Distributed Coordination

Azer Bestavros

September 23, 2003

Scribe: Wei Li

References (and quotations)

Computer Science

[1] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks, MobiCom 2001.

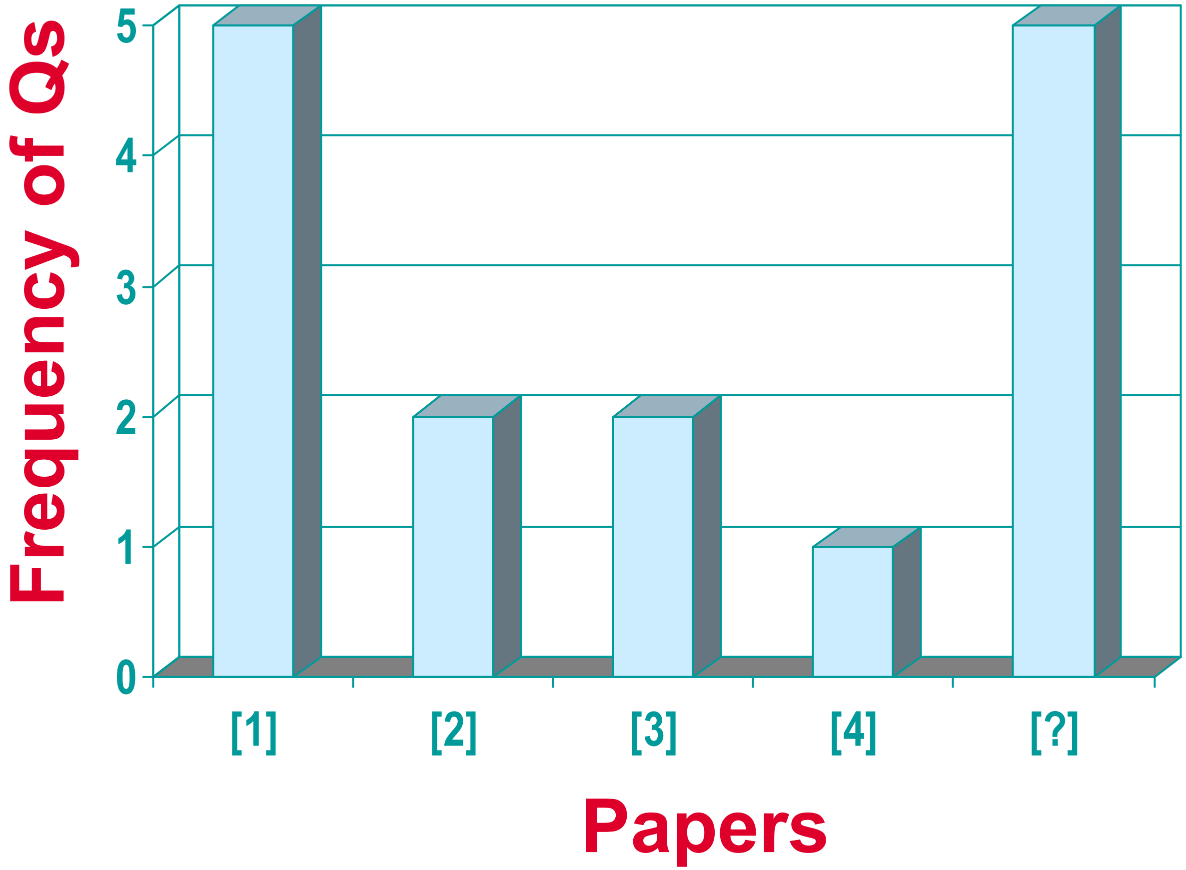
[2] K. Whitehouse and D. Culler. Calibration as parameter estimation in sensor networks. WSNA 2002.

[3] J. Elson, L. Girod and D. Estrin. Fine-Grained Network Time Synchronization using Reference Broadcasts, OSDI 2002.

[4] R. Karp, J. Elson, D. Estrin, and S. Shenker. Optimal and Global Time Synchronization in Sensornets. Technical Report CENS. April 2003.

Did you read the papers?

Computer Science



Why Keep the Radio On?

Computer Science

- To save power, an idle node should snooze
- But, multi-hop networks require (otherwise idle) nodes to forward packets bound for other hosts
- Need to balance snooze schedules with network connectivity/capacity

- An optimization problem:
  - Minimize power consumption without sacrificing capacity

Span [1]

Routing Layer	GPSR	DSR	AODV
	Span		
MAC/PHY	802.11		

- Sits between the routing and MAC layers
- “Preserves network capacity”
- Rotates coordinator duties among the hosts
- Allows idle hosts to sleep normally
- Takes pity on nearly-dead batteries

Coordination Logic

Computer Science

- A node is eligible to be a coordinator if it cannot establish that two of its neighbors can reach each other directly or via one or two coordinators.

  - No guarantee of minimum # of coordinators, but preserves connectedness.
  - Random delay factors lower the chance of coordinator contention.

Coordinating coordinators ☺

Computer Science

- Periodically decide eligibility
- If eligible wait for a random delay that reflects "cost"
  - % of consumed battery charge
  - % of neighbors in need of node
- Volunteer if no volunteers step up before delay expires
- Bridging hosts are always coordinators and die faster, but mobile networks tend not to suffer from this issue.

$$delay = \left(\left(1 - \frac{E_r}{E_m} \right) + \left(1 - \frac{C_i}{\binom{N_i}{2}} \right) + R \right) \times N_i \times T$$

E_r: Remaining energy

E_m: Maximum energy

C_i: Nodes potentially connected by *i*

N_i: Nodes neighboring *i*

R: Random value from [0, 1]

T: Round-trip delay for a small packet

Coordinator Withdrawal

Computer Science

- Withdraw if every pair of neighbors can reach each other with other coordinators.
- To ensure fairness, withdraw if other nodes can become coordinators and satisfy this requirement.
- After notifying everyone of its withdrawal, a node continues to coordinate for a short time to minimize service impact.

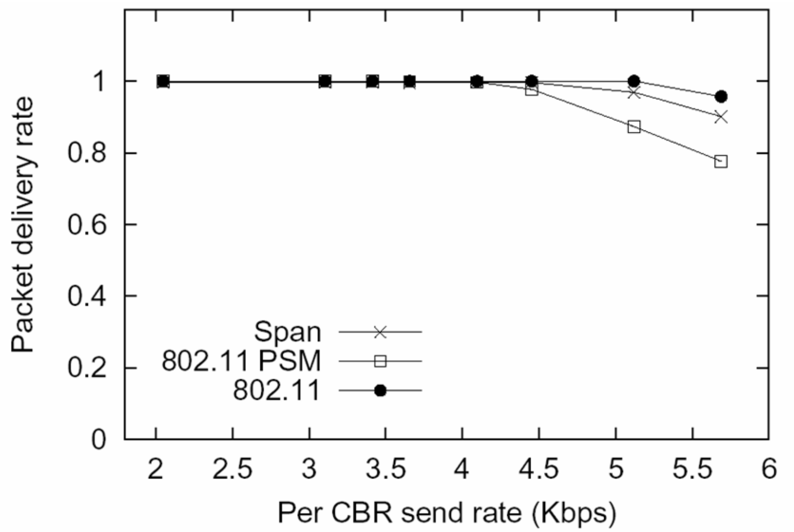
Evaluation

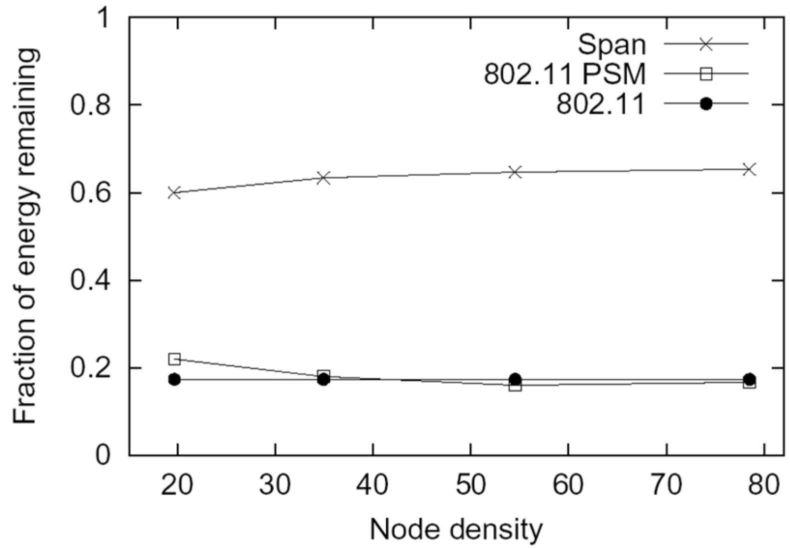
Boston University Computer Science

- Use Geographic Routing (GR) as routing protocol on top of:
  - SPAN
  - IEEE 802.11 (tweaked for SPAN)
  - IEEE 802.11 PSM

- Use ns-2 + wireless extensions to simulate
- Use energy model for energy consumption
- Use *random waypoint* to model/study mobility

Results

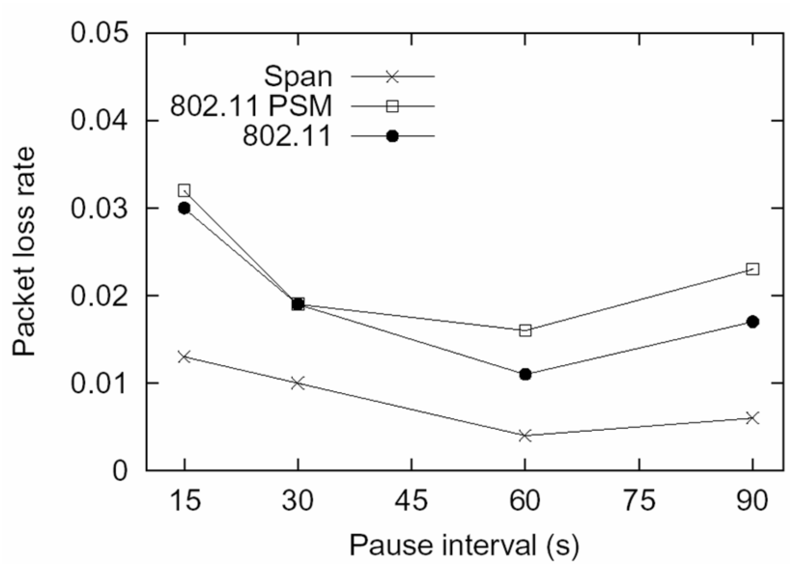


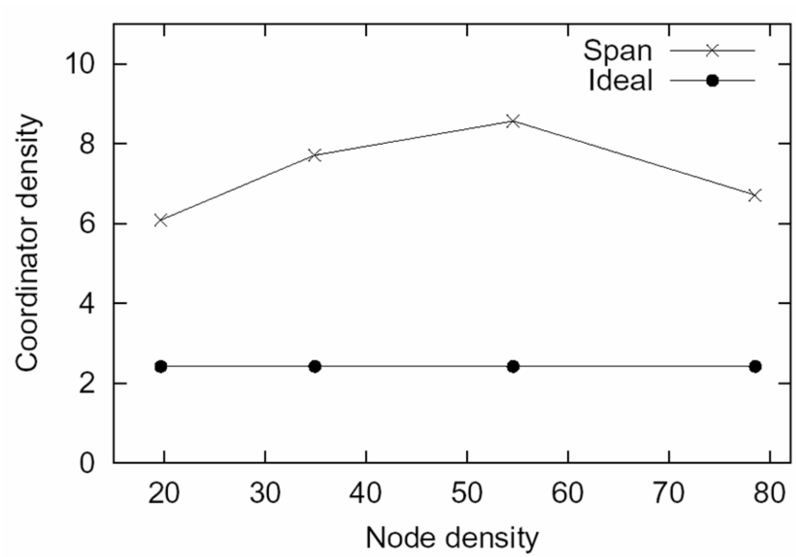


SPAN preserves capacity

SPAN preserves power

Results





SPAN shines with mobility

SPAN is not optimal

Questions

Computer Science

"The basic idea that a path with many short hops is sometimes more energy-efficient than one with few long hops could be applied to any ad hoc network with variable-power radios and knowledge of positions. This technique and Span's are orthogonal, so their benefits could potentially be combined" [1]

"What to they mean? How do they prove it?" – Georgios Smaragdakis

"It is mentioned that SPAN uses only local information to decide which nodes will be coordinators and which will sleep. But SPAN eventually uses the geographical forwarding algorithm. But dosn't geographical information require global information?" – Kanishka Gupta

Thoughts/Questions

Computer Science

- Does SPAN *really* preserve capacity, or does it merely preserve connectedness?

  "There are many mentions of "preserving system capacity" as a goal, but nothing concrete I found that specifically dealt with it. The good results seem to result (by design) from the broadcast nature and the coordinators filling in the (most) direct paths." – Jef Considine

- SPAN ignores "demand for connectedness".

  "I would think that having more coordinators amongst the set of nodes which are more active would yield benefits than having them spread out solely on the basis of the network topology." – Vijay Erramilli

  "How might Span benefit when electing coordinators if nodes advertised their expected traffic levels" – Bill Mullally

- Why preserve connectedness if it is not needed?

Thoughts/Questions

Computer Science

- Is this the right way to evaluate SPAN?

  "The evaluation is mainly simulation based. It will be more interesting to know what is the competitive analysis of SPAN. How close is it to an offline algorithm in saving the power." – Dhiman Barman

- How does it *"really"* scale?

  "What issues might arise in larger networks of thousands, or hundreds of thousand, which are the network sizes we usually expect in sensor networks. It seems like it would scale, since it uses local decisions for routing and power saving." – Luis Hernandez

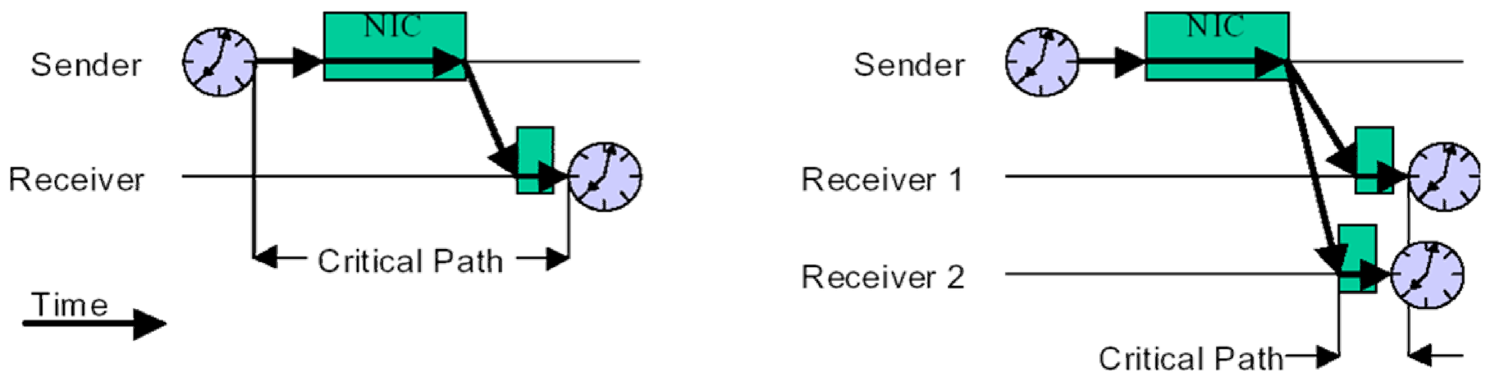
- Why not be proactive?

  "How could span be extended to encourage, or at least notify, nodes that better spatial configurations are possible." – Bill Mullally

Clock Synchronization [3]

Computer Science

- Needed for a host of reasons in any distributed system, and in particular in sensor networks

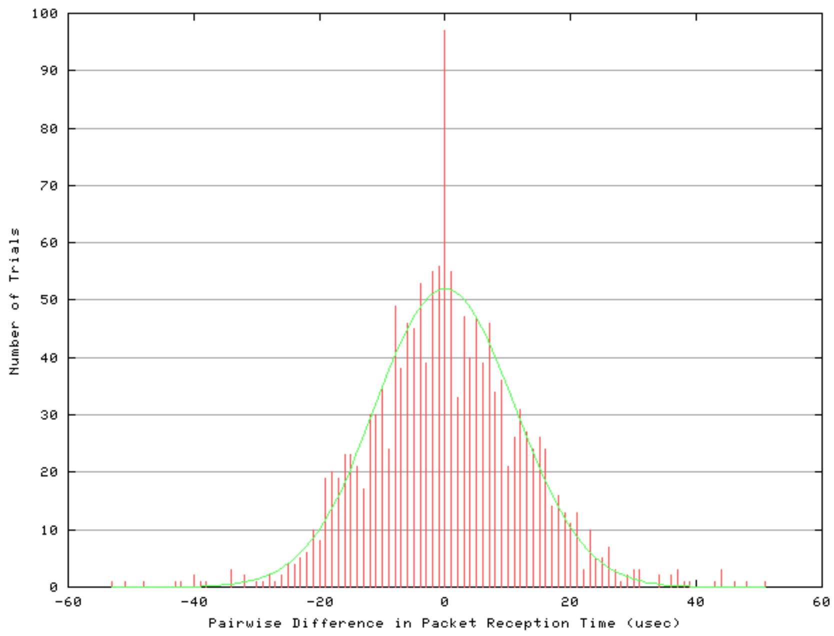


- Traditional techniques (e.g., NTP) rely on synching clients with servers—introducing 4 sources of non-determinism
- *Reference Broadcast Synchronization* leverages physical-layer broadcasts of wireless networks to remove the most nondeterministic part of the system from the critical path

RBS Clock Synchronization

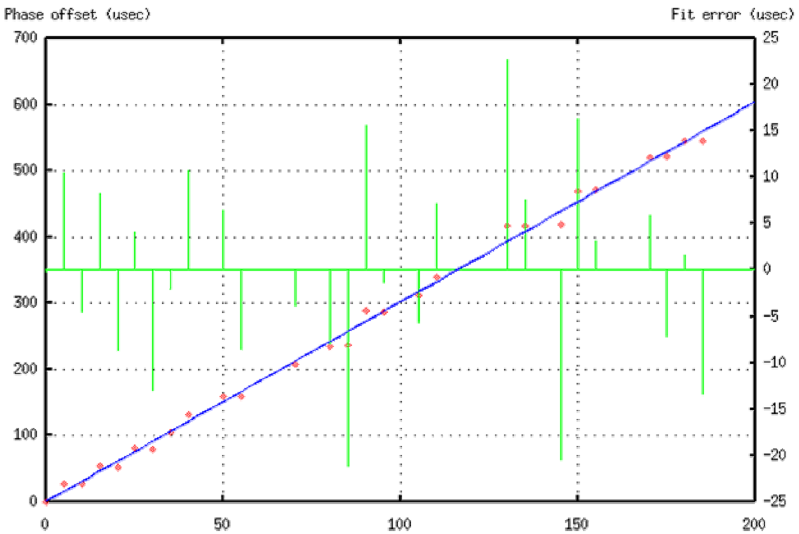
Computer Science

- RBS is only sensitive to propagation delay difference and to receiver processing non-determinism
- Non-determinism introduced by receiver processing is well-behaved (a.k.a., normal distribution)



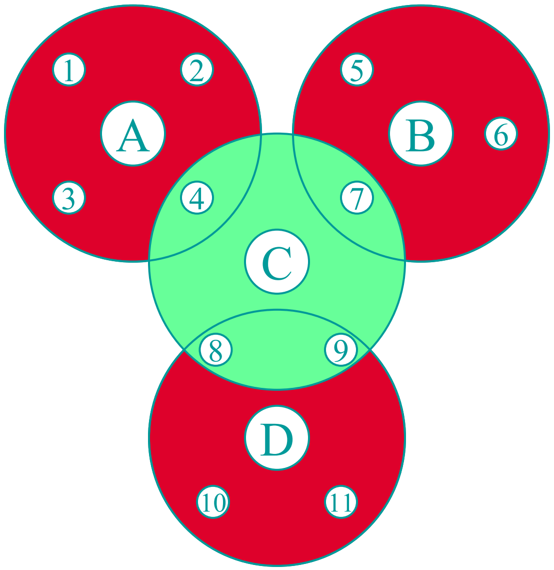
RBS Clock Synchronization

- Well behaved = Good
  - Error can be reduced statistically, by sending multiple pulses and building confidence in estimate
- Problem: Clock skew
  - It takes time to send multiple pulses
  - By the time we do, clocks would have drifted
- Solution: Use better model
  - Don't average; fit a line instead



RBS: Multi-Hop Sync

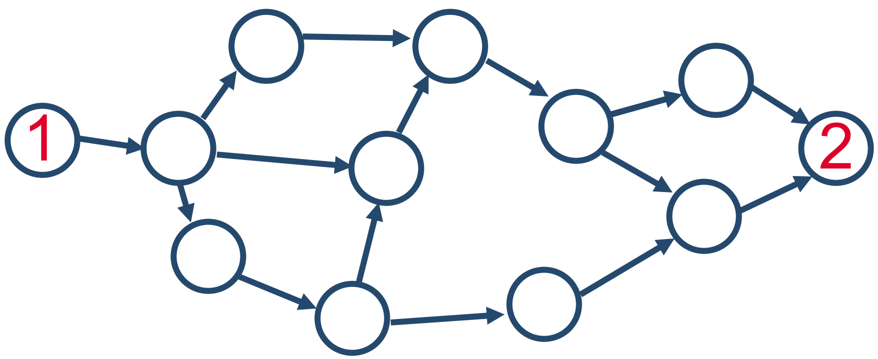
- How to sync nodes in different broadcast domains?
  - Nodes at the intersection of such domain would sync the domains
  - Node 4 would reconcile A & C
  - Node 7 would reconcile B & C
  - Node 8 would reconcile C & D
  - Node 9 would reconcile C & D



- Hmm.. What happens if synching D through 8 does not yield the same answer as through 9?

Multi-Path Sync [4]

- Synchronization through multiple paths between two nodes (not in the same domain)
- Consider the set of paths from $r_1 \rightarrow r_2$



- Each path is a sequence of nodes with adjacent nodes in the same broadcast domain

Multipath Sync

Computer Science

Consistency

- Using RBS to convert the local times of i to j and j to k is not the same as from i to k. Transitivity does not hold ➔ Pairwise synchronization is not necessarily globally consistent.

Precision

- Pairwise synchronization is not optimally “precise” because it ignores relevant information (e.g., sync results from multiple sources/receivers).

Two sides of the same coin!

- Most precise synchronization ➔ global consistency

The Model

Boston University
Computer Science

- At time *k*, a node will get a "pulse"
- What is the difference between the "universal" time U_k (which is unknown) and the local time T_i ?

$$y_{ik} = U_k + T_i + e_{ik}$$

The Model

Computer Science

- Consider one such path. We can model that by a sequence

$r_{i_1}, s_{k_1}, r_{i_2}, s_{k_2}, \cdots, s_{k_t}, r_{i_{t+1}}$ where $r_{i_1} = r_1$ and $r_{i_{t+1}} = r_2$

- Think about the above sequence as

- What is the value of T_1-T_2 ?

Estimating Clock Offsets

Computer Science

- An *unbiased* estimator of T_1-T_2 over that path is

$$y_{i_1,k_1} - y_{i_2,k_1} + y_{i_2,k_2} - \cdots - y_{i_{t+1},k_t}$$

- Substituting from $y_{ik} = U_k + T_i + e_{ik}$ we get

$$T_1 - T_2 + e_{i_1,k_1} - e_{i_2,k_1} + e_{i_2,k_2} - \cdots - e_{i_{t+1},k_t}$$

- The variance of this unbiased estimator is:

$$V_{i_1,k_1} + V_{i_2,k_1} + V_{i_2,k_2} + \cdots + V_{i_{t+1},k_t}$$

Estimating Clock Offsets

Computer Science

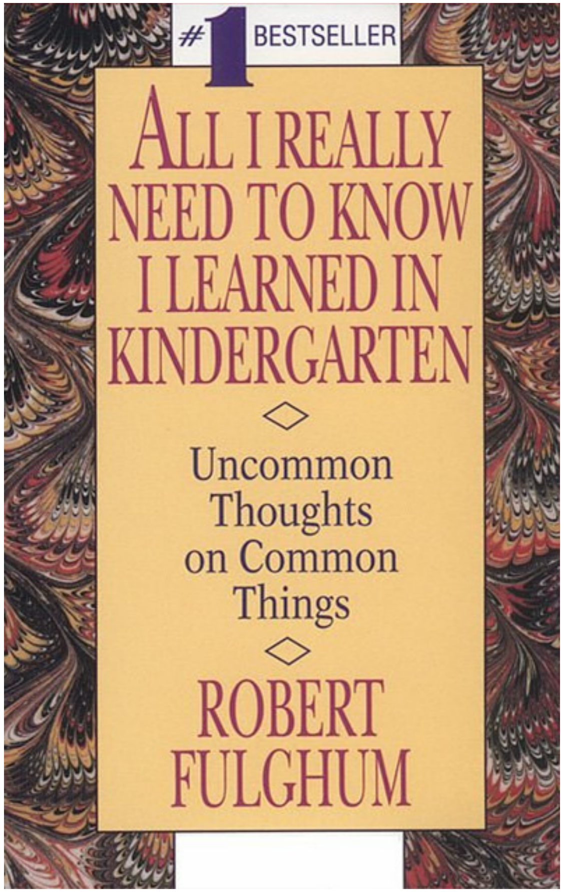
- What about all the other paths? Each one of them will give us a different unbiased estimate!
- Any *weighted combination* (as long as the sum of weigths = 1) is also an estimator!
- We need to find the set of weights that minimize the variance

Theorem 1. *The unbiased estimators of $T_1 - T_2$ are precisely the linear expressions $\sum_{ik} f_{ik} y_{ik}$ such that $\{f_{ik}\}$ is a flow of value 1 from r_1 to r_2. Here f_{ik} is positive if the flow on edge $\{r_i, s_k\}$ is directed from r_i to s_k, and negative if the flow is directed from s_k to r_i. The variance of the unbiased estimator $\{f_{ik}\}$ is $\sum f_{ik}^2 V_{ik}$. A similar statement holds for the unbiased estimators of $T_j - T_i$, for any i and j.*

"Someday you'll appreciate this"

Computer Science

- Flow of currents in alternative paths between two nodes in an electric circuit is such that the power consumed is minimal
  - Power = Voltage * Current
  - Voltage = Current * Resistance
  - Power = Current^2 * Resistance
- Well, not quite kindergarten, but high school ☺



Estimating Clock Offsets

Computer Science

- I want to find the f_{ik}'s that minimize

$$\sum f_{ik}^2 V_{ik}$$

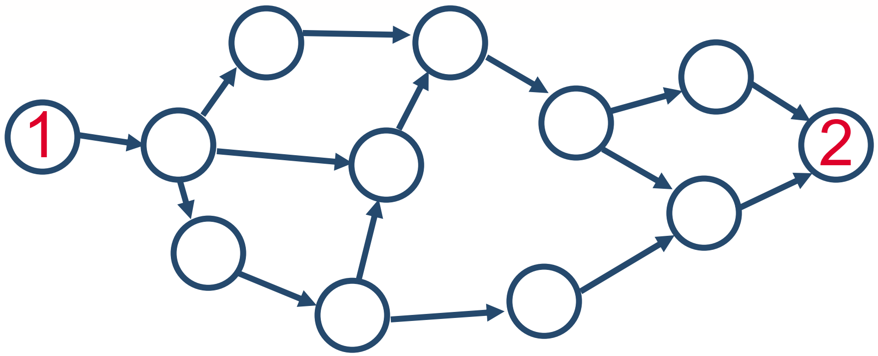
- …and I know that in an electric circuit the following is minimized

$$\sum_{(u,v)\in E} c(u,v)^2 R(u,v)$$

- Map path *weights* to *current* values, and map *variances* to *resistances*

Estimating Clock Offsets

Boston University
Computer Science



- Think of the above as an electric circuit.
- Force a current of value 1 (amp) from 1→2
- Current will flow on every one of the above "hops"
- The value of those currents will minimize power
- By virtue of equivalence the current values are the weights that minimize the variance of the estimator of T_1-T_2 !
- BTW, the voltage differential between 1 & 2 = R_{eff} (*1 amp)

Estimating Clock Offsets

Computer Science

- By equivalence current values are the weights that minimize the variance of the estimator of T1-T2 !

Theorem 2. *The minimum variance of an unbiased estimator of $T_1 - T_2$ is the effective resistance between r_1 and r_2, and the corresponding estimator is $\sum_{ik} f_{ik} y_{ik}$ where f_{ik} is the current along the edge from r_i to s_k when one unit of current is injected at r_1 and extracted at r_2.*

Global Consistency

Computer Science

- Let A(i,j) denote the minimum-variance estimator for $T_i - T_j$
- Are the A(i,j)'s consistent? In other words is it the case that

$$A(i,m) + A(m,j) = A(i,j)$$

- Yes! Proof follows trivially from superposition principle for currents

$$c_{e_1+e_2}(u,v) = c_{e_1}(u,v) + c_{e_2}(u,v)$$

which means that

$$A(i,j) = y \cdot c(e(i,j)) = y \cdot c(e(i,m)) + c(e(m,j)) = A(i,m) + A(m,j)$$

What about skew?

Computer Science

- Use pairs of synch signals and use “differences” (which would cancel out the offsets and let us apply the previous techniques to estimate the skew
- Use technique over short timescales to estimate local times and over longer timescales to estimate skew

Questions/Thoughts

Computer Science

- If resistance = variability, what is capacitance, inductance, …?
- What other "theories" can we leverage to solve CS problems?
- Is it all "theory"? Is it practical? Why bother if not?

> It is an open question whether this result will lead to a practical synchronization method. The key issue is whether the rate of iterations needed to meet the desired precision bounds is too energy intensive. However, as we described in Section 4, if one is only interested in synchronizing a pair of nodes, then one can greatly reduce the scope of nodes involved in the synchronization process.
>
> Even if our results do not lead to a feasible synchronization algorithm, they can provide a yardstick against which to compare methods of computing offsets and skews from reference-broadcast time-of-arrival data. Designers now know the optimal results that can be achieved, and can make their own energy-precision tradeoffs with that in mind.