Kapitel 5: Mobile Ad Hoc Networks

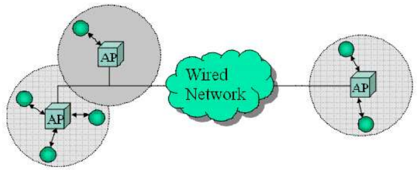
Mobilkommunikation 2
WS 08/09

Prof. Dr. Dieter Hogrefe
Prof. Dr. Xiaoming Fu

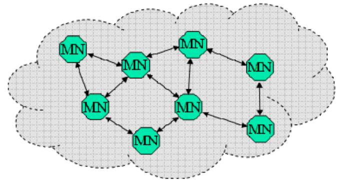
mk2@cs.uni-goettingen.de

Wireless Communication

- **Wireless communication networks types**
  - Infrastructure-based networks
  - Infrastructureless networks
    - Ad hoc networks



Infrastructure-based

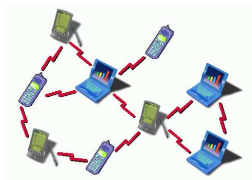


Infrastructureless

Characteristics

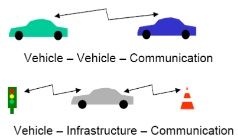
- **Infrastructure-based Networks**
  - Fixed backbone
  - Nodes communicate with access point
  - Suitable for areas where APs are provided

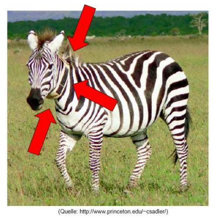
- **Infrastructureless Networks**
  - Without any backbone and access point
  - Every station is simultaneously router

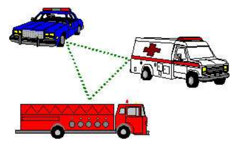


Applications of Ad Hoc Networks

- Vehicular ad hoc networks
- Wireless sensor networks
- user groups (conferences, search and rescue, military, …)







Ad Hoc Networks: Pros and Cons

- **Advantages**
  - Connectivity in areas without fixed infrastructure
  - Reduced administrative costs
  - Self-organizing, fast and easy deployed
- **Challenges**
  - All network entities may be mobile; very dynamic topology
  - No central entities; operation in completely distributed manner

Characteristics of Ad Hoc Networks

- **Nodes**
  - limited resources
  - dynamic topology
  - Address assignment
- **Wireless channels**
  - relatively high error rate
  - high variability in the quality
  - low bandwidth
  - broadcast nature
  - security aspect

Address assignment

- Since there is no DHCP server in ad hoc networks, there must be a special protocol address distribution and management
- The protocol must have the following features:
  - Avoidance of address conflicts
  - Addressing only during the stay of the node in the network
  - New nodes should be refused only if all of the addresses are dedicated
  - Possibility of merging networks
  - Considering authorisation

Address assignment: e.g. Buddy system (I)

- At the beginning there is only one node, A, in the network that has the entire pool of IP addresses
- When an un-configured node, B, wishes to join a network, it requests the nearest configured neighbor node for an IP address
- Node A assigns an IP address from its pool to the requesting node B, dividing its pool into two parts and gives one half to the requesting node B

Buddy system (II)

- A node can leave the network either gracefully or abruptly
- When node A leaves a network gracefully, it gives its address pool to any node nearby (e.g. to the 'buddy' node which owns the other half)
- when a node leaves the network abruptly it results in an IP address hole; because there is some IP addresses which are neither assigned to any node nor available for assignment to an un-configured node

Buddy system (III)

- Nodes synchronize from time to time to keep track of the IP addresses assigned and detect any holes in the available pool of IP addresses. Every node keeps a record of all the IP address blocks in the network by maintaining a table similar to this:

IP ADDRESS TABLE

NodeID	IPAddresses
1	0 - 31
5	32 - 63
3	64 - 127
...	...

- Literatures:
  - http://www.utdallas.edu/~mmohsin/publications/IPAssignment.pdf
  - http://www.acticom.de/fileadmin/data/publications/WWRF6_White.pdf

Routing Protocols

- A routing protocol is needed to discover routes from sender to receiver
- The routing protocol should have this characteristics:
  - Autonomous self-starting operation
  - Minimal overhead of control functions
  - Routing functionality over multiple hops
  - Managing the dynamic topology
  - Preventing loops in discovered routes

Two existing routing methods (I)

- **Proactive** routing protocols
  - Each node maintains routes to every other node
  - Periodically and/or event-based exchange of routing information
- Examples:
  - DSDV: Destination Secuenced Distance Vector routing
  - TBRPF: Topology Broadcast based on Reverse Path Forwarding
  - OLSR: Optimized Link State Routing

Two existing routing methods (II)

- **Reactive** (on-demand) routing protocols:
  - Sources initiate route discovery on-demand
  - Only active routes are managed
  - Usually less control overhead, and better scalability than proactive routings
- Disadvantage:
  - latency due to route calculation before communication
- Examples:
  - AODV: Ad hoc On-demand Distance Vector routing
  - DSR: Dynamic Source Routing

DSDV: Destination Secuenced Distance Vector routing

- Proactive
- Each node maintains a route for every node in the network in its routing table

  <dest-addr, dest-seq#, next-hop, hop-count, install-time>
- Each node manages the sequence numbers for its own routes
  - Update with every new information of network changes
  - Guarantee loop free routes

DSDV: Routing Updates

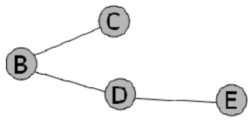
- Each node sends a periodic update to its neighbors
  - The update contains its own dest-seq# and other routing information for the destination

    < dest-addr, dest-seq#, hop-count >
- Also each node sends routing table updates caused by important changes in the network (e.g. link failure)
- When a node receives two routes to the same destination from two neighbors it would:
  - Choose the route with the largest sequence number
  - If the same: Choose the smalest hop-count

DSDV: Full/Incremental Updates

- Full updates
  - Contain all the routing information
  - Relatively are rare
- Incremental updates
  - Contain only the information which have been changed scince the last full update
  - Usually sent in a NPDU (Network Protocol Data Unit)
  - If the updates do not fit a NPDU, a full update would be sent

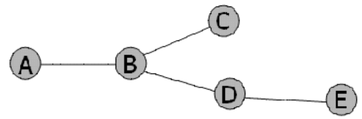
DSDV: Adding a link



B's Route Table:
B, 132, B, 0
C, 144, C, 1
D, 202, D, 1
E, 155, D, 2

1. An ad hoc network

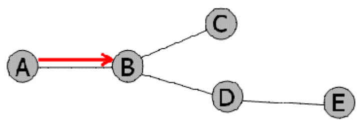
DSDV: Adding a link



B's Route Table:
B, 132, B, 0
C, 144, C, 1
D, 202, D, 1
E, 155, D, 2

2. Node A is added to the network

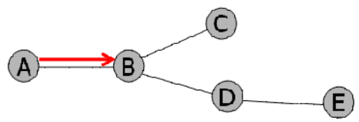
DSDV: Adding a link



B's Route Table:
B, 132, B, 0
C, 144, C, 1
D, 202, D, 1
E, 155, D, 2

3. Node A sends its routing table to its neighbors
<A, 101, 0>

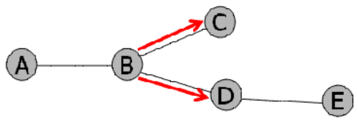
DSDV: Adding a link



B's Route Table:
B, 132, B, 0
C, 144, C, 1
D, 202, D, 1
E, 155, D, 2
A, 101, A, 1

4. Node B receives the information and adds
<A, 101, 1> to its own routing table

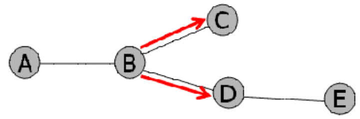
DSDV: Adding a link



B's Route Table:
B, 132, B, 0
C, 144, C, 1
D, 202, D, 1
E, 155, D, 2
A, 101, A, 1

5. Node B propagates the new routing information to its neighbors <A, 101, 1>

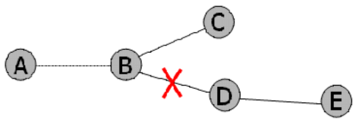
DSDV: Adding a link



B's Route Table:
B, 132, B, 0
C, 144, C, 1
D, 202, D, 1
E, 155, D, 2
A, 101, A, 1

5. Node B propagates the new routing information to its neighbors <A, 101, 1>
6. The neighbors complement their routing tables <A, 101, B, 2> and propagate their information to their neighbors

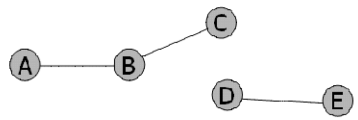
DSDV: Failure of a link



B's Route Table:
B, 132, B, 0
C, 144, C, 1
D, 202, D, 1
E, 155, D, 2
A, 101, A, 1

1. A link between B and D fails

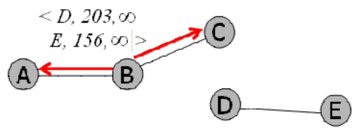
DSDV: Failure of a link



B's Route Table:
B, 132, B, 0
C, 144, C, 1
D, 203, D, ∞
E, 156, D, ∞
A, 101, A,1

2. Node B notices the failure:
- Sets hop-counts for D and E on ∞
- increment seq# for D and E

DSDV: Failure of a link



B's Route Table:
B, 132, B, 0
C, 144, C, 1
D, 203, D, ∞
E, 156, D, ∞
A, 101, A, 1

2. Node B notices the failure:
- Sets hop-counts for D and E on ∞
- increment seq# for D and E

3. Node B sends new routing information to its neighbors

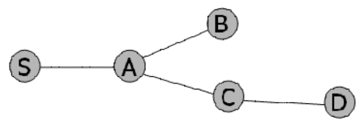
DSDV: Summary

- Proactive
- Routes are managed through periodic and event-based routing table exchanges
- Proactive protocols work best in networks with low or moderate mobility and a few nodes

AODV: Ad Hoc On Demand Distance Vector Routing

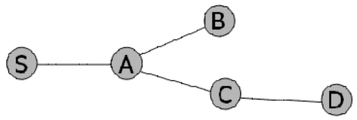
- Reactive
- Managing only active routes
- Sequence numbers serve as unique IDs and are used to prevent loops in the discovered routes, as a route refreshment criteria
- AODV is a descendant of DSDV
- Support unicast and multicast communication
- Nodes keep track of their neighbors by listening for HELLO messages that each node broadcasts at set intervals

AODV: Route Discovery



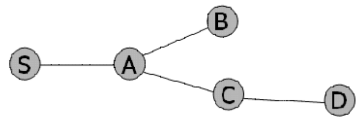
1. Node S needs to find a route to D

AODV: Route Discovery



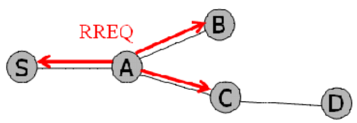
2. S initiates a Route Request (RREQ) message <D‘s-IP-addr., D‘s-Seq#, S‘s-IP-addr., S‘s-Seq#, hop-count(=0)>

AODV: Route Discovery



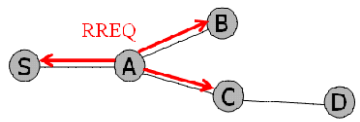
2. S initiates a Route Request (RREQ) message <D‘s-IP-addr., D‘s-Seq#, S‘s-IP-addr., S‘s-Seq#, hop-count(=0)>
3. S sends the RREQ message to its neighbors

AODV: Route Discovery



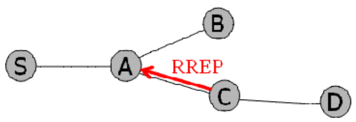
4. Node A receives RREQ
- makes route reverse entry for S
  Des=S, nexthop=S, hopcount=1
- A has no direct route to D, So sends RREQ to its neighbors

AODV: Route Discovery



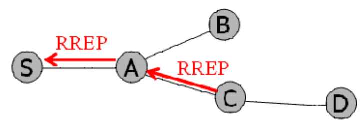
5. Node C receives the RREQ
- makes route reverse entry for S
  Des=S, nexthop=A, hopcount=2
- C has a direct route to D, and the seq# for route to D is S's seq# in RREQ

AODV: Route Discovery



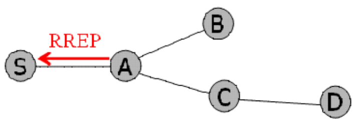
6. Node C receives RREQ
- C initiates a route reply (RREP)
RREP=<D‘s_IP_addr, D‘s-seq#, S‘s_IP_addr, hopcount_to_D(=1)>
- C sends the RREP to A

AODV: Route Discovery



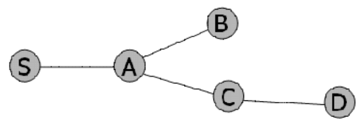
7. Node A receives RREP
- A makes a route forward entry for D
 Dest=D, nexthop=C, hopcount=2
- A sends RREP to S

AODV: Route Discovery



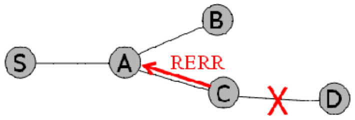
8. Node S receives RREP
- S makes a route forward entry for D
 Dest=D, nexthop=A, hopcount=3

AODV: Route Discovery



9. Node S receives RREP
- S makes a forward entry route for D
 Dest=D, nexthop=A, hopcount=3
- S sends the data packets on the new route to D

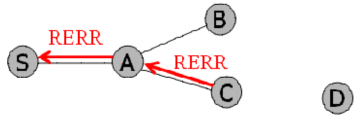
AODV: Route Management



1. The route between C and D fails
2. Node C invalidates the route to D in its routing table
3. Node C generates Route Error message (RERR)

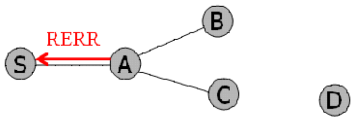
- Lists all of the destinations which are not accessible now
- Sends RERR to the upstream neighbor

AODV: Route Management



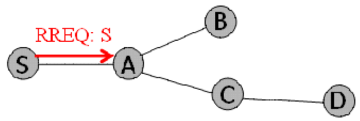
4. Node A receives RERR message
- Checks whether C is the next hop on the route to D
- Deletes the route to D
- RERR broadcast continues to S

AODV: Route Management



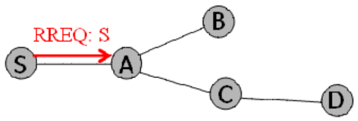
5. Node S receives RERR message
- Examines if A is the next hop on the route to D
- Deletes the route to D
- Finds a new route to D if necessary

DSR: Route Discovery



Node S needs a route to D
1. S sends RREQ message

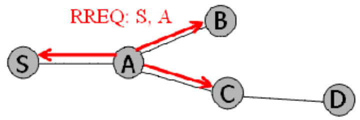
DSR: Route Discovery



2. Node A receives the message, but it has no route to D

- A adds its own address and sends the message to its neighbors

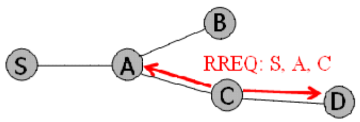
DSR: Route Discovery



3. Node C receives RREQ message, but has no route to D

- adds its own address and sends the packet to its neighbors

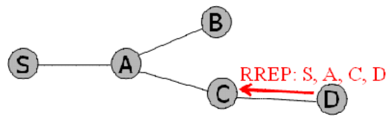
DSR: Route Discovery



4. Node C receives RREQ message, but has no route to D

- adds its own address and sends the packet to its neighbors

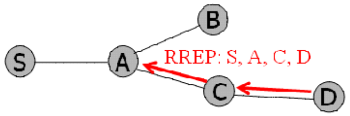
DSR: Route Discovery



5. Node D receives RREQ and sends a RREP back to C

- The RREP contains the route from S to D

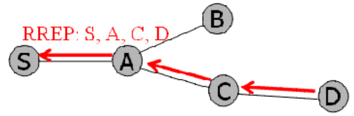
DSR: Route Discovery



6. Node C receives RREP

- RREP is broadcasted continuesly to A

DSR: Route Discovery



7. Node A receives the RREP message

- sends RREP to S

8. Node S receives the RREP message

- S will use the discovered route to forward data

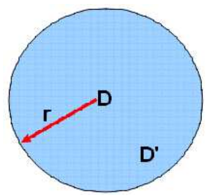
- DSR nodes may learn more than one route during route discovery

AODV and DSR differences

- DSR uses sourse routing
  AODV uses next hop record
- DSR route uses cashe
  AODV uses routing table
- DSR route cash entries have no lifetimes
  AODV routing table entries have lifetimes
- DSR has alternate route available when one breaks
  AODV nodes do not alternate route when one breaks

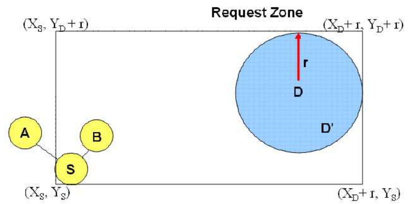
LAR: Location Aided Routing

- Using location information for routing
  - Location information is used (e.g. Obtained from GPS)
  - All nodes should know their current location
- **Expected Zone:** The area that probably contains the desired destination
  - EZ is estimated based on any previous location information and the destination‘s velocity
- D= last known location of node D at time t0
- D‘= location of node D at present time t1
  - D‘ is unknown to the sender
- r=(t1- t0)*Vd
- Vd= estimated speed of D



LAR: Request Zone

- Request Zone contains the expected zone and the location of the sender and the RREQ is limited in that
  - Only nodes within the request zone forward the RREQ
- Request zone explicitly is specified in RREQ
- Each node must know its location to determine whether it is within the request zone or not

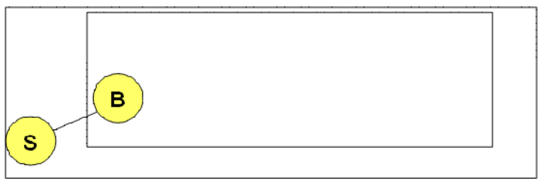


LAR: Request Zone

- If route discovery with the smallest request zone fails, the sender would try to discover the route with a larger request zone (after a time out)
  - The larger zone can cover the entire network
- The rest of the route discovery is similar to DSR

LAR Variation I: Adaptive Request Zone

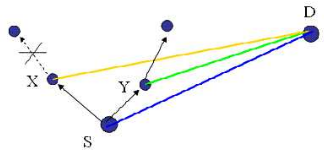
- Each node can modify the request zone in the RREQ
- The modification is based on newer and usually more accurate information and could be smaller than the original zone



LAR Variation II: Implicit Request Zone

- In the previous scheme the request zone was explicitly specified in RREQ
- The alternative approach is:

  node X forwards a route request received from Y if X is seemed to be closer to D compared to Y
- The motivation is to bring the route request closer to the destination node after each forwarding



LAR Variation III: Implicit Request Zone

- The assumption is that, *initially,* location information for node Y becomes known to X only during a route discovery
- This location information is used for a future route discovery
  - Each route discovery yields more updated information which is used for the next discovery
- **Variations**
  - Location information can also be piggybacked on any message from Y to X
  - Y may also proactively distribute its location information in the network

LAR: Advantages and Disadvantages

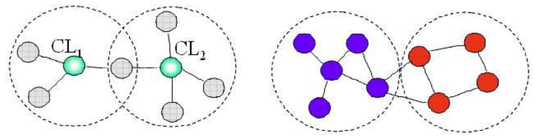
- **Advantages**
  – reduced RREQ flood (only in the request zone)
  – reduced route discovery overhead
- **Disadvantages**
  – Nodes need to know their locations
  – The possible existence of obstructions for radio transmissions is not taken into account

Hybrid: Zone Routing Protocol (ZRP)

- Proactive + Reactive
- Each node manages a zone (e.g. two hops)
- Intra-zone routing: proactive routing protocol within a zone (e.g. DSDV)
- Inter-zone routing: reactive routing protocol for nodes outside the zone
  - Boarder casting: the nodes at the edge of the zone lead the floods

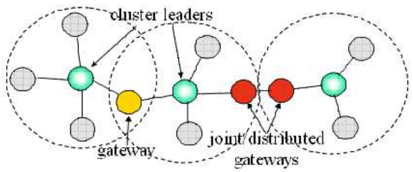
Cluster building/ Hierarchial routing

- The network is hierarchically structured
  - Cluster Leader (CL) represents the cluster; or
  - The control is distributed
- **Cluster Options:**
  - overlapping clusters / overlap free cluster
  - one-level hierarchy / multi-level hierarchy



Cluster building/ Hierarchial routing

- Each node communicates with its cluster leader
- Gateways connect the overlapping cluster
- Joint distributed gateways connect the non-overlapping clusters



Cluster building

- Clusters can be shaped considering
  - Current positions
  - Functional requirements

- Cluster leaders can be elected by
  - Highest ID algorithm
  - First Come, First Cluster

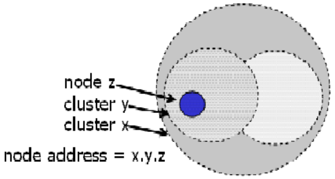
Cluster Leader Administration

- Clustering protocols have cluster leader election and revocation functions
- **Selection:** A node can be a cluster leader if it is out of the reach of any other cluster leader
- **Revocation:** To prevent growth in the number of cluster leaders some mechanisms should be provided, such as cluster leader can not be aleader again

Revocation Algorithms

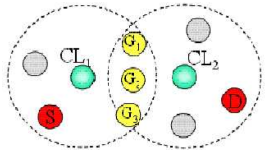
- If two cluster leader are in contact with each other, one of them should leave leadership
  - ID based algorithms
  - Weighted algorithms
  - Cluster members based algorithms
  - Subset based algorithms

Hierarchial Addressing



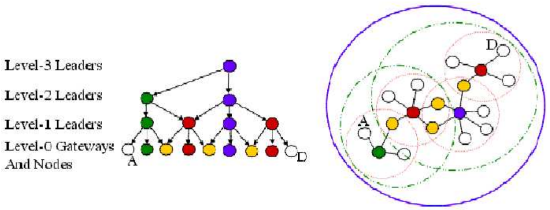
- Addresses can be assigned based on their cluster membership
  - If a node ID is z and he is a member of cluster y, then its address could be yz
  - If the node is generally a member of hierarchical clusters, then its address could be x .... yz
  - It could also have multiple addresses, if the clusters are overlapped

Hierarchical Routes



- Routes are defined Hierarchically
  - hierarchical routing increases the robustness of routes
    - In this example CL1 can use any gateway to reach CL2
- physicale Route: $S \rightarrow CL_1 \rightarrow G_2 \rightarrow CL_2 \rightarrow D$
- logical Route: $S \rightarrow CL_1 \rightarrow CL_2 \rightarrow D$
- Nodes exchange information by Hello messages
  - Hello from non-leader nodes includes a list of its cluster leaders address (cluster leader list)

Multi-level Hierarchical Routes



- Scalability of hierarchical routing is limited
- Multi-level hierarchical routing scaled for larger networks

Cluster building / Hierarchical routing

- **Advantages**
  - Increase the robustness of routes in delivery of data packets
  - Hierarchical address architecture
  - Reduction in routing control messages
- **Disadvantages**
  - Require periodic messages to maintain the cluster
  - Centralization of possible routes through Cluster Leader

Multipath Routing

- Establishment of multiple paths for each Couple (source, destination)
  - If the primary route fails, alternative routes will be used
- Balancing the load by splitting traffic on several routes

Energy Efficient Routing

- Several schemes are proposed recently
- Using energy routing metrics (e.g. energy consumption per packet transmission, network connection duration, etc.)
- Topology control: cessation of transmission of energy for each node
- Off nodes are eliminated
- Using directional antennas