Lecture 7

Internet Protokolle II

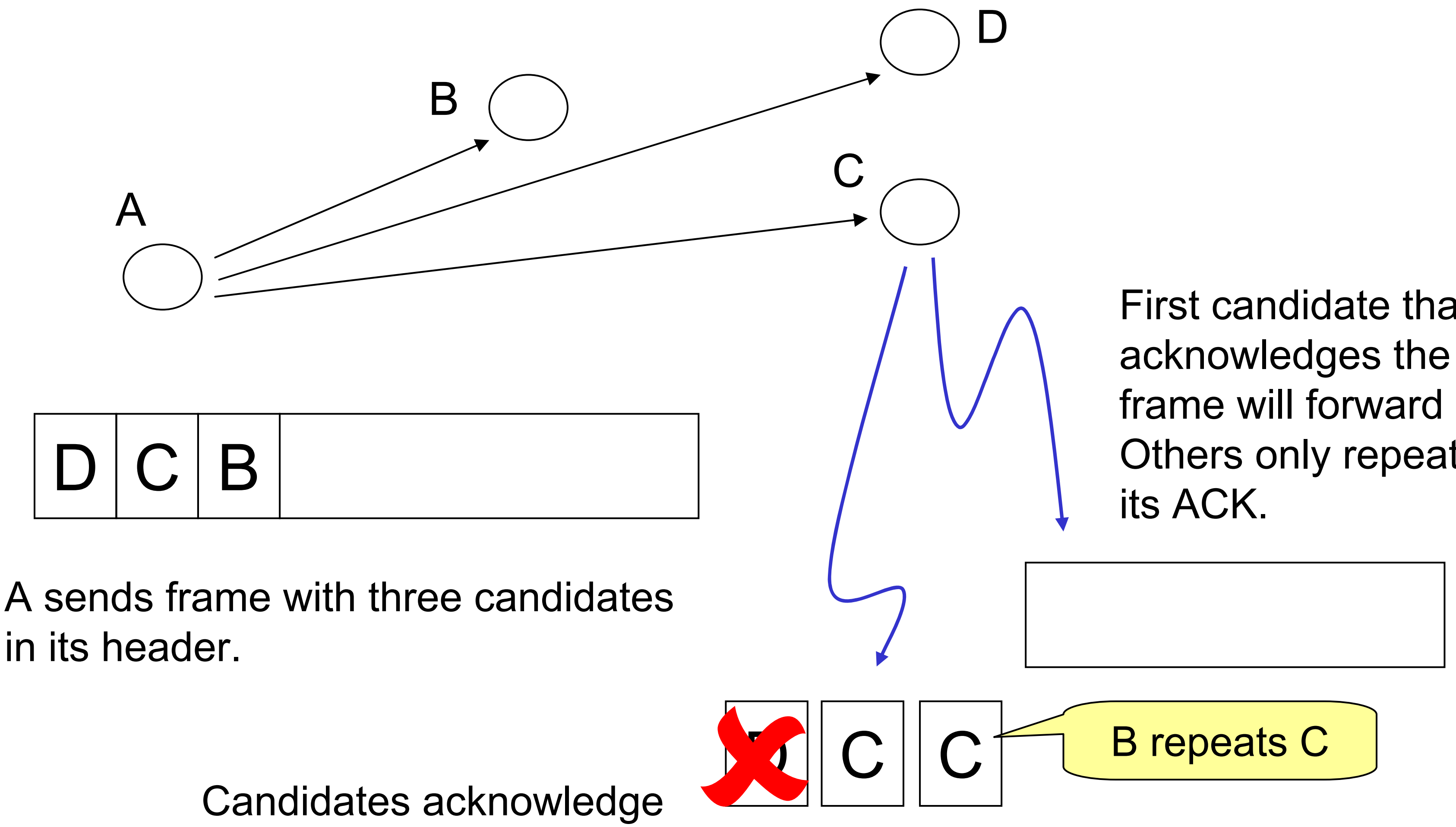
Ad-Hoc Networks &
Sensor Networks

Boone and Bane of Wireless Media

Thomas Fuhrmann

Network Architectures
Computer Science Department
Technical University Munich

Extremely Opportunistic Routing

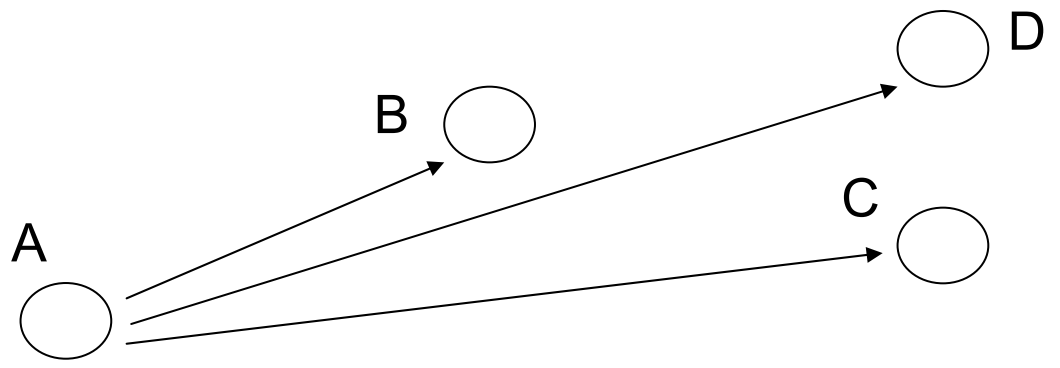


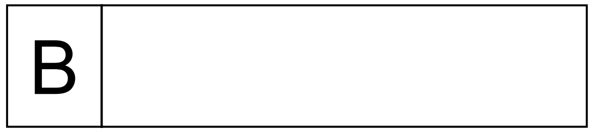
First candidate that acknowledges the frame will forward it. Others only repeat its ACK.

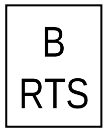
A sends frame with three candidates in its header.

Candidates acknowledge the frame in the order they appeared in the header.

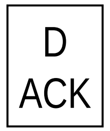
Opportunistic Routing in SSR (1)



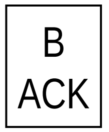




After the data frame, or after a missing RTS, other nodes may ACK the frame too.

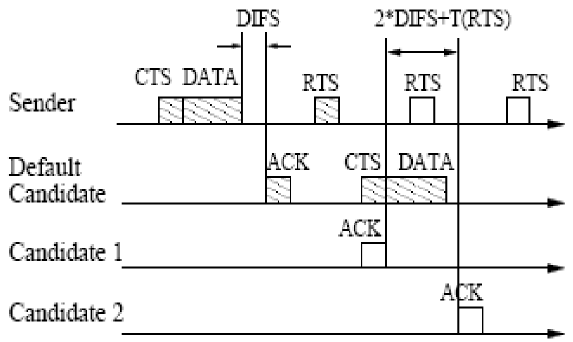


A sends frame to neighbor B because it seems to fit best.

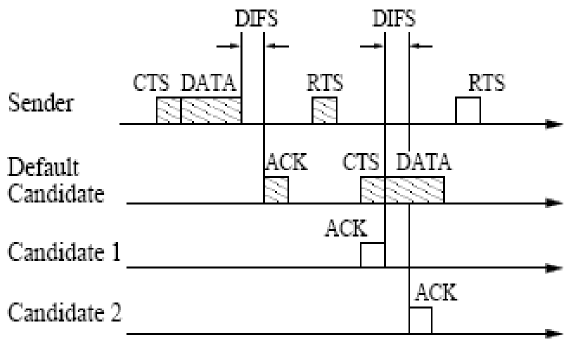


B acknowledges the frame and will forward it after B has requested it to do so.

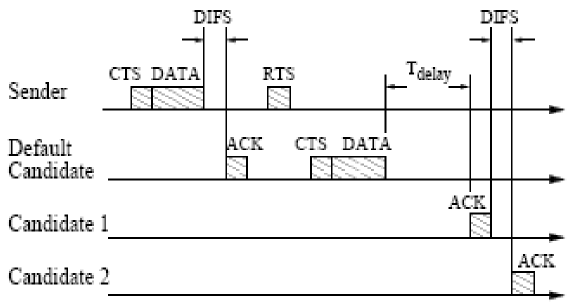
Opportunistic Routing in SSR (2)



(a) The sender always waits for the first ACK. If the default candidate responds, candidate 1 and 2 will cancel their ACKs upon receiving the RTS; otherwise, candidate 1 and 2 will send their ACKs in the predetermined sequence unless they receive an RTS.



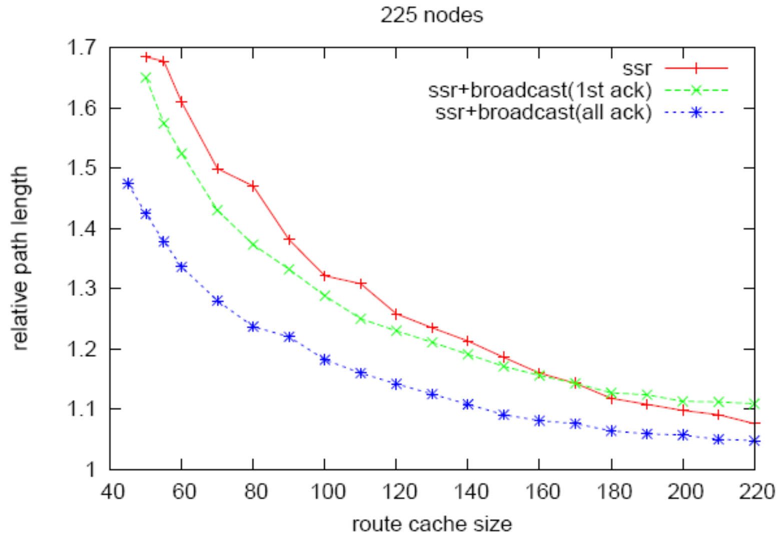
(b) The sender always waits for the first ACK. If the default candidate responds, candidate 1 and 2 will cancel their ACKs upon receiving the RTS; otherwise the sender waits for the ACKs of all candidates before it selects the forwarder.

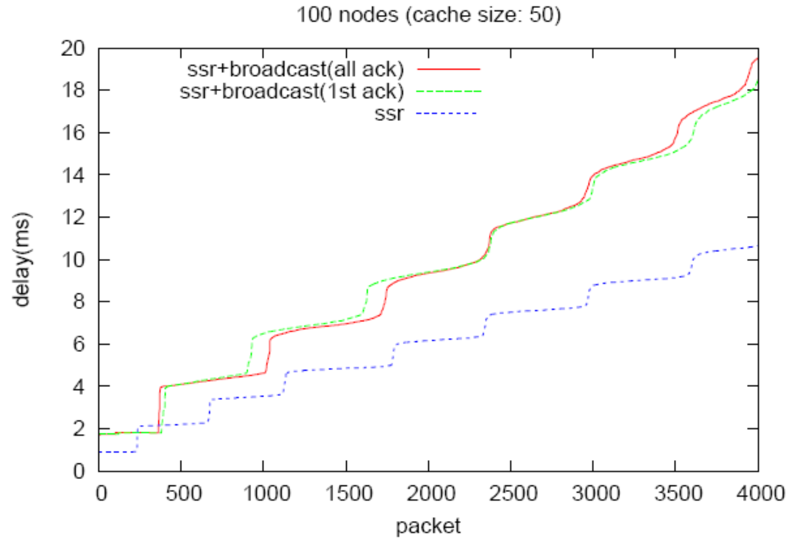


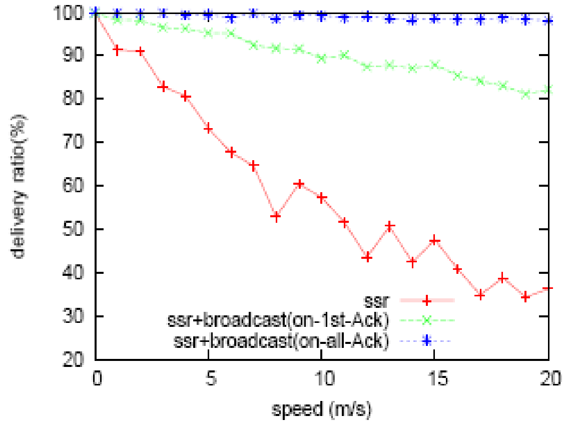
(c) If the default candidate succeeds to respond, other candidates may nevertheless acknowledge to improve future forwarding actions. Note the ACKs are rescheduled with $T_{delay} \geq 3 \cdot T_{DIFS} + T_{ACK} + T_{RTS}$.

Pengfei Di and Thomas Fuhrmann. Using Link-Layer Broadcast to Improve Scalable Source Routing. Proceedings of the Workshop on Mobile Peer-to-Peer Networking, Leipzig, Germany, June 2009

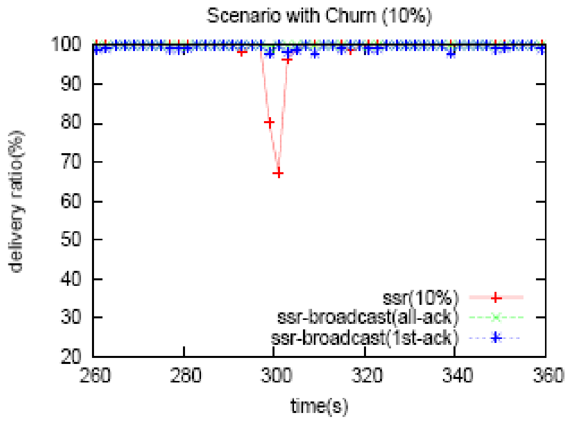
Opportunistic Routing in SSR (3)



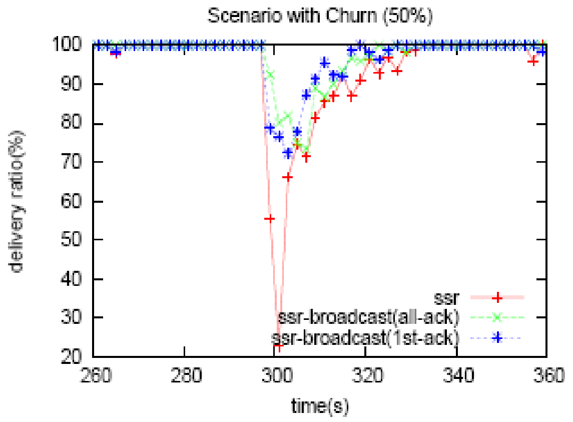




(a) Mobility scenarios



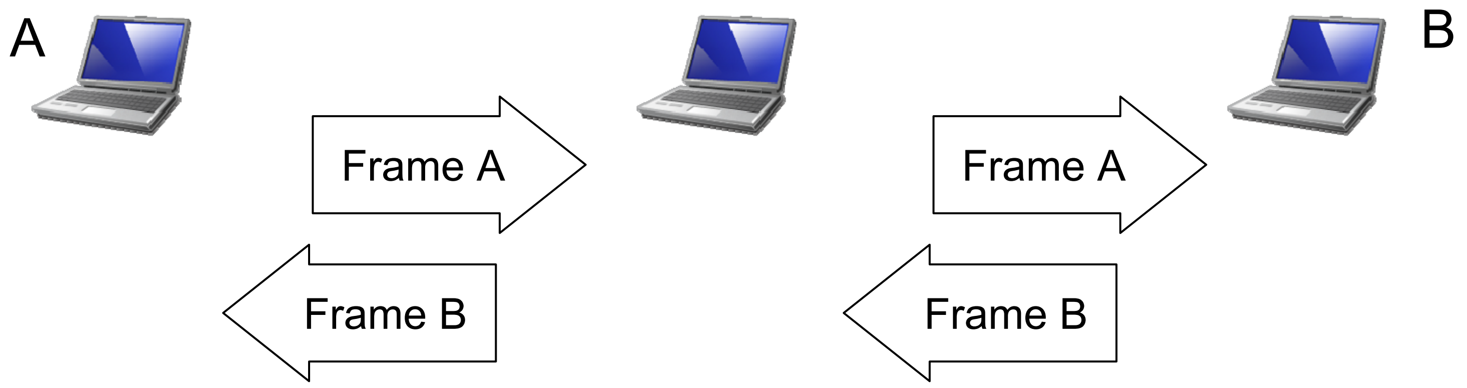
(b) Churn scenarios (churn rate 10%)



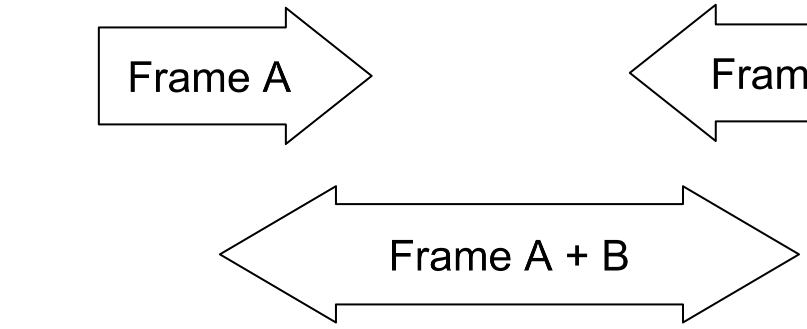
(c) Churn scenarios (churn rate 50%)

Network Coding

Traditional approach:

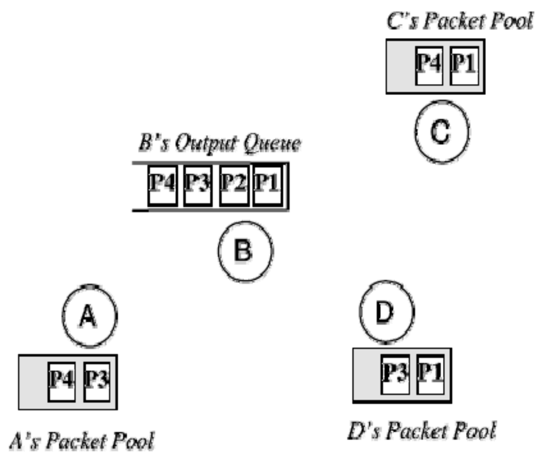


Network coding:

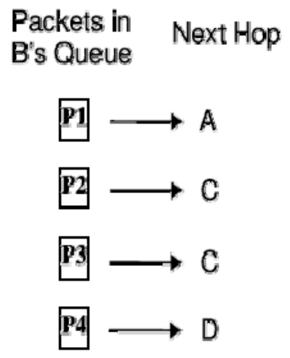


Coding Opportunistically (1)

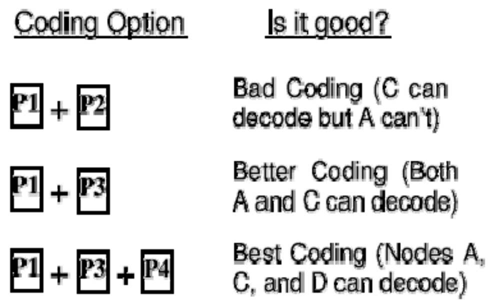
- COPE: Coding Opportunistically [10]
  - Network Coding: Intelligent mixing of frames improves throughput
  - COPE generalizes simple A – B scenario to multiple unicast flows
- Exploit Shared Nature of Wireless Medium
  - Store Overheard Packets for Short Time
  - These packets are used for encoding & decoding subsequent packets
- First real world implementation of Wireless Network Coding



(a) B can code packets it wants to send



(b) Nexthops of packets in B's queue

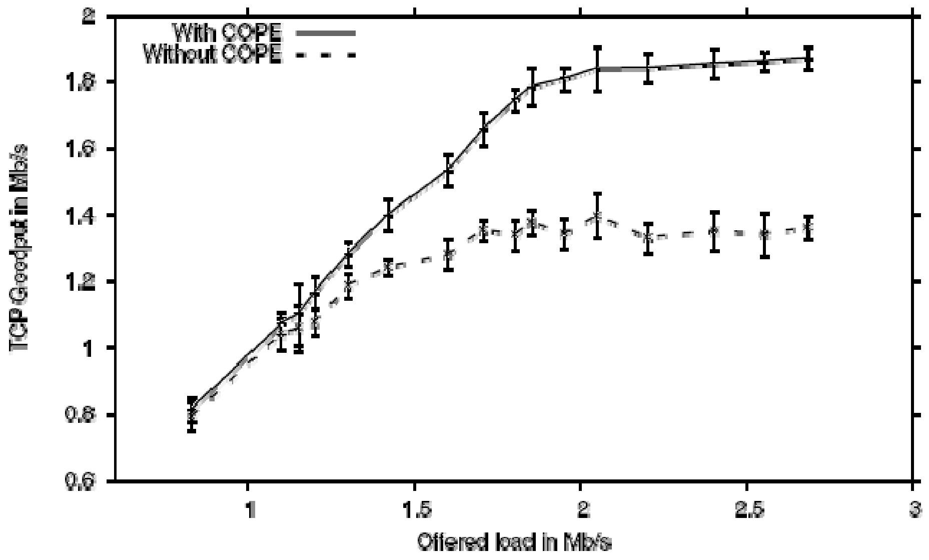


Coding Option
Is it good?
P1 + P2
Bad Coding (C can decode but A can't)
P1 + P3
Better Coding (Both A and C can decode)
P1 + P3 + P4
Best Coding (Nodes A, C, and D can decode)

(c) Possible coding options

Coding Opportunistically (2)

- Each node uses knowledge of what it's neighbors have
- This data allows for source to send XOR'ed packets intelligently: It knows who will be able to decode the encoded packet
- Allows for more than two flows
- Can code multiple packets at the same time
- Gain up to factor 3 -4 when congested with mainly UDP flows
- For mesh networks connected to Internet via access point, the gain depends on total download / upload Traffic at access point
- w/o hidden terminals TCP throughput increases about 38%

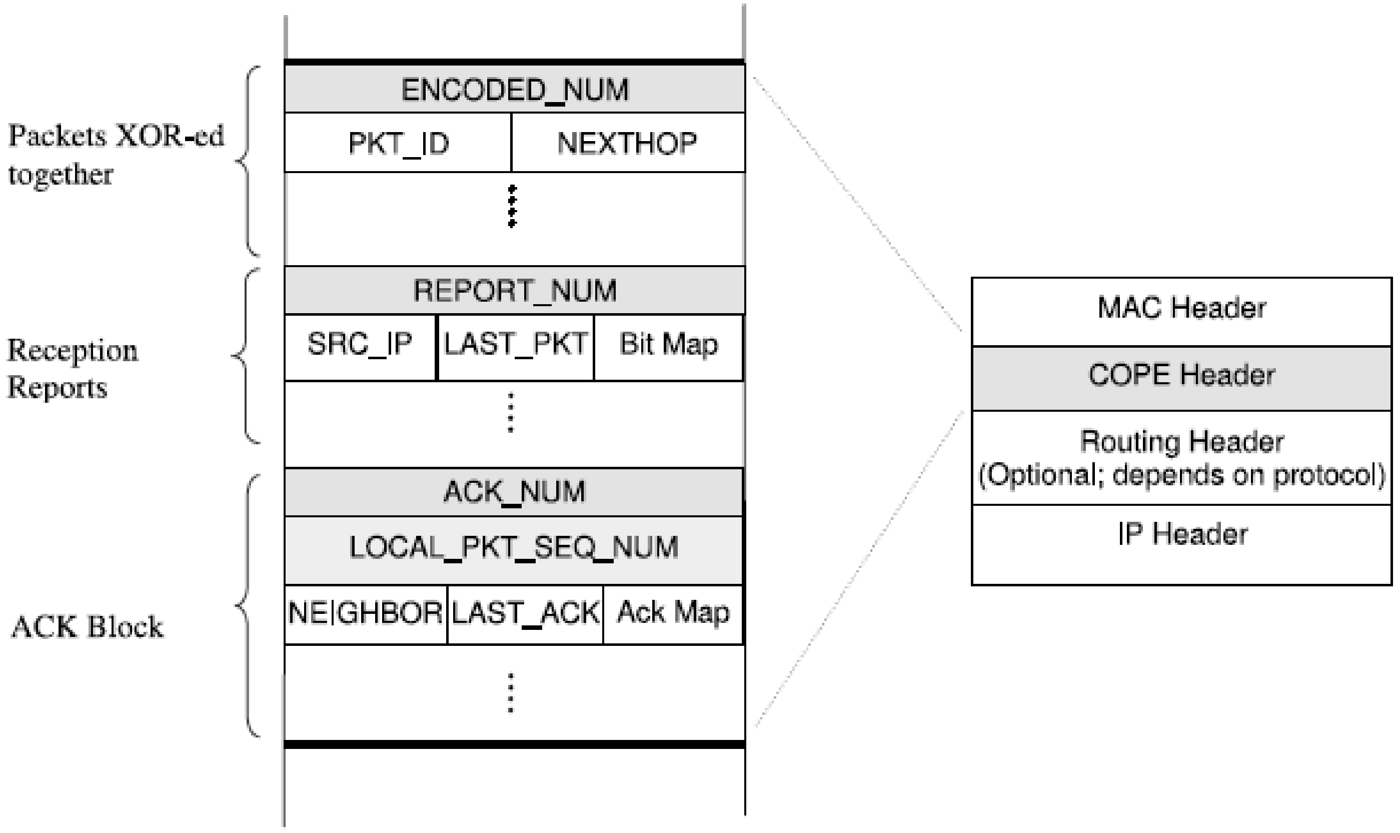


20 nodes with TCP connections, no hidden nodes

Coding Opportunistically (3)

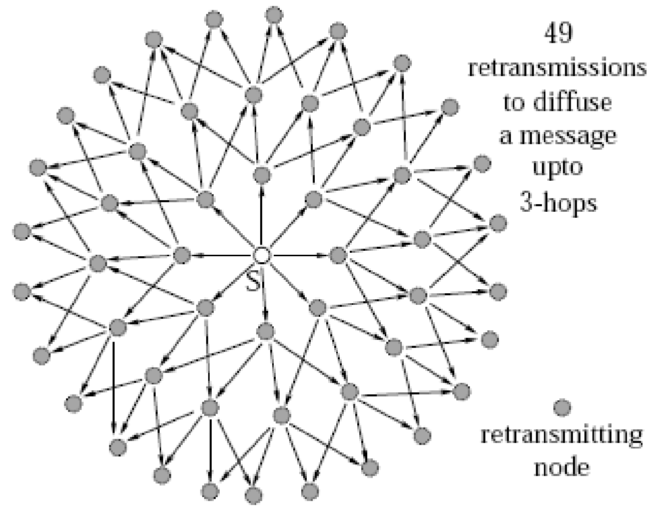
- 3 Main Techniques
  - Opportunistic Listening
  - Opportunistic Coding
  - Learning Neighbor State
- Opportunistic Listening
  - Wireless is a broadcast medium
  - Many chances for nodes to overhear
  - COPE sets all nodes as promiscuous
  - They store overheard packets for a short time, e.g. 500ms
  - Reception reports include sequence number of stored packets
  - These reports are piggy backed onto normal output
- Opportunistic Coding
  - Which packets do we combine to achieve maximum throughput?
  - Simple answer: Send as many (native packets) as possible while ensuring that the next hop has enough information to decode
- Learning neighbor state
  - Each node announces its stored packets in reception reports
  - Sometimes reports don’t get through, e.g. during congestion or in times of light traffic
  - Use educated guess to estimate the probability that a neighbor has a packet, e.g. based on delivery probability

Coding Opportunistically (4)

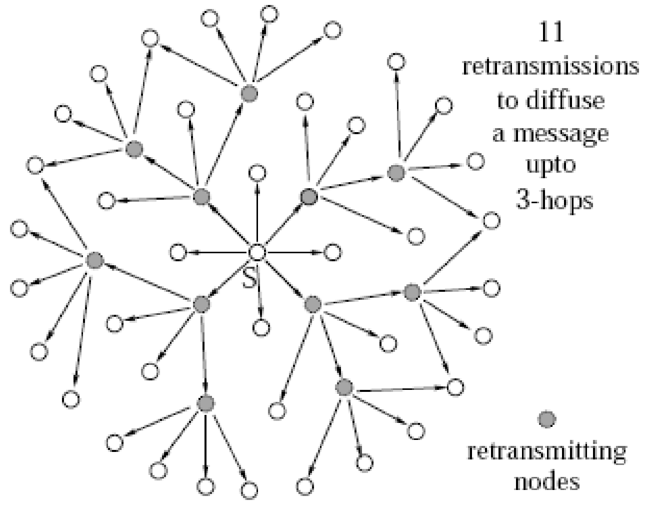


Data dissemination

- Data dissemination has a special traffic pattern
  - Not n to n, but 1 to n or n to 1
  - Example, e.g., the sink in WSN
- Flooding is the easiest way
  - Causing duplication and collision
  - Used in other routing protocols, e.g., AODV, DSR
- Optimizations to flooding
  - Multipoint Relaying (MPR) [11]
  - Gossip [12]



Diffusion of a broadcast message using pure flooding



Diffusion of a broadcast message using multipoint relays

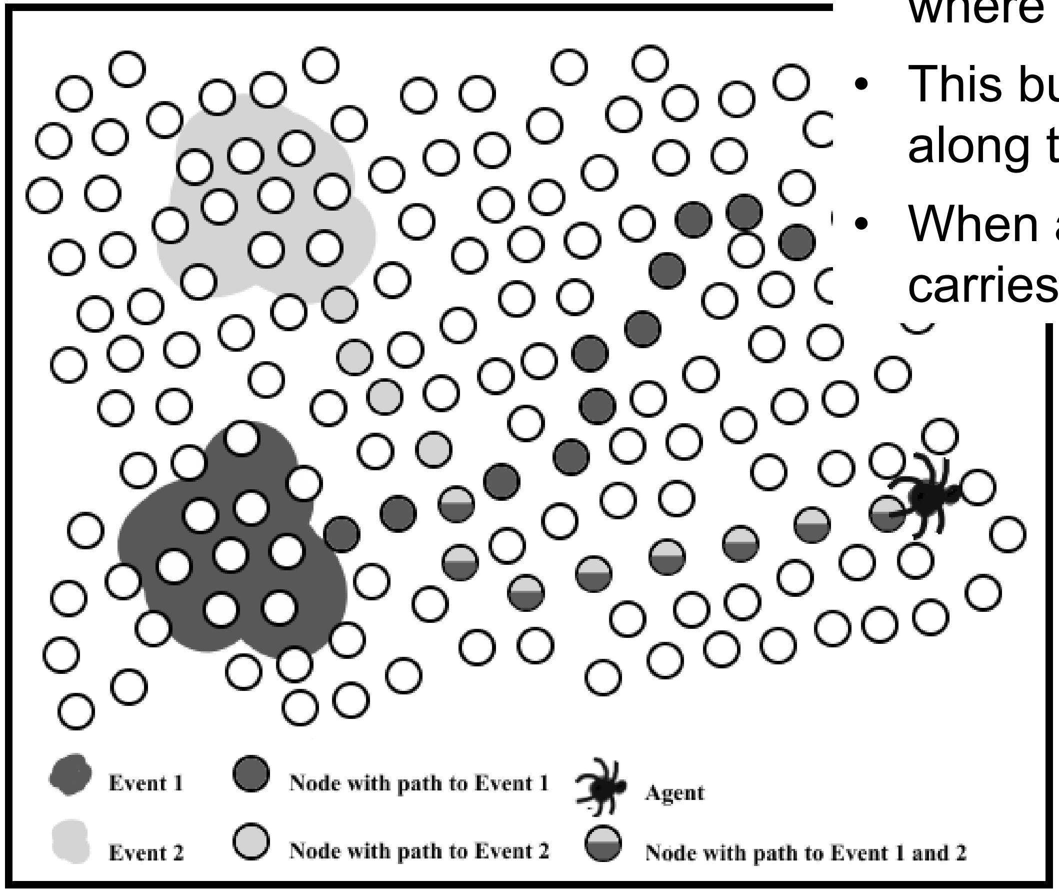
Multipoint Relaying (MPR)

- Definitions
  - neighbors of node x as N(x),
  - the set of its two-hop neighbors as N2(x).
  - Let the selected multipoint relay set of node x be MPR(x).
- Heuristic for the selection of multipoint relays
  - Start with an empty multipoint relay set MPR(x)
  - First select those one-hop neighbor nodes in N(x) as multipoint relays that are the only neighbor of some node in N2(x),
  - add these one-hop neighbor nodes to the multipoint relay set MPR(x)
  - While there still exist some node in N2(x) that is not covered by the multipoint relay set MPR(x):
    - For each node in N(x) which is not in MPR(x), compute the number of nodes that it covers among the uncovered nodes in the set N2(x)
    - Add that node of N(x) in MPR(x) for which this number is maximum.

Gossip(p) and Gossip(p, k)

- Basic idea of Gossip(p):
  - A source sends the packet with probability 1.
  - When a node first receives the packet
    - with probability p it broadcasts the packet to its neighbors
    - with probability 1 − p it discards the packet;
    - called GOSSIP(p).
  - Problems
    - there is a chance that none of them will gossip, and the gossip will die.
- Modified protocol Gossip(p, k):
  - A source and its first k Hops sends the packet with probability 1.
  - The other nodes perform Gossip(p)
  - The parameters (p, k) depend on network topology
  - reduce control traffic up to 35% when compared to flooding
  - the routes found by gossiping may be up to 10-15% longer than those found by flooding

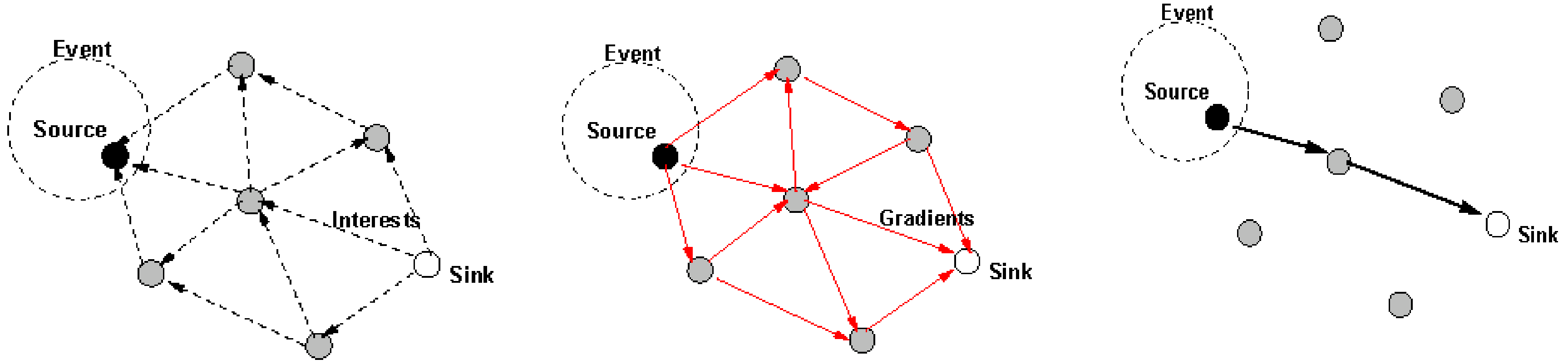
Rumor Routing


Event 1
Node with path to Event 1
Agent
Event 2
Node with path to Event 2
Node with path to Event 1 and 2

Rumor routing

- Agents are sent out from regions where an event is detected
- This builds up state about the event along the path.
- When agent paths cross, the agent carries information about both events.

Directed Diffusion



- ***Naming***
  - *Data* is named using attribute-value pairs
- ***Interests***
  - A node requests data by sending *interests for named data*
- ***Gradients***
  - *Gradients* is set up within the network designed to “draw” events, i.e. data matching the interest.
- ***Reinforcement***
  - Sink *reinforces* particular neighbors to draw higher quality (higher data rate) events

References

[1] Jakob Eriksson, et al., *PeerNet: Pushing Peer-to-Peer Down the Stack* IPTPS 2003

[2] Ni, Lionel M. et al., *SIDA : self-organized ID assignment in wireless sensor networks* IEEE MASS 2007. Pisa, 8-11 October 2007

[3] Douglas S. et al., *A high-throughput path metric for multi-hop wireless routing.* ACM MobiCom '03) , San Diego, California, 2003

[4] Richard Draves. et al., *multi-hop wireless mesh networks.* In MobiCom '04: New York, NY, USA, 2004.

[5] Jun Wang, et al., *Interference-aware load balancing for multihop wireless networks*. Technical report,University of Illinois at Urbana-Champaign, 2005.

[6] Weirong Jiang, et al., *Optimizing routing metrics for large-scale multi-radio mesh networks*. In Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007., Shanghai, China, 2007.

[7] Anand Prabhu Subramanian, et al., *Interference aware routing in multi-radio wireless mesh networks*. In Proceedings of the 2nd IEEE Workshop on Wireless Mesh Networks, Reston, VA, USA, 2006

[8] Liang Ma, et al., *A routing metric for load-balancing in wireless mesh networks*. In AINAW '07: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops, Washington, DC, USA, 2007

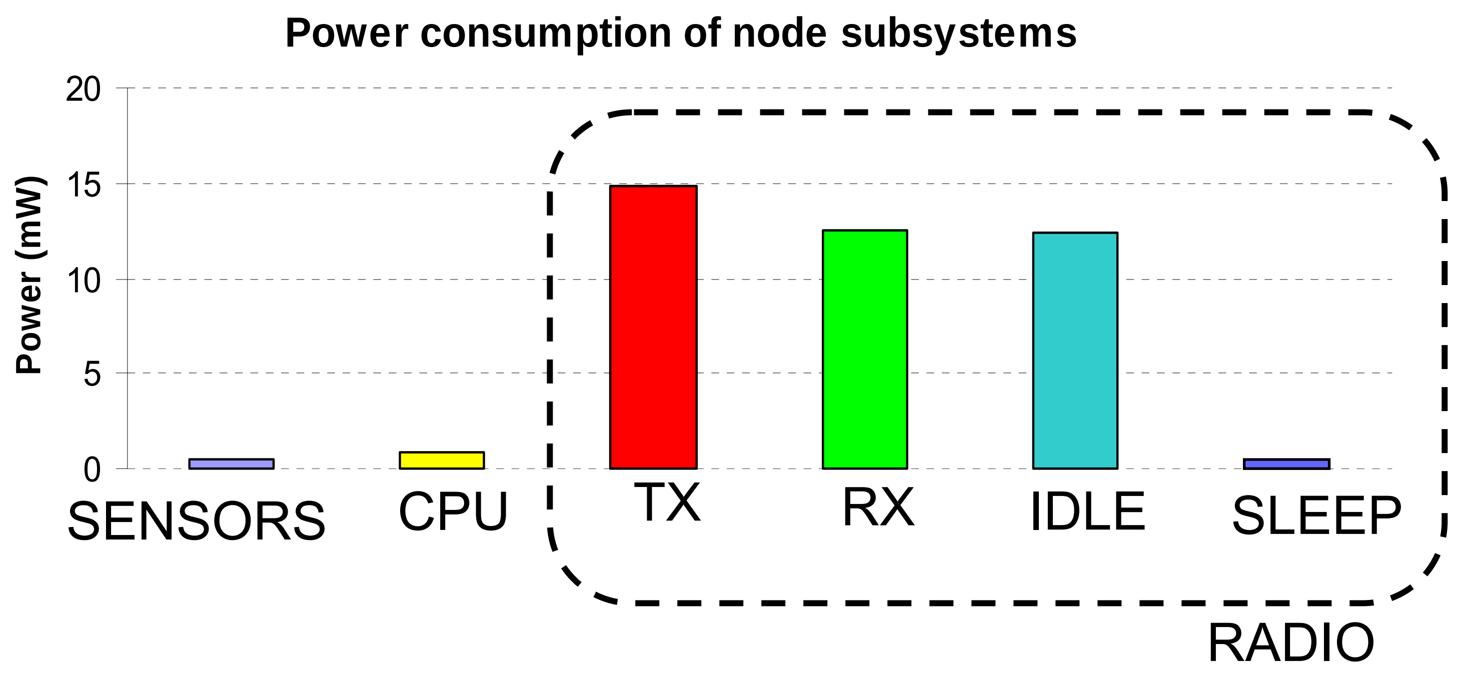
[9] Sanjit Biswas, et al., *Opportunistic routing in multi-hop wireless networks,* ACM SIGCOMM Computer Communication Review, Volume 34 , Issue 1, January 2004

[10] Sachin Katti, et al., XORs in the air: practical wireless network coding, IEEE/ACM Transactions on Networking (TON) Volume 16 , Issue 3, June 2008

[11] Amir Qayyum, et al., *Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks, P*roceeding of the 35th annual Hawaii International Conference on System Sciences, 2002

[12] Zygmunt J. Haas, et al., Gossip-based ad hoc routing, IEEE/ACM Transactions on Networking (TON), Volume 14 , Issue 3 June 2006

Identifying the Energy Consumers



- Simple sensor node spend almost all their energy on the radio incl. Waiting for the radio.
- Optimizing node life-time thus means optimizing the radio, esp. media access.

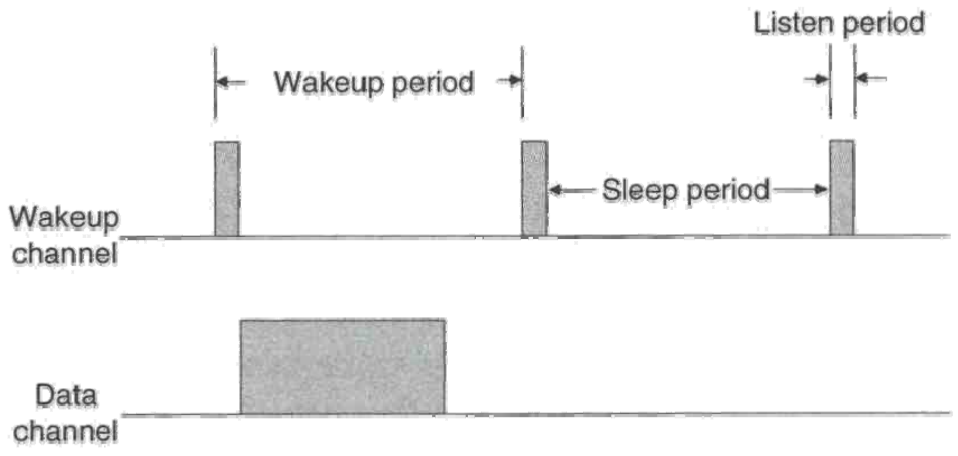
Energy Efficiency in MAC

- Major sources of energy waste
  - Idle listening
    - Long idle time when no sensing event happens
    - Collisions
    - Control overhead
    - Overhearing

  Common to all wireless networks
- Try to reduce energy consumption from all above sources
- TDMA requires slot allocation and time synchronization
- Combine benefits of TDMA + contention protocols

Sparse Topology & Energy Management Protocol

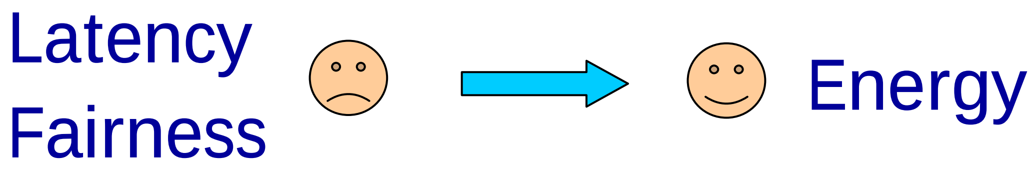
- Two channels
  - Announce data on the wakeup channel
  - Then send data on the data channel
  - Otherwise the data channel is in sleep mode
- Status of a sensor
  - Monitor state, i.e. nodes are idle, no transmission
  - Transfer state
- STEM-B
  - Transmitter wakes up the receiver using a beacon on the wakeup channel
  - no RTS/CTS
- STEM-T
  - Transmitter sends busy tone signal on the wakeup channel to get the receiver‘s attention



Slides adapted from Holger Karl and Anreas Willig.

Sensor-MAC (S-MAC) Design

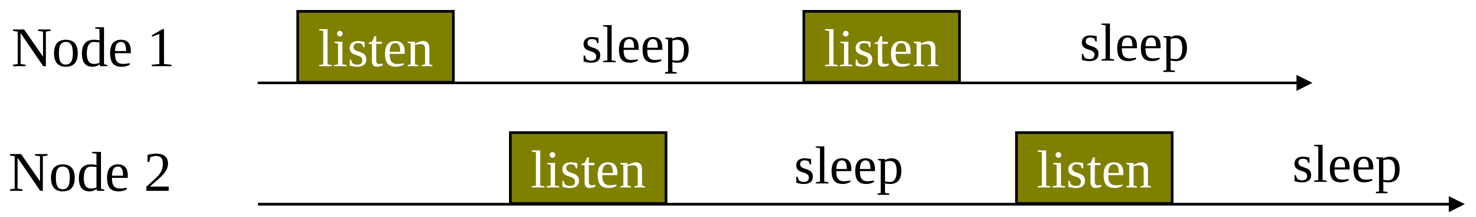
- Tradeoffs

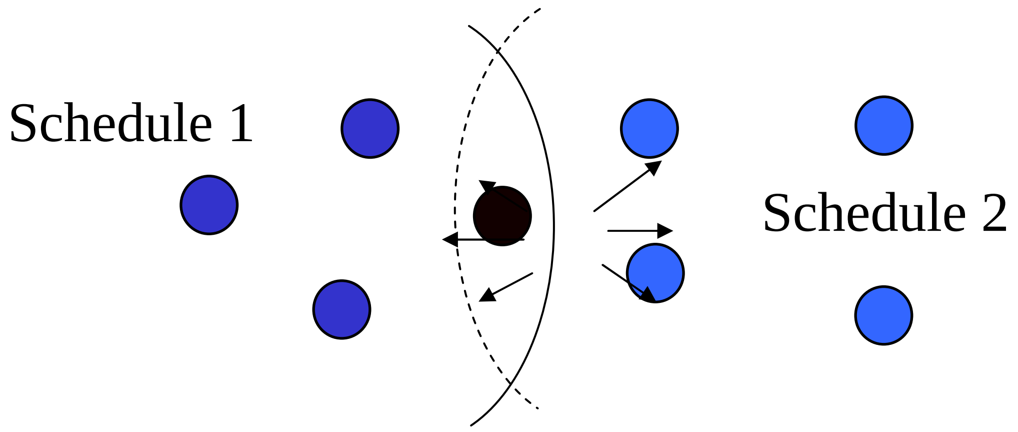


- Major components of S-MAC (Wei et al. 2002)
  - Periodic listen and sleep
  - Collision avoidance
  - Overhearing avoidance
  - Message passing

Periodic Listen and Sleep

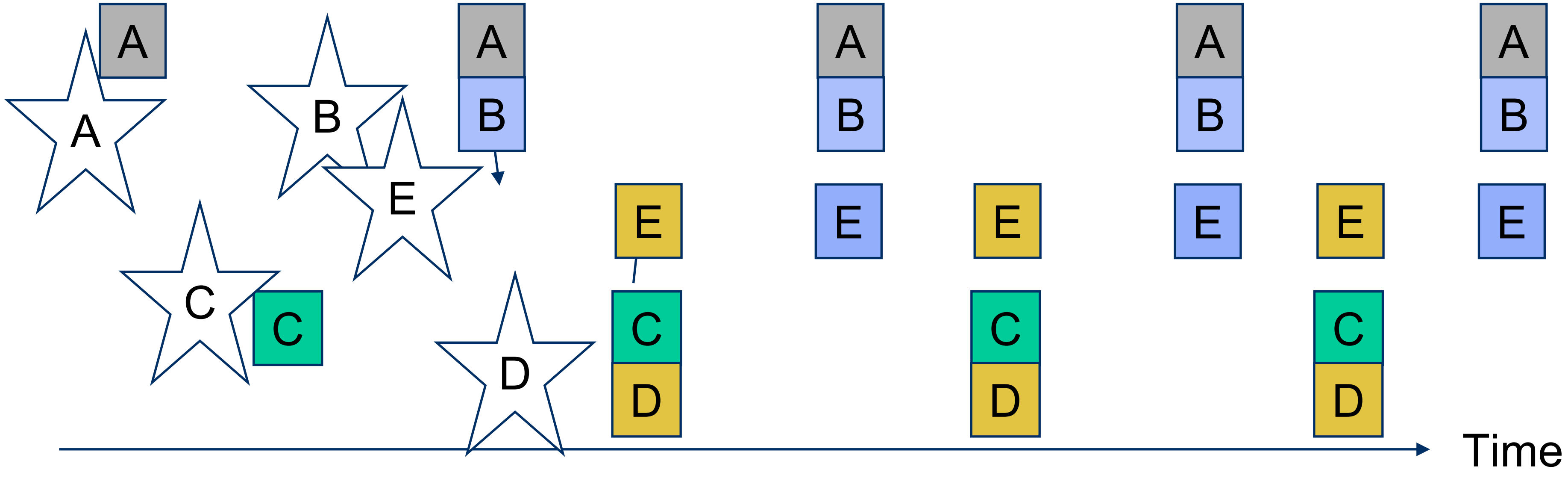
Preferable if neighboring nodes have same schedule — easy broadcast & low control overhead. But routers at boundaries must obey two schedules.





- Schedule maintenance
  - Remember neighbors' schedules to know when to send to them
  - Each node broadcasts its schedule every few periods
  - Refresh on neighbor's schedule when receiving an update
  - Schedule packets also serve as beacons for new nodes to join a neighborhood

Periodic Listen and Sleep



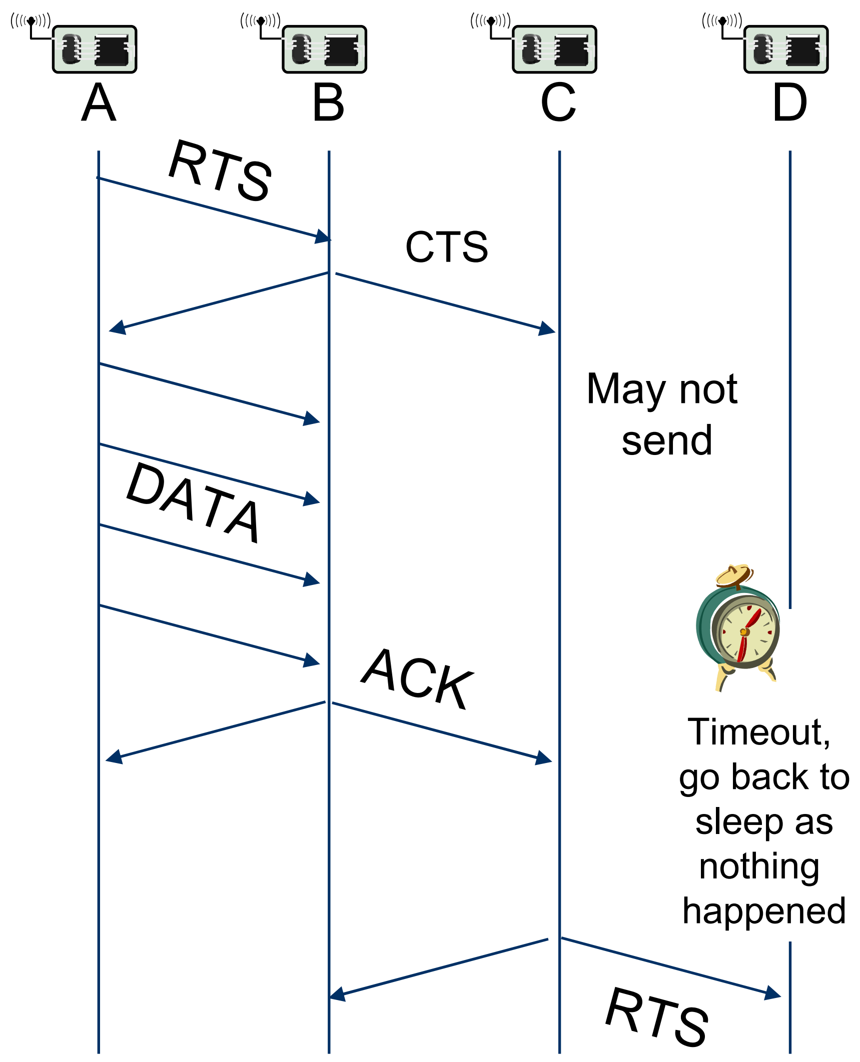
Collision & Overhearing Avoidance

- Problem:
  Multiple senders want to talk
- Options: Contention vs. TDMA
- Solution: Similar to IEEE 802.11 ad hoc mode (DCF)
  - Physical and virtual carrier sense
  - Randomized backoff time
  - RTS/CTS for hidden terminal problem
  - RTS/CTS/DATA/ACK sequence

- Problem:
  Receive packets destined to others
- Solution: Sleep when neighbors talk
  - Basic idea from PAMAS (Singh 1998)
  - But we only use in-channel signaling
- Who should sleep?
  - All immediate neighbors of sender and receiver
- How long to sleep?
  - The *duration* field in each packet informs other nodes the sleep interval

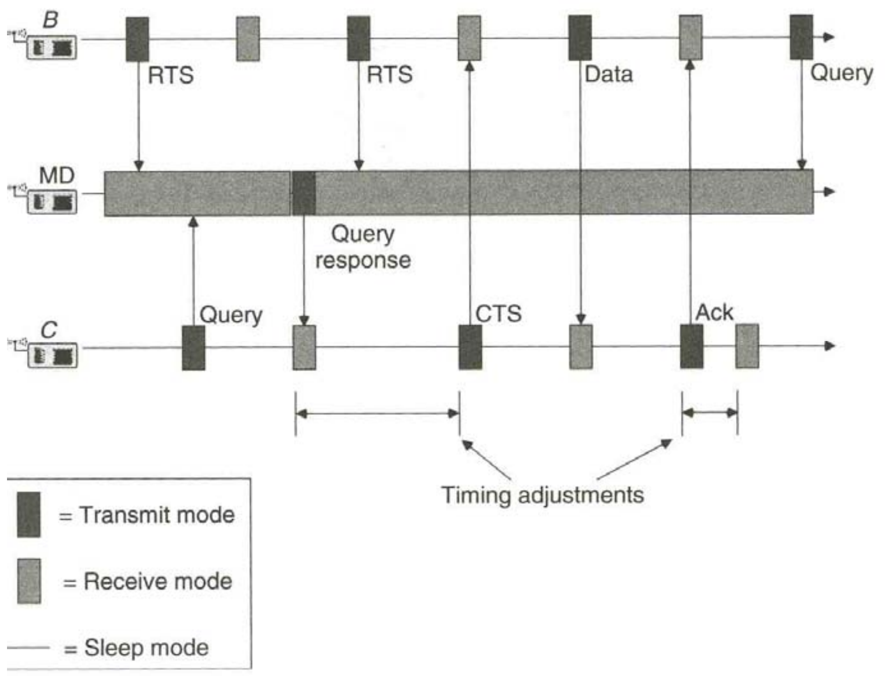
Timeout-MAC (T-MAC)

- In S-MAC, active period is of constant length
- What if no traffic actually happens?
  - Nodes stay awake needlessly long
- Idea:
  - Adaptive duty cycle!
  - Prematurely go back to sleep mode when no traffic has happened for a certain time (=timeout)!
- Remaining problem: Early sleeping
  - C wants to send to D, but is hindered by transmission A! B



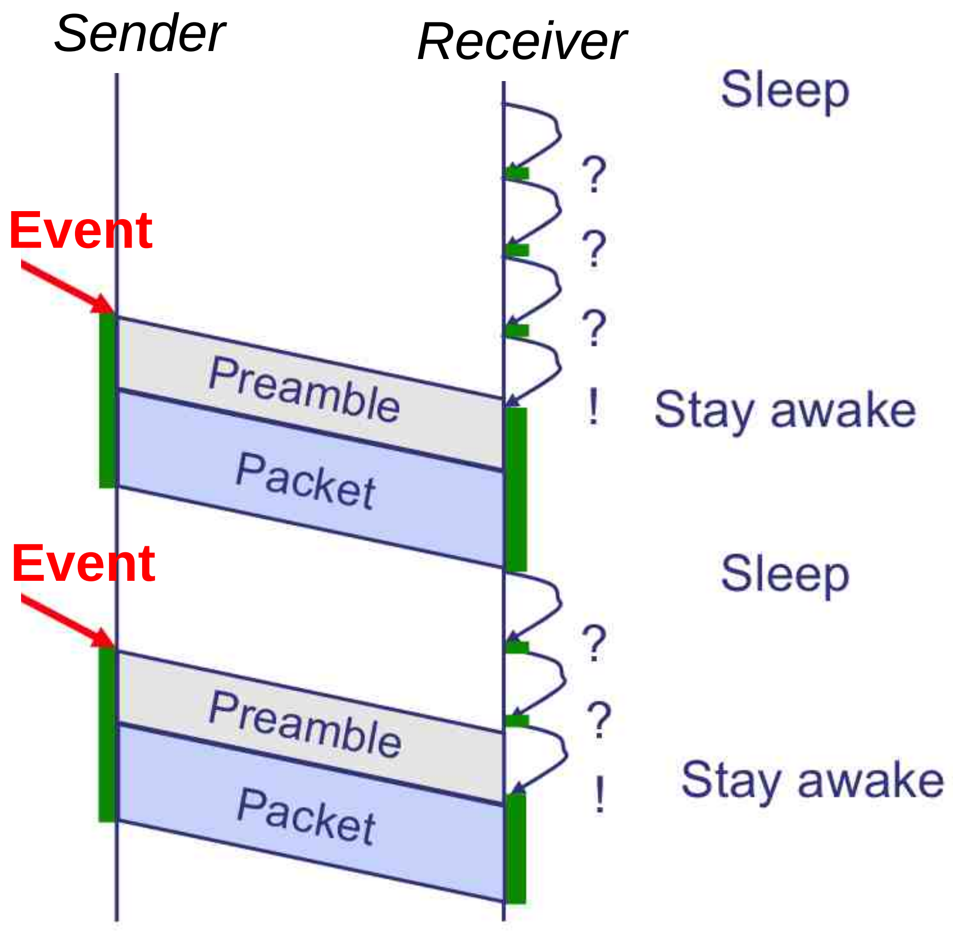
Mediation Device Protocol

- Goal: Avoid useless listening on the channel for messages
- Uses: mediation device (MD) which is available all the time
- Protocol
  - Sender B sends RTS to MD
  - MD stores this information
  - Receiver C sends query to MD
  - MD tells reciever C when to wake up
  - C sends CTS to B (now in sync)
  - B sends data
  - C acknowledges
  - C returns to old timing
- Main disadvantage:
  - MD has to be energy independent
  - Solution: Distributed Mediation Device Protocol
    - Nodes randomly wake up and serve as mediation device
- Problem: no guarantees on full coverage of MD



Preamble Sampling

- So far: Periodic sleeping supported by some means to synchronize wake up of nodes to ensure rendez-vous between sender and receiver
- Alternative option: Don’t try to explicitly synchronize nodes
  - Have receiver sleep and only periodically sample the channel
- Use ***long preambles*** to ensure that receiver stays awake to catch actual packet
- Example: WiseMAC



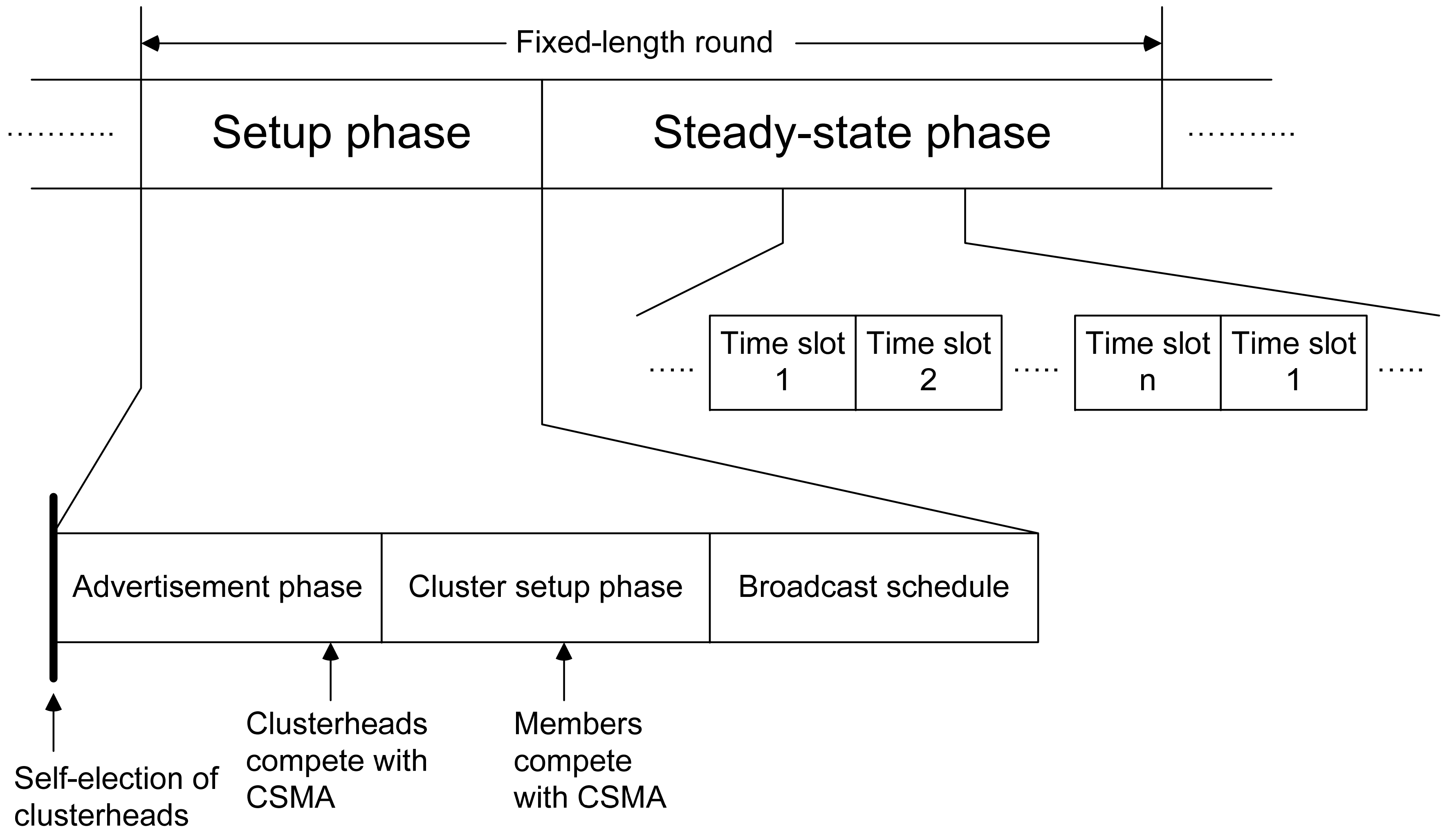
Berkeley MAC (B-MAC)

- Combines several of the above discussed ideas
  - Takes care to provide practically relevant solutions
- Clear Channel Assessment
  - Adapts to noise floor by sampling channel when it is assumed to be free
  - Samples are exponentially averaged, result used in gain control
  - For actual assessment when sending a packet, look at five channel samples – channel is free if even a single one of them is significantly below noise
  - Optional: random backoff if channel is found busy
- Optional: Immediate link layer acknowledgements for received packets
- Low Power Listening (= preamble sampling)
  - Uses the clear channel assessment techniques to decide whether there is a packet arriving when node wakes up
  - Timeout puts node back to sleep if no packet arrived
- B-MAC does *not* have
  - Synchronization
  - RTS/CTS
  - Results in simpler, leaner implementation
  - Clean and simple interface
- Often considered as the default WSN MAC protocol

Low-Energy Adaptive Clustering Hierarchy (LEACH)

- Given: dense network of nodes, reporting to a central sink, each node can reach sink directly
- Idea: Group nodes into "***clusters***", controlled by ***clusterhead***
  - About 5% of nodes become clusterhead (depends on scenario)
  - Role of clusterhead is rotated to share the burden
  - Clusterheads advertise themselves, ordinary nodes join CH with strongest signal
  - Clusterheads organize
    - CDMA code for all member transmissions
    - TDMA schedule to be used within a cluster
- In steady state operation
  - CHs collect & aggregate data from all cluster members
  - Report aggregated data to sink using CDMA

LEACH – Overview



Traffic Adaptive Medium Access Protocol (TRAMA)

- Nodes are synchronized
- Time divided into cycles, divided into
  - Random access periods
  - Scheduled access periods
- Nodes exchange neighborhood information
  - Learning about their two-hop neighborhood
  - Using ***neighborhood exchange protocol***: In random access period, send small, incremental neighborhood update information in randomly selected time slots
- Nodes exchange schedules
  - Using ***schedule exchange protocol***
  - Similar to neighborhood exchange
- Adaptive Election Protocol
  - Elect transmitter, receiver and stand-by nodes for each transmission slot
  - Remove nodes without traffic from election

TRAMA – adaptive election

- Given: Each node knows its two-hop neighborhood and their current schedules
- How to decide which slot (in scheduled access period) a node can use?
  - Use ***node identifier*** x and globally known ***hash function*** h
  - For time slot t, compute ***priority*** p = h (x, t)
  - Compute this priority for next k time slots for node itself and all two-hop neighbors
  - Node uses those time slots for which it has the highest priority

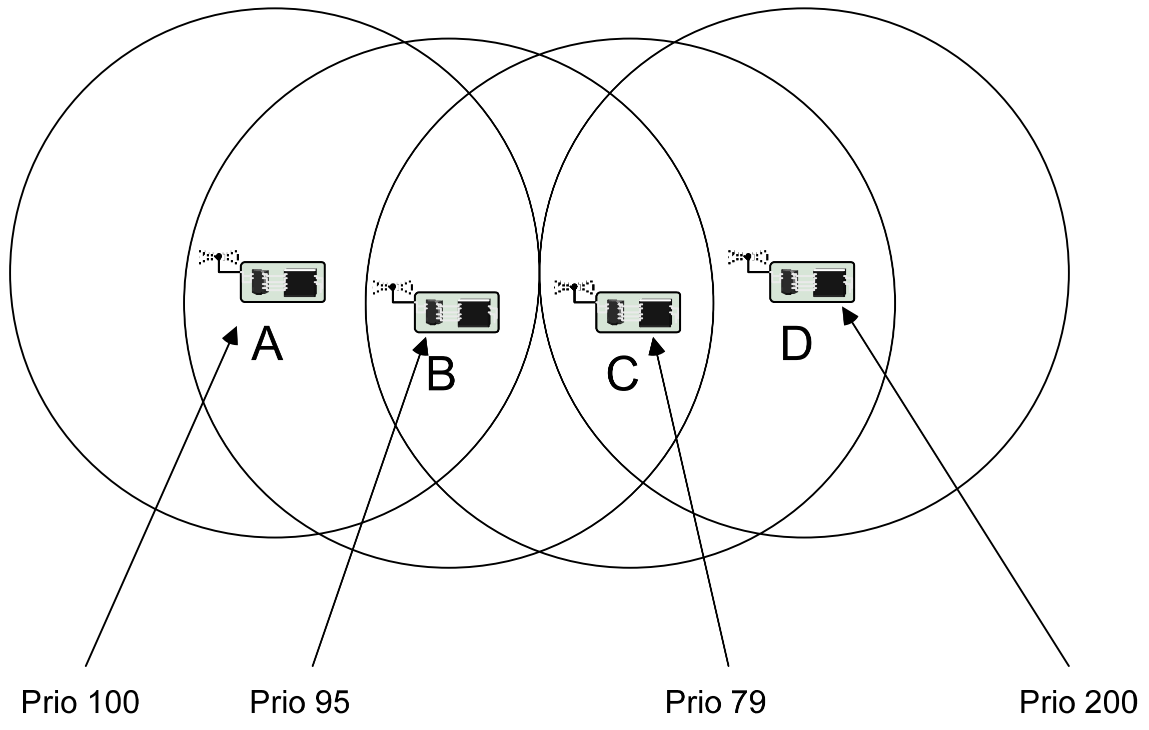
Priorities of node A and its two neighbors B & C

	t = 0	t = 1	t = 2	t=3	t = 4	t = 5
A	14	23	9	56	3	26
B	33	64	8	12	44	6
C	53	18	6	33	57	2

TRAMA – possible conflicts

- When does a node have to receive?
  - Easy case: one-hop neighbor has won a time slot and announced a packet for it
  - But complications exist – compare example

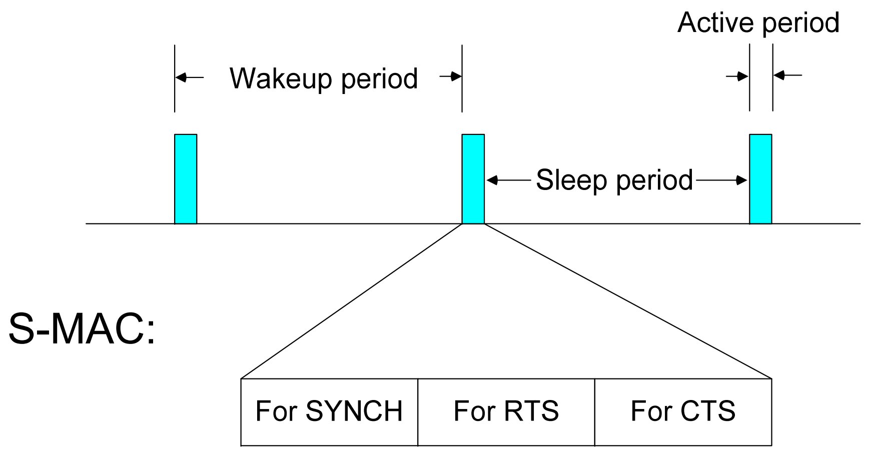
- What does B believe?
  - A thinks it can send
  - B knows that D has higher priority in its 2-hop neighborhood!
- Rules for resolving such conflicts are part of TRAMA



Comparison: TRAMA versus S-MAC

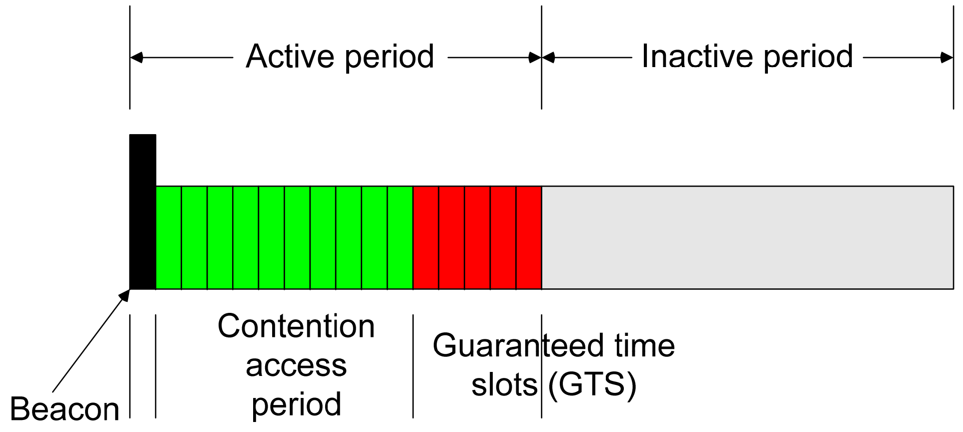
- Comparison between TRAMA & S-MAC
  - Energy savings in TRAMA depend on load situation
  - Energy savings in S-MAC depend on duty cycle
  - TRAMA (as typical for a TDMA scheme) has higher delay but higher maximum throughput than contention-based S-MAC

- TRAMA disadvantage: substantial memory/CPU requirements for schedule computation



Clock Synchronization

- Example: IEEE 802.15.4 Zigbee
- Star networks: ***devices*** are associated with ***coordinators***
  - Forming a PAN, identified by a PAN identifier
- MAC protocol
  - Single channel at any one time
  - Combines contention-based and schedule-based schemes
- Beacon-mode superframe structure
  - Guaranteed time slots (GTS) assigned to devices upon request



The role of time in WSNs

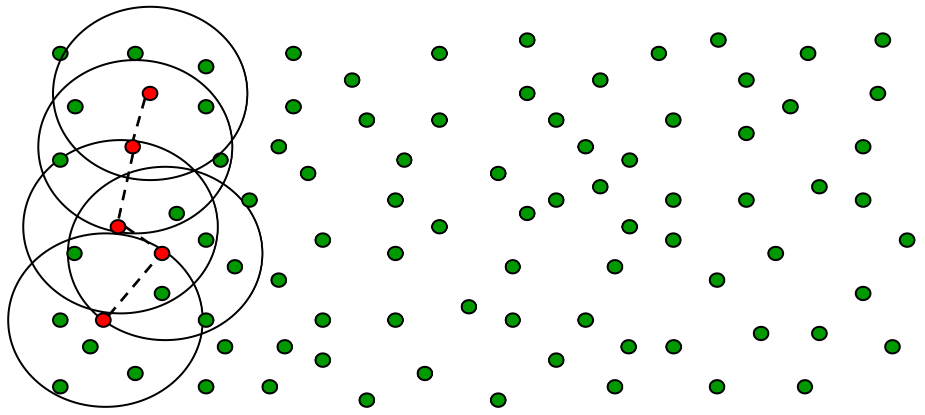
- WSN have a direct coupling to the physical world,
  - notion of time should be related to ***physical time***:
- physical time = wall clock time, real-time
  - one second of a WSN clock should be close to one second of real time
- Commonly agreed time scale for real time is UTC
  - Coordinated Universal Time
  - generated from atomic clocks
  - modified by insertion of leap seconds to keep in synch with astronomical timescales (one rotation of earth)
- Universal Time (UT)
  - timescale based on the rotation of earth
- Other concept: logical time (Lamport)
  - relative ordering of events counts but not their relation to real time

Properties of Synchronization Mechanisms

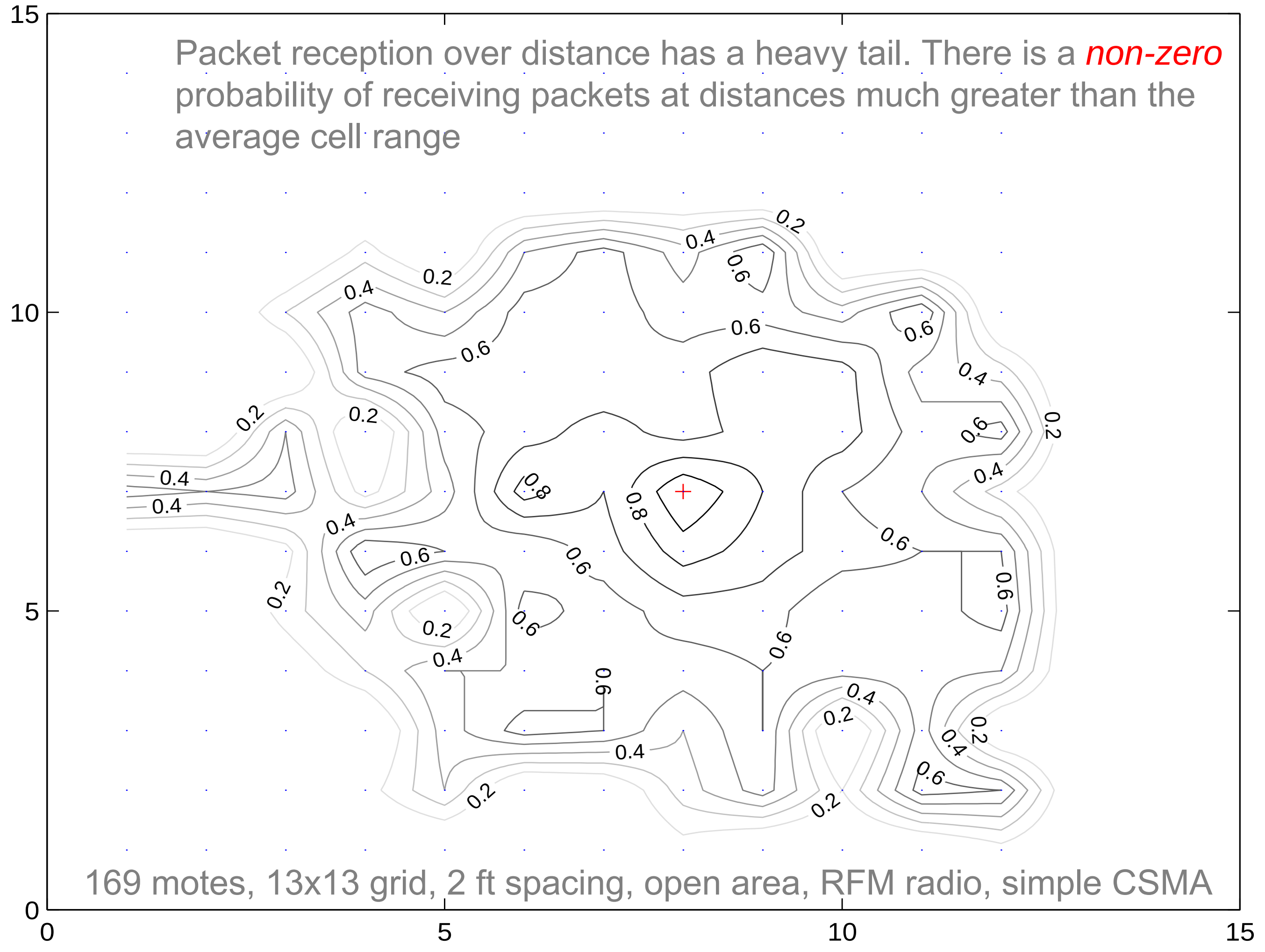
- Physical time versus logical time
- External versus internal synchronization
  - External synchronization with external real time scale like UTC as provided by an outside station
  - If no external timescale available, nodes must agree on common time
- Global versus local algorithms
  - Keep all nodes of a WSN synchronized or only a local neighborhood?
- Absolute versus relative time
- Hardware versus software-based mechanisms
  - GPS, Galileo, GLONASS, DCF77 receiver would be a hardware solution
  - but too heavyweight, costly, and energy-consuming in WSN nodes
  - line-of-sight to at least four satellites is required
- A-priori vs. a-posteriori synchronization
  - Is time synchronization achieved before or after an interesting event? (cf. post-facto synchronization)
- Deterministic vs. stochastic precision bounds
- Local clock update discipline
  - Should backward jumps of local clocks be avoided? (cf. version control)
  - Avoid sudden jumps

Adaptive Topology

- Can we do more than shut down radio in between transmissions and receptions?
- Can we put nodes to sleep for longer periods of time?
- Goal:
  - Exploit high density (over) deployment to extend system lifetime
  - Provide topology that adapts to the application needs
  - Self-configuring system that adapts to environment without manual configuration
- Simple Formulation (Geometric Disk Covering)
  - Given a distribution of ***N*** nodes in a plane.
  - Place a ***minimum*** number of disks of radius ***r*** (centered on the nodes) to cover them.
  - Disk represents the radio connectivity (***simple*** circle model).
- The problem is NP-hard.



Connectivity Measurements



Can't just determine connectivity clusters thru geographic coordinates…

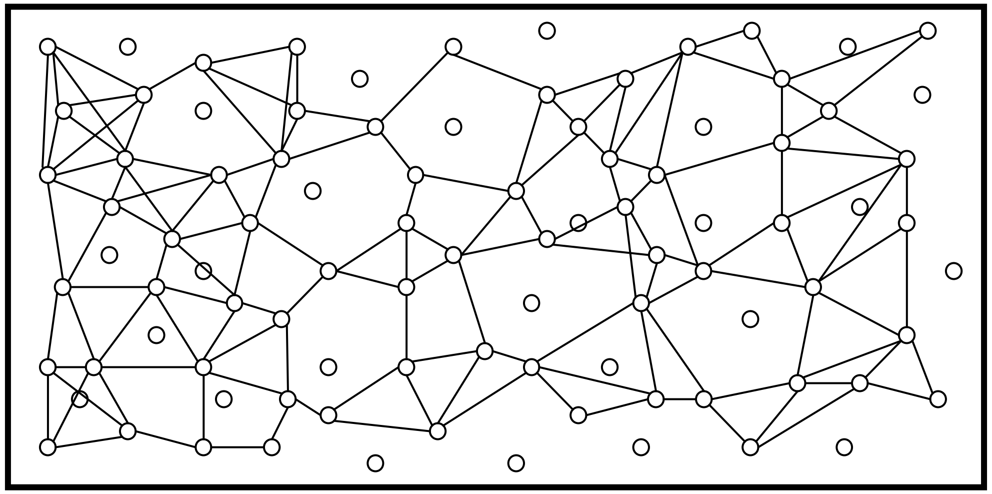
For the same reason you can't determine coordinates w/connectivity

An Empirical Study of Epidemic Algorithms in Large Scale Multihop Wireless Networks
Ganesan, Krishnamachari, Woo, Culler, Estrin and Wicker, UCLA/CSD-TR 02-0013.

Tradeoff

How many nodes to activate?

- ***few*** active nodes:
  - distance between neighboring nodes high
  - increase ***packet loss*** *and* higher ***transmit power and reduced spatial reuse;***
  - need to maintain sensing coverage (see earlier session on coverage/exposure)
- ***too many*** active nodes:
  - at best, ***expending unnecessary energy;***
  - at worst nodes may ***interfere*** with one another by ***congesting*** the channel.



In-Network Processing

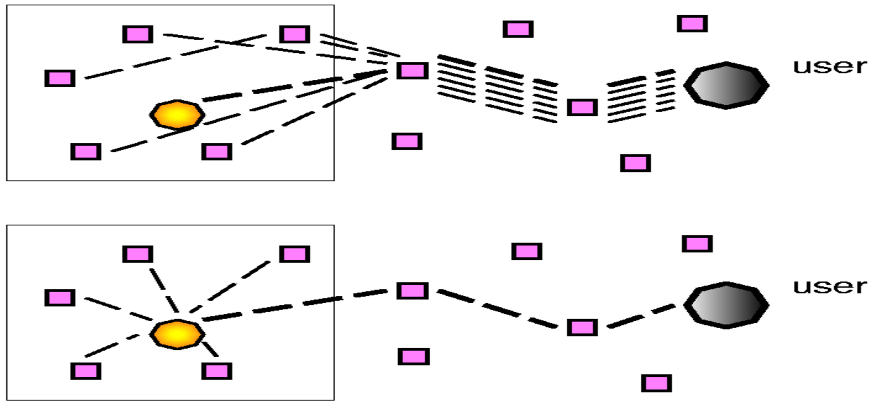
- In-Network Processing is often assume the key to Sensor Network scalability
- Gupta and Kumar pointed out fundamental limits of large scale wireless networks (per node throughput O(1/sqrtN)
- However, densely deployed sensor network data will be correlated and can be aggregated
- Scalability and lifetime will thus also depend on techniques for in-network processing of data
  - Naming Data: Directed Diffusion
  - Data base perspectives:TAG, Sylph
  - Programming mechanisms: Sensorware, Mate

Directed Diffusion: Data Centric Routing

- Basic idea
  - ***name data*** (not nodes) with externally relevant attributes
    - Data type, time, location of node, SNR, etc
  - diffuse requests and responses across network using application driven routing (e.g., geo sensitive or not)
  - optimize path with gradient-based feedback
  - ***support in-network aggregation*** and processing
- Data sources publish data, Data clients subscribe to data
  - However, all nodes may play both roles
    - A node that aggregates/combines/processes incoming sensor node data becomes a source of new data
    - A sensor node that only publishes when a combination of conditions arise, is a client for the triggering event data
  - T*rue* ***peer to peer system***
- Implementation defines ***namespace and simple matching rules*** with filters
  - Linux (32 bit proc) and TinyOS (8 bit proc) implementations

Diffusion as a construct for in-network processing

- Nodes pull, push, and store named data (using tuple space) to create efficient processing points in the network
  - e.g. duplicate suppression, aggregation, correlation
- Nested queries reduce overhead relative to “edge processing”
- Complex queries support collaborative signal processing
  - propagate function describing desired locations/nodes/data (e.g. ellipse for tracking)
  - Interesting analogs to emerging peer-to-peer architectures
- Build on a data-centric architecture for queries and storage



Data Centric vs. Address Centric

- Address Centric
  - Distinct paths from each source to sink.
- Data Centric
  - Support aggregation in the network where paths/trees overlap
  - Essential difference from traditional IP networking
- Building efficient trees for Data centric model
  - Aggregation tree: On a *general* graph if k nodes are sources and one is a sink, the aggregation tree that minimizes the number of transmissions is the minimum Steiner tree. NP-complete….Approximations:
    - Center at Nearest Source (CNSDC): All sources send through source nearest to the sink.
    - Shortest Path Tree (SPTDC): Merge paths.
    - Greedy Incremental Tree (GITDC): Start with path from sink to nearest source. Successively add next nearest source to the existing tree.

Programming Paradigm

- Move beyond simple query with in-network aggregation model
- How do we task a 1000+ node dynamic sensor network to conduct complex, long-lived queries and tasks ??
  - What constructs does the query language need to support?
- What sorts of mechanisms need to be “running in the background” in order to make tasking efficient?
  - Small databases scattered throughout the network, actively collecting data of nearby nodes, as well as accepting messages from further away nodes?
  - Active messages traveling the network to both train the network and identify anomalous conditions?

Questions?

Thomas Fuhrmann

Department of Informatics
Self-Organizing Systems Group
c/o I8 Network Architectures and Services
Technical University Munich, Germany

fuhrmann@net.in.tum.de