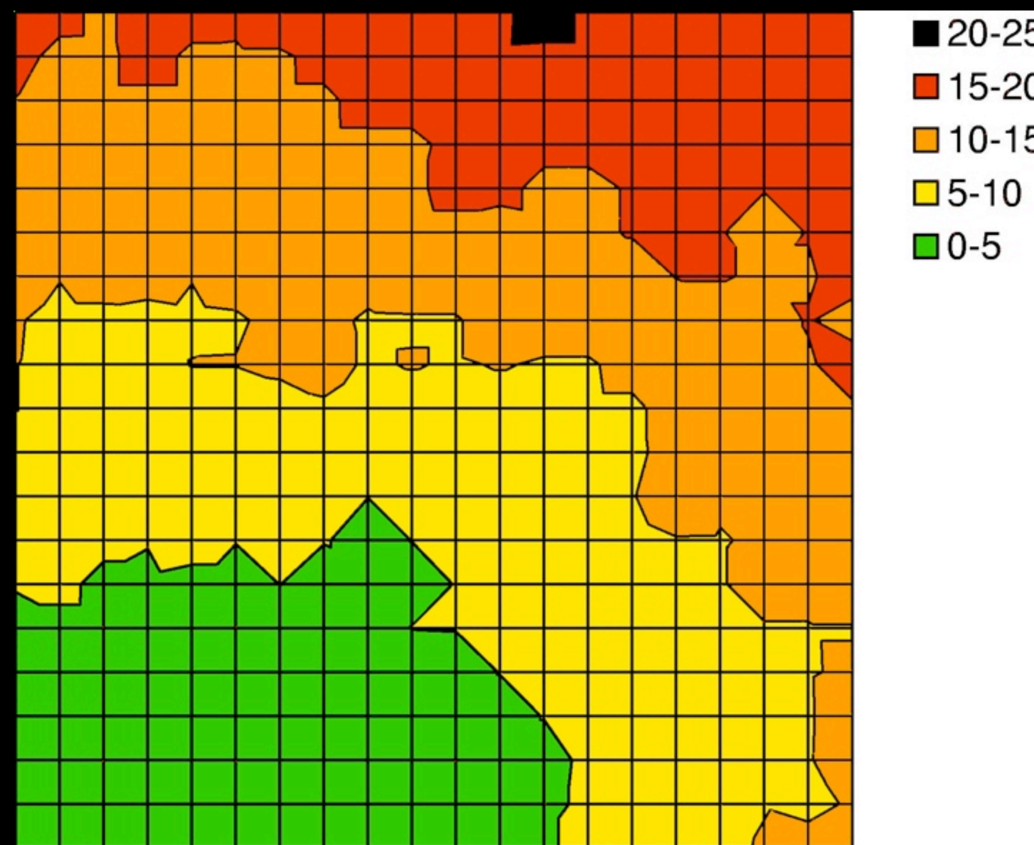
Intervals

(exponential timers)

- Two constants: $\tau_l << \tau_h$
- One variable: τ
- Operate over time intervals of length τ
  - At end of interval, double τ up to τ_h
  - On detecting an inconsistency, set τ to τ_l
- Consistent network has large intervals
- Inconsistency leads to small intervals

Simulated Propagation

- Inconsistency at lower left corner
- 16 hop network
- Time to reception in seconds
- Set τ_l = 1 sec
- Set τ_h = 1 min
- 20s for 16 hops
- Wave of activity



Example: Routing
(distance vector)

- Reset τ when
  - Receive a packet with a higher distance
  - Distance drops significantly
- Use τ_l =32ms, τ_h =1 hour, compare with fixed beacons of 30s
  - Reduces control traffic by 75%
  - Reduces latency to repair loops by 99.9%

Details

- Current implementations require
  - 4-7 bytes of RAM
  - 30-100 lines of code
- Diversity addresses topology edge cases
  - Node diversity
  - Spatial diversity
  - Temporal diversity
- Self-regulating and adapting

Summary

- Trickle: algorithm for eventual consistency in a wireless network
- Very simple, highly efficient
- Many uses
  - Routing topology
  - Reliable broadcasts
  - Neighbor discovery

References

- Philip Levis, Eric Brewer, David Culler, David Gay, Samuel Madden, Neil Patel, Joe Polastre, Scott Shenker, Robert Szewczyk, and Alec Woo. "The Emergence of a Networking Primitive in Wireless Sensor Networks." In Communications of the ACM, Volume 51, Issue 7, July 2008.

Best starting point.

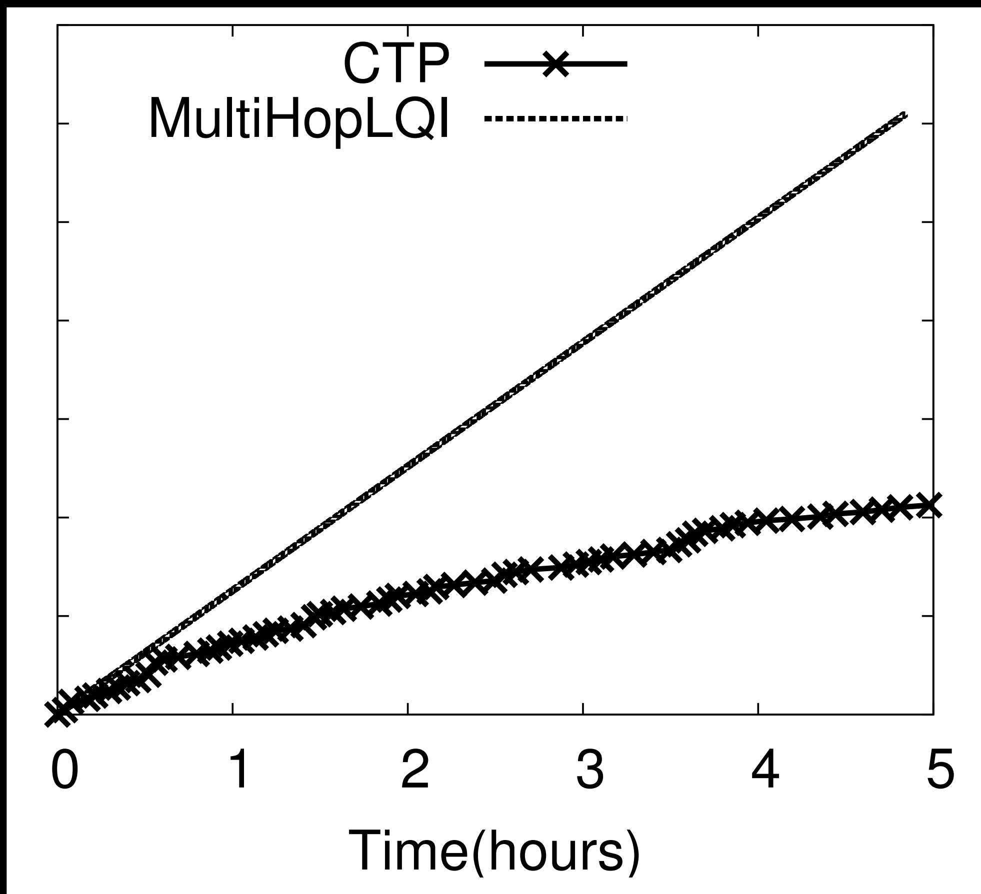
- Jonathan W. Hui and David E. Culler. "IP is Dead, Long Live IP for Wireless Sensor Networks." In Proceedings of the 6th International Conference on Embedded Networked Sensor Systems (SenSys), 2008.
- Philip Levis, Neil Patel, David Culler, and Scott Shenker. "Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks." In Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2004).
- Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, and Philip Levis. "Robust and Efficient Collection through Control and Data Plane Integration." Technical Report SING-08-02.

Questions

Draft Plans

- Precise algorithm specification
- Statement of how to reference algorithm in protocol specification documents
  - Consistency criteria
  - Constants: k, τ_l, τ_h
- Discussion of interoperability concerns and performance implications of inconsistent constant values

Experimental Data 1



Experimental Data 2