



Sincronização do tempo em RSSF

Aluno: Euler Xavier de Oliveira

Disciplina: Redes de Sensores sem Fio



Conteúdo

- Introdução
- Motivações
- Implementações
 - Problemas
 - Abordagens



Introdução

- Sincronizar
 - Combinar (ações) para o mesmo tempo; ajustar eventos com rigorosa precisão
- Segundo
 - Definido pelo *International System of Units*
 - “É a duração de 9.192.631.770 períodos da radiação correspondente à transição entre os dois níveis hiperfinos do estado fundamental do Césio-133”



Introdução

- Sistemas distribuídos quaisquer
 - Não apenas sistemas computacionais
- Remadores podem ter um ritmo ditado pelas batidas de um tambor
- Metrônomo marca o compasso de toda uma orquestra sinfônica



Introdução

- Tomar cuidado com o que se usa pra fazer sincronismo!
- Exemplo: ataque “simultâneo” dos exércitos
 - **Às 6:00h.** E se estiverem em regiões com diferentes fusos horários?
 - **Ao nascer do sol.** E se se situarem em meridianos suficientemente distantes?



Motivação

Usos em sistemas distribuídos tradicionais

■ Logging e Debugging

- É importante saber momentos em que determinados eventos ocorreram, a fim de:
 - Correlacionar informações de atividades dos nós
 - Possibilitar entendimento global do sistema a ser observado
- *Logs* sem sincronização dificultam (ou impossibilitam)
 - Determinar causalidade
 - Reconstruir exata seqüência de eventos



Motivação

Usos em sistemas distribuídos tradicionais

- **Consultas a Bancos de Dados**
 - Esquemas atualmente implementados procuram esconder a complexidade da rede
 - Realizam complicadas pesquisas distribuídas, como se fossem consultas em um banco centralizado
 - Abstrai os detalhes em prol do usuário
 - Exigem acesso sincronizado aos dados



Motivação

Usos em RSSF

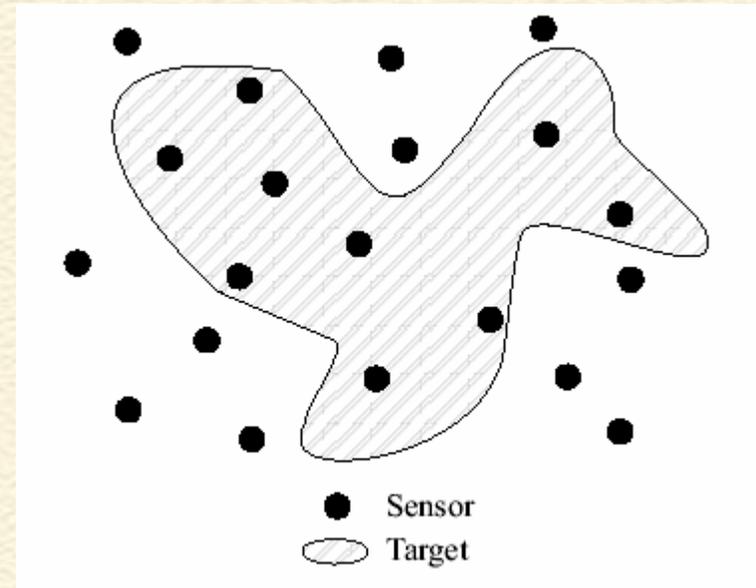
- **Integração de dados multi-sensores**
 - *Multi-sensor data integration*
 - Sensoriamento distribuído
 - Colocação de vários sensores próximos ao fenômeno de interesse;
 - Aspecto importante:
 - Informações enviadas por cada sensor não possuem grande valor;
 - Combinação de informações permite uma **visão de mais alto nível**
 - Não detectável por um sensor isolado



Motivação

Usos em RSSF

- Integração de dados multi-sensores (continuação)
 - Exemplo:
 - Nodos espalhados sobre uma região marítima;
 - Área hachurada é a área poluída;
 - Nodos possuem sensores químicos;
 - Envia:m:
 - Posição;
 - Valor booleano:
 - ✓ Estou na área contaminada
 - ✓ Não estou
 - Pode-se inferir
 - Tamanho da mancha
 - Formato
 - Velocidade

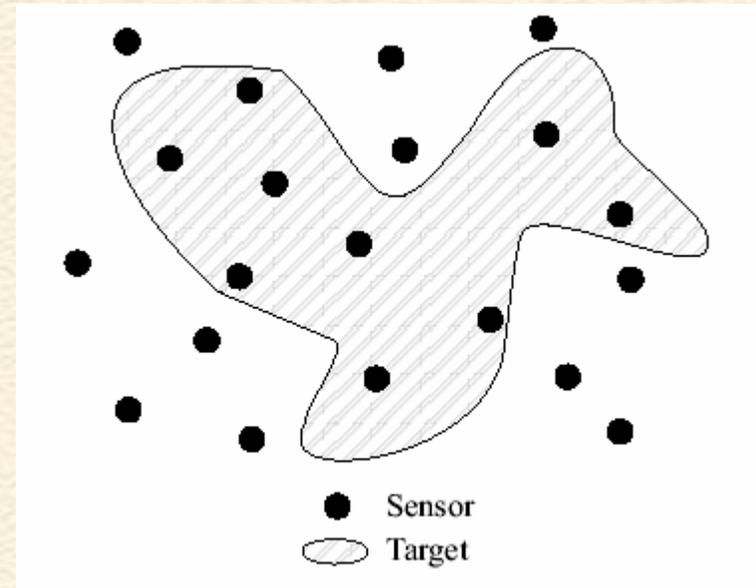


Motivação

■ Integração de dados multi-sensores (continuação)

- Sincronização é necessária?
 - **Não**, se o fenômeno é estacionário;
 - **Não**, se atraso na transmissão (*jitter*) é insignificante
 - Se comparado com a velocidade de espalhamento da região contaminada
 - Uma consulta simultânea a todos os nodos satisfaz à consulta
 - **SIM**, se a mancha se move rapidamente
 - Relativamente ao tempo gasto na transmissão dos dados;
 - Neste caso, *jitter* compromete a validade das consultas;
 - Solução: adicionar *timestamp*;

Usos em RSSF



Motivação

Usos em RSSF

- **Processamento na própria rede**
 - *In-network processing*
 - Utilização de
 - Processamento local
 - Colaboração hierárquica
 - Domínio do conhecimento
 - Convertem dados crus em informações de mais alto nível
 - Aumentam o custo computacional por nó



Motivação

Usos em RSSF

- **Processamento na própria rede**
 - Reduz dramaticamente o gasto no envio de informações
 - Sistema perfeito: reduz a máxima quantidade possível de dados, o mais cedo possível
 - Denomina-se *redução de dados*



Motivação

Usos em RSSF

- **Processamento na própria rede**
 - **Exemplo: supressão de duplicações**
 - Caso especial de redução de dados que depende de sincronização;
 - Previne notificação redundante de um mesmo evento
 - Por parte de dois ou mais sensores
 - Relógios sincronizados ajudam os sensores a responderem à questão:
 - Vimos o mesmo evento ou não?



Motivação

Usos em RSSF

■ Escalonamento em sistemas TDMA

- *Energy-efficient Radio scheduling*
- Nós gastam considerável energia
 - mesmo quando estão apenas passivamente ouvindo;
- Protocolos MAC para RSSF foram projetados para
 - Proporcionar a possibilidade de rádios ficarem desligados o maior tempo possível
- Muita energia pode ser conservada
 - Se acordarem apenas para trocar curtas mensagens
- Ideal:
 - Dormir sempre que possível
 - Acordar apenas quando necessário

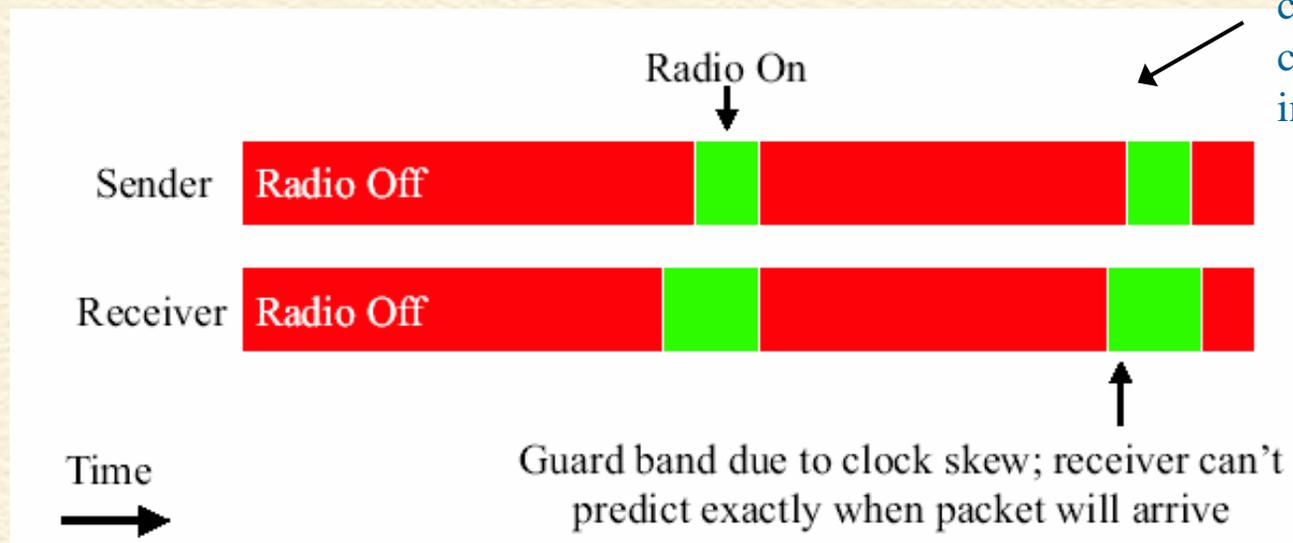


Motivação

Usos em RSSF

■ Escalonamento em sistemas TDMA

- Nós envolvidos devem gastar tempo constantemente para realizar novas sincronizações, devido ao *clock-skew*
 - Osciladores de quartzo típico podem variar em uma parte para cada 10^5 ;
 - Ou seja, 0,6ms após 60s;



Emissor e receptor combinaram de se comunicar em intervalos regulares



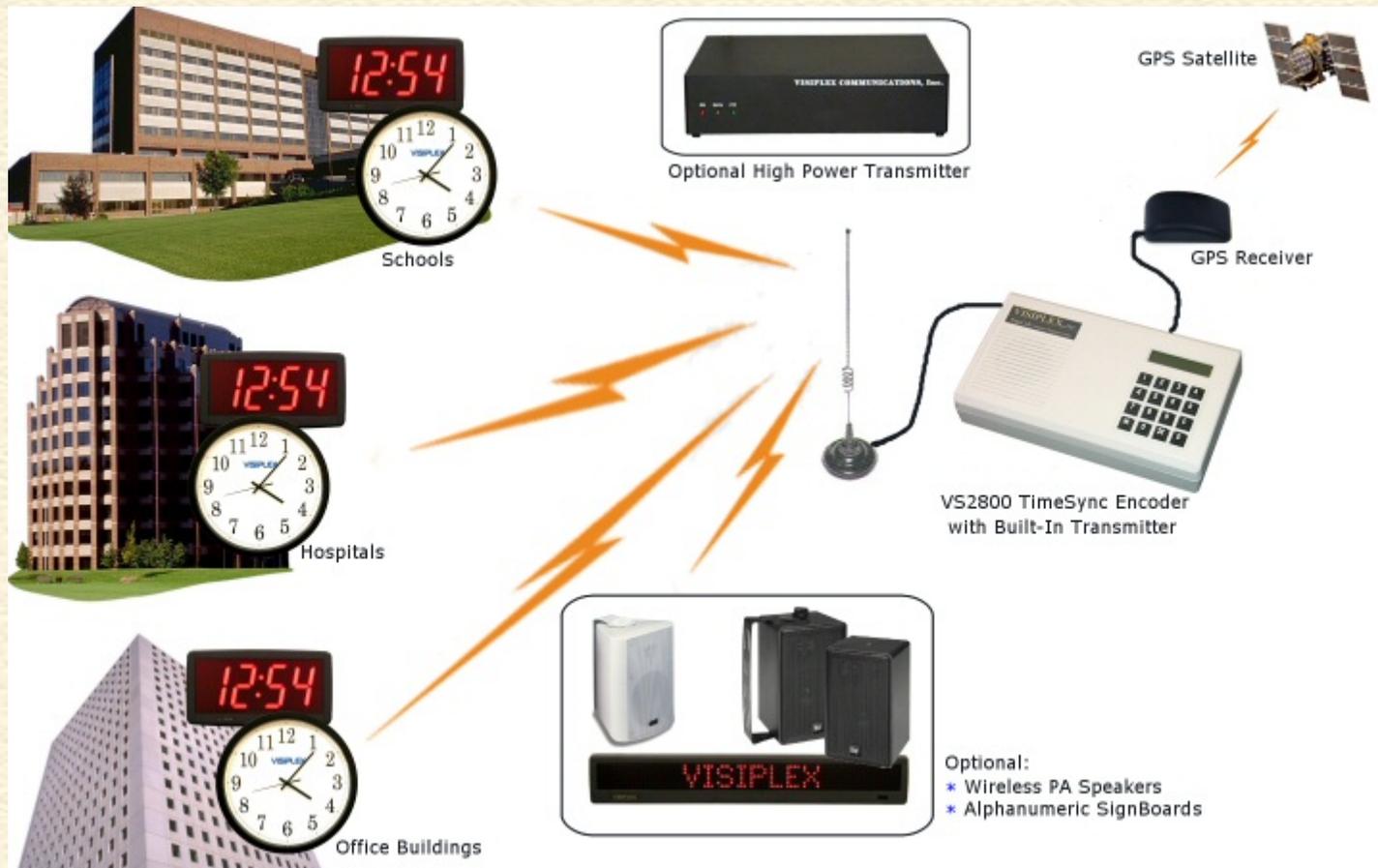
Implementação de sincronismo

- Utiliza-se, em larga escala, o protocolo **NTP** (*Network Time Protocol*);
- Sincroniza relógios de servidores e roteadores na Internet;
 - Todo Windows/XP possui um cliente NTP
- Precisão nominal:
 - Dezenas de milisegundos em WANs;
 - Sub-milisegundos em LANs;
 - Sub-microsegundos, usando uma fonte de tempo precisa



Implementação de sincronismo

■ Utilizando GPS:



Implementação de sincronismo

- NTP é uma técnica *two-way*;
- Técnicas *two-way* assumem:
 - O tempo de transferência em um caminho é recíproco
 - Tempo gasto no envio de um pacote é igual ao tempo gasto na recepção
 - Portanto existe determinismo
- *Delay* é estimado como a metade do tempo de trânsito do pacote



Implementação de sincronismo

- Exemplo:
 - Nodo A inicia o protocolo de sincronização;
 - t_i é o timestamp no tempo i ;
 - t_1 e t_4 são medidos por A;
 - t_2 e t_3 são medidos por B;
 - $d1$ = atraso na propagação da mensagem;
 - $d2$ = *clock offset* entre A e B;
 - Considera-se $d1$ e $d2$ como constantes;



Implementação de sincronismo

- Exemplo:

- Dados:

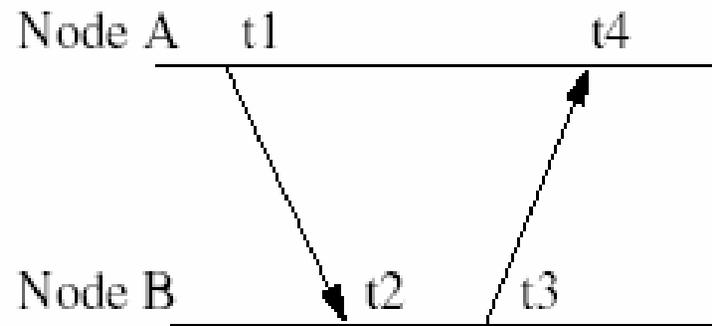
- $t1 = 0$;
- $t2 = 5$;
- $t3 = 7$;
- $t4 = 10$;

- Inference-se:

- $d1$ (delay)= 4;
- $d2$ (clock offset)= 1;

$$d1 = \frac{(t2 - t1) + (t4 - t3)}{2}$$

$$d2 = \frac{(t2 - t1) - (t4 - t3)}{2}$$



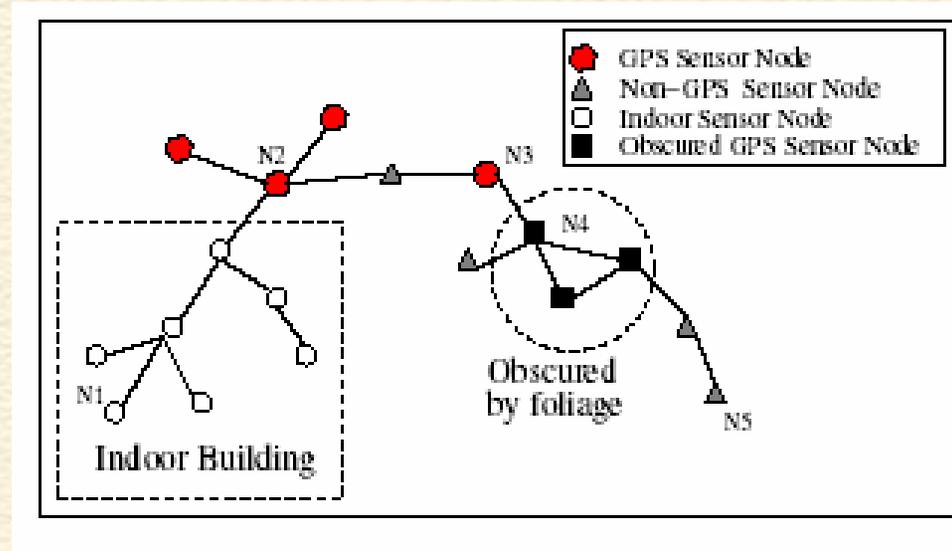
$$t4 = t3 + d1 - d2$$

$$t2 = t1 + d1 + d2$$



Implementação de sincronismo

- Inviabilidades em RSSF
 - GPS:
 - Nem sempre sensores estão sob seu alcance
 - *Indoor Buildings*
 - Regiões obscurecidas
 - ✓ Por folhagens
 - Custo:
 - 1 nó = 5US\$;
 - 1 receptor de GPS = 50US\$
 - Entretanto pode ser usada para o nó de referência



Implementação de sincronismo

- Inviabilidades em RSSF
 - Técnicas two-way (como o NTP)
 - Assumem que o atraso no envio das mensagens é constante
 - Ida
 - Volta
 - Em RSSF, esta hipótese deve ser descartada
 - Na realidade, pode haver variações nos atrasos, o que impossibilita o uso do NTP
 - ✓ A menos que precisão no sincronismo não seja um requisito importante



O problema dos atrasos em RSSF

- São não determinísticos;
- Podem ser ordens de magnitude maiores do que a precisão requerida para sincronização;
- Exigem análise cuidadosa, para que compensações sejam implementadas
- Possuem, segundo [1], 6 fatores principais



Fatores que geram atrasos

1 : Tempo de envio

- Monta a mensagem;
- Enviar requisição de envio à camada MAC, no lado do transmissor
- Depende de:
 - Overhead na chamada ao sistema;
 - Carga atual do processador;
- É altamente não determinístico;
- Pode ser tão grande quanto centenas de ms



Fatores que geram atrasos

2 : Tempo de acesso

- Atraso ocorre devido à espera pelo acesso ao canal de transmissão;
- Este intervalo finaliza quando a transmissão efetivamente inicia;
- É a parte menos determinística
 - no processo de entrega de mensagens em RSSF;
- Varia de alguns ms até alguns segundos
 - Depende do tráfego atual na rede



Fatores que geram atrasos

3 : Tempo de transmissão

- Tempo que o transmissor leva para transmitir a mensagem;
- Este tempo é da ordem de dezenas de milisegundos
 - Depende do comprimento da mensagem e da velocidade do rádio



Fatores que geram atrasos

4 : Tempo de propagação

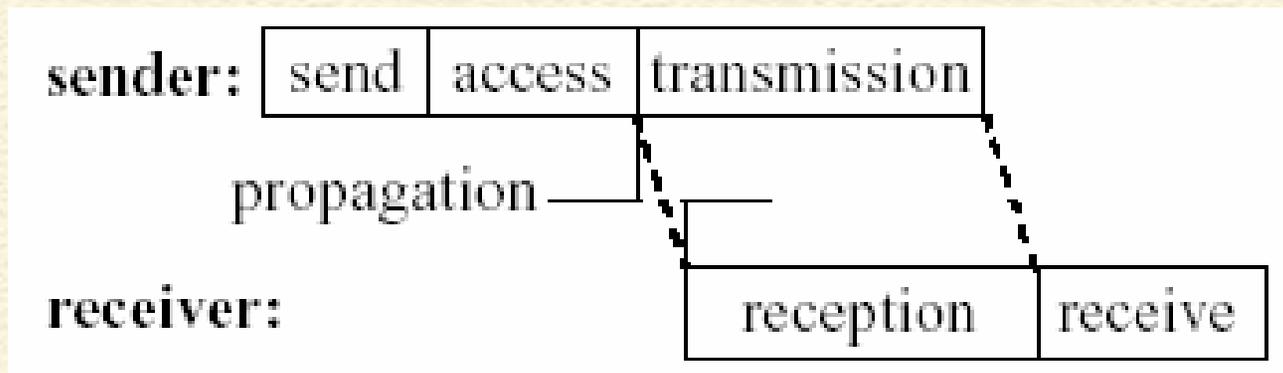
- Tempo gasto para o sinal navegar pelo ar até encontrar o receptor
 - A velocidade de propagação do rádio é $300\text{m}/\mu\text{s}$;
- É altamente determinístico, dada a distância entre os dois nós;
 - Erros decorrentes são insignificantes;
 - Cobertura do rádio é geralmente pequena;
 - Usualmente inferior a 100m;



Fatores que geram atrasos

5 : Tempo de recepção

- Tempo que o receptor gasta para receber a mensagem
 - É igual ao tempo de transmissão



Fatores que geram atrasos

6 : *Receive time*

- Tempo gasto para processar a mensagem recebida
 - E notificar à aplicação recebedora da mensagem;
 - Suas características são similares às do tempo de envio



Abordagens existentes

- RBS
- TPSN
- FTSP
- Post-facto



Abordagens existentes

RBS

- *Reference Broadcast Synchronization* [2];
- Nó periodicamente difunde *beacons* aos seus vizinhos;
- Nós receptores utilizam tempo de chegada dos *beacons* como pontos de referência
- Nós vizinhos compartilham *timestamps*
 - Para calcular oscilações de relógio;
 - Realizam sincronização mútua;



Abordagens existentes

RBS

- Vantagens:
 - Remove muitas fontes não determinísticas do problema tradicional;
 - Conseguem precisão de $1 \mu\text{S}$ após a terceira difusão
- Desvantagens:
 - Overhead causado pela troca de mensagens muito freqüente entre vizinhos;
 - Não realizam sincronização através de *multi-hops*



Abordagens existentes

TPSN

- *Timing-sync Protocol for Sensor Networks*
- Primeiramente cria uma estrutura hierárquica na rede;
- Realiza uma sincronização par a par ao longo das arestas
- Cada nodo é sincronizado
 - Pela troca de duas mensagens com o nodo de referência, um nível acima na hierarquia;



Abordagens existentes

TPSN

- Vantagens
 - Desempenho é melhor que o RBS, pois armazena o timestamp na camada MAC da pilha do rádio
- Desvantagens:
 - Não estima a variação do clock dos nodos, o que limita sua precisão;
 - Não suporta topologias dinâmicas
 - Seu desempenho foi verificado experimentalmente apenas em pequenas redes *multi-hop*;



Abordagens existentes

Post-facto

- Nesta solução, nodos estão normalmente fora de sincronismo;
- Ao ocorrer um estímulo, cada nodo armazena seu próprio *timestamp*;
- Realiza sincronização com um nodo que age como um *beacon* (ligado ao sistema GPS, por exemplo)
 - Que difunde um pulso de sincronização
 - Nodos que recebem este pulso sincronizam momentaneamente seus relógios
- Este tipo de sincronização não é aplicável em todas as situações, obviamente



Referências

- [1] Miklos Maroti, Branislav Kusy, Gyula Simon and Akos Ledeczi. *The Flooding Time Synchronization Protocol - Technical report* Institute for Software Integrated Systems Vanderbilt University
http://www.isis.vanderbilt.edu/publications/archive/Maroti_M_2_12_2004_The_Floodi.pdf
- [2] J. Elson, L. Girod and D. Estrin. Fine-Grained Network Time Synchronization using Reference Broadcasts *Proceedings of the fifth symposium OSDI '02*, December 2002.
- [3] S. Ganeriwal, R. Kumar, M. B. Srivastava. *Timing-Sync Protocol for Sensor Networks SenSys '03*, November 2003
- [4] Saurabh Ganeriwal, Ram Kumar and Mani B. Srivastava. *Timing-sync Protocol for Sensor Networks* Networked and Embedded Systems Lab, University of California, Los Angeles 56-125B Eng. IV, UCLA EE Dept., Los Angeles CA
- [5] Jeremy Elson and Deborah Estrin. *Time Synchronization for Wireless Sensor Networks* <http://www.circlemud.org/~jelson/writings/timesync/timesync.html>



Perguntas?

