UBICOMP

Episode 6: Routing

Hannes Frey and Peter Sturm
University of Trier

Outline

- Introduction
- Geographic Routing (greedy, planar graph traversal)
- Topology-based Routing (proactive, reactive)
- Comparison of both approaches

References

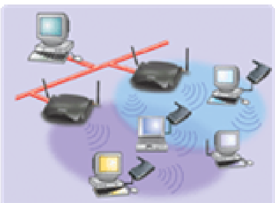
- S. Giordano, I. Stojmenovic, L. Blazevic, "Position based routing algorithms for ad hoc networks: a taxonomy", 2003
- E. M. Royer, C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks", IEEE Personal Communications 6(2), 1999

UBICOMP

Introduction

Mobile ad-hoc Networks (preview)

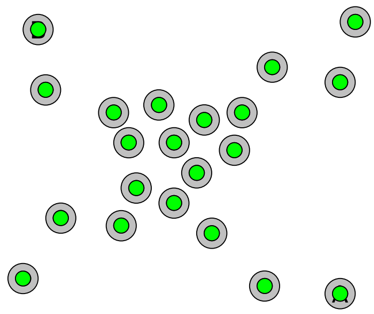
- Single-hop ad-hoc networks
  - MH, MSS, Handover, e.g. Mobile IP
- Mobile multihop ad-hoc networks
  - Each device acts as router and end system
- Multihop ad-hoc network scenarios
  - Spontaneous collaboration, disaster recovery, military operations
  - Sensor networks
  - Rooftop networks



- Why routing?
  - Limited transmission range
  - Energy consumption
  - What about flooding?

Communication by flooding

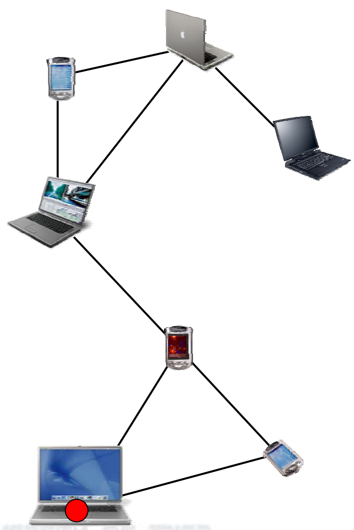
- Pro
  - Simple
  - Works also for highly dynamic topology changes
- Con
  - Lots of message duplicates
  - All nodes always involved
  - Loops have to be avoided
  - Redundant message transmissions
  - Broadcast storms
  - Memorization of sent messages
- Improvements?



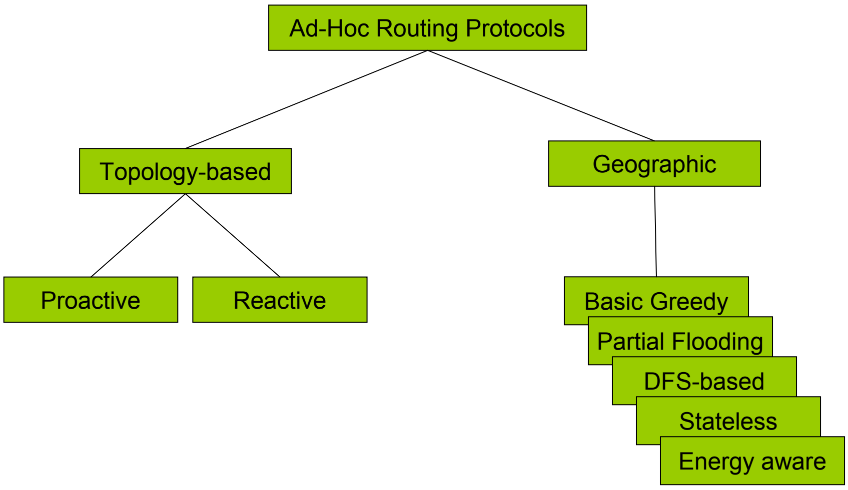
Play FLOOD Simulation

Challenges for routing in ad-hoc networks

- Changing topology
- Unreliable network-links
  - Due to mobility
  - Signal Attenuation
  - Unrelated transmissions and signal errors
  - Obstacles
  - Multipath and fading
- Address information is independent of device location
- Communication along intermediate hops



Protocol classification



A Protocol Taxonomy

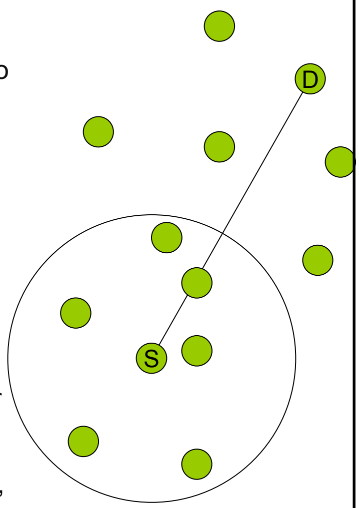
- Loop-freedom (mobility cased loops)
- Distributed operation: localized, global, zonal
- Path strategy: single path, flooding, multi-path
- Metrics: hop count, power, cost
- Memorization
- Guaranteed delivery
- Scalability
- Robustness

UBICOMP

Geographic Routing

Geographic Routing Principle

- Each device has to know its position
- Position information has to be announced to devices in vicinity
  - 1-hop, 2-hop; exception: beaconless routing
- Opposed to classical routing, destination address is a position stored in message
- On receipt of a message forward to “best” neighbor(s) regarding own, destination, neighbor(s) position, and metric being optimized
- Position of receiver device has to be acquired in advance
  - Message update in each routing possible for proactive location service
- New communication patterns: Geocast, GHT, Marketplaces, Location-aided routing,



Network Model

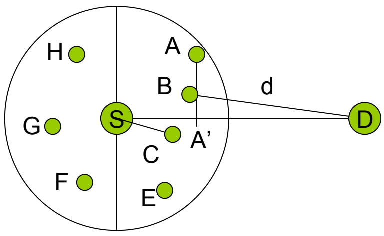
- Idealized: Radio signals form a sphere around a sending device
- Even unicast is a also a broadcast; devices in promiscuous mode
- Signal is attenuated with a fixed exponent
- Routing needs bidirectional links
- Wireless networks can be modeled as geometric graphs
  - Geographical position of a device defines a point in the plane
  - There is a link between two nodes if they are bidirectional connected
  - What about unidirectional links?
- How to provide bidirectional links in reality?
  - Proactive link acknowledge
  - Link removal on missing packet acknowledge
  - Geometric properties of the network model

UBICOMP

Greedy Packet Forwarding

Greedy Packet Forwarding

- Select neighbor node(s) with the “best” location(s) regarding the metric being optimized
- Each node applies this greedy principle until destination is eventually reached
- Characteristics of greedy routing depends on metric being optimized
- Basic strategies
  - Progress (forward/backward direction)
  - Distance
  - Direction

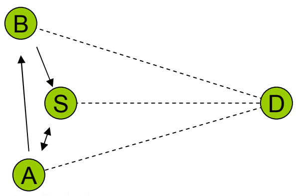


Loop-freedom of Greedy Routing Methods

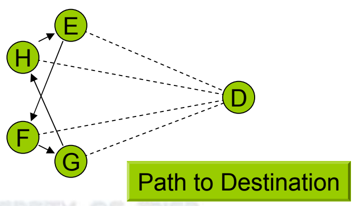
- Forwarding based on distance and progress consider nodes in forward direction only to provide loop-free operation (see Fig. (a))
- Direction-based strategies do not guarantee loop-free operation (see Fig. (b))

Theorem: Each greedy routing algorithm forwarding packets to neighbors closer to the destination or within the most forward progress guarantees-loop free operation, while greedy algorithms forwarding packets to the neighbor with closest direction (and possible to other neighbors) are not loop-free.

(a)

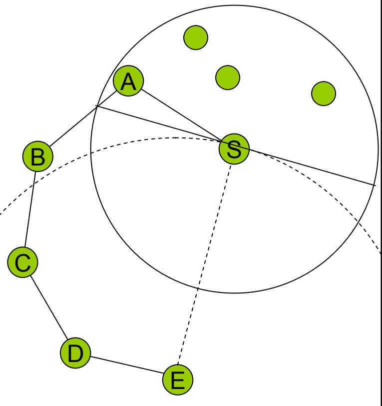


(b)



Greedy Routing Failure

- Choosing node in backward direction may lead to packet loops
- Nevertheless, may exist a path from S to D (S may also be an intermediate node)
- Node S forms a local minimum, termed concave node
- Loop-freedom and delivery rate are conflicting goals
- Solutions?

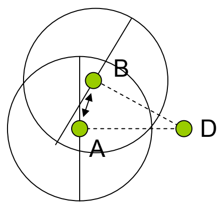


CONCAVE

Basic Single-Path Strategies (1)

- The first position-based routing methods (mid 80s)
  - Takagi, Kleinrock: most forward within radius (MFR)
  - Finn: distance-based greedy routing (widely applied)
- Produce nearly the same path
- If successful performance close to Dijkstra's Single Source Shortest Path algorithm
- Delivery rate decreases significantly in sparse networks

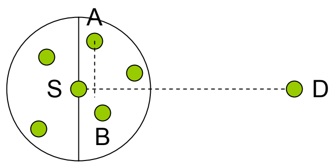
- A remark on loop-freedom of MFR and distance-based greedy routing



MFR GREEDY

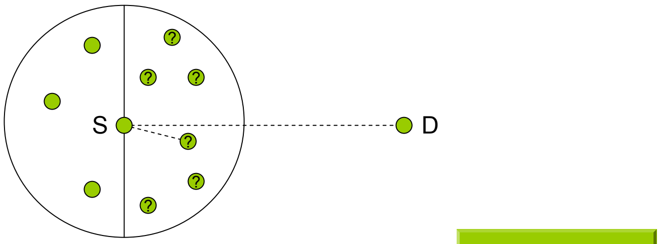
Basic Single-Path Strategies (2)

- Is maximizing progress always a good choice?
  - Energy consumption when signal strength can be adjusted
  - Signal attenuation, node mobility
- Hou, Li: nearest within forward direction (NFP)
  - Probability of message loss reduced significantly
- Stojmenovic, Lin: nearest closer (NC)



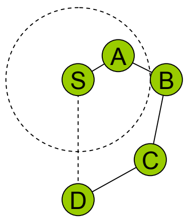
Basic Single Path Strategies (3)

- Nelson, Kleinrock: random progress method (RPM)
  - Overcome tradeoff between progress and transmission-success
- Kranakis et al.: compass routing (DIR)
  - First direction based approach
  - Tries to minimize Euclidean path length a packet has to travel



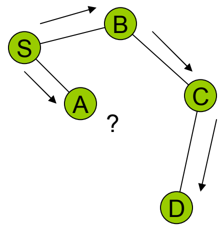
Improved Single-Path Strategies (1)

- Allow message to travel one hop in backward direction (Stojmenovic, Lin: GEDIR)
- i.e. packet dropped only if it would be sent back to the previous node



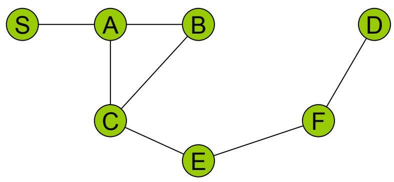
GEDIR

- Select “best” node among 2-hop neighbors
  - If next hop not reachable, select 1-hop neighbor greedily
  - (2-MFR, 2-GEDIR, 2-DIR, …)



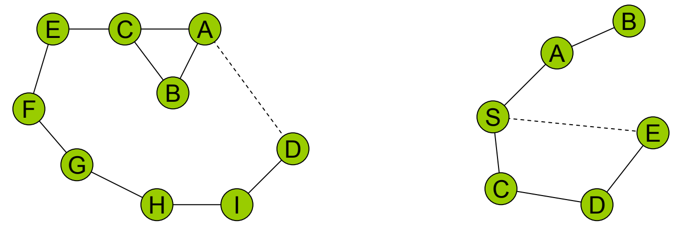
Improved Single-Path Strategies (2)

- Memorization: allow more than one node in backward direction avoiding message loops
- Alternate-{GREEDY,DIR,MFR}
  - *ith* received message forwarded to the *i*th “best” neighbor
  - Example: path selected for message from S to D: SABACBCEFD
- Disjoint-{GREEDY,DIR,MFR}
  - Forwarding neighbors removed from set of possible next hop nodes
  - Example: path selected for message from S to D: SABCEFD
- In general, disjoint performs better than alternate in most cases



Improved Single-Path Strategies (3)

- Does alternate- and disjoint-greedy guarantee delivery?



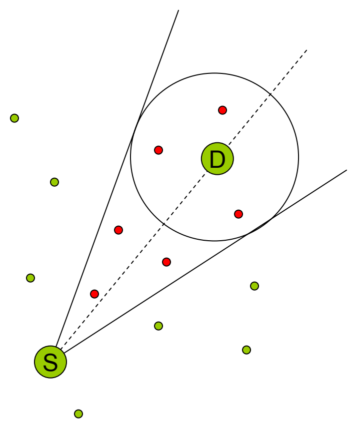
Alternate: ABACBCA → Drop
Disjoint: ABCE…D → Ok

Alternate: SABASCDE → Ok
Disjoint: SAB → Drop

- Example for both not successful?
- Simulation results show that disjoint-greedy is more successful than alternate-greedy

Restricted Directional Flooding (1)

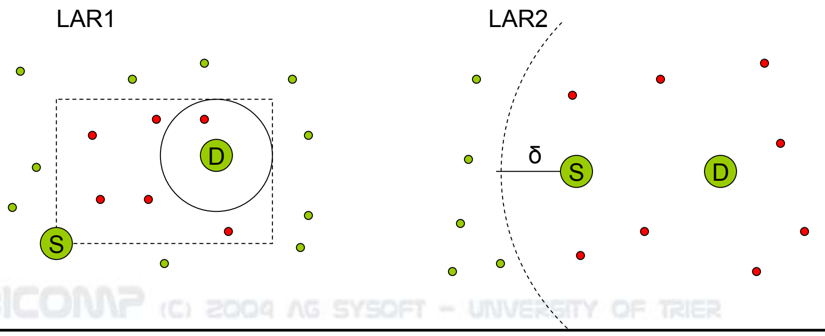
- Forward message to nodes in general direction to destination
- Redundant message transmission increases success and message complexity
- Requires memorization, however, delivery is not guaranteed
- Basagni et al.: DREAM
  - Forward to all nodes in a certain angular range
  - Radius of circle centered around D reflects maximum possible movement



DREAM

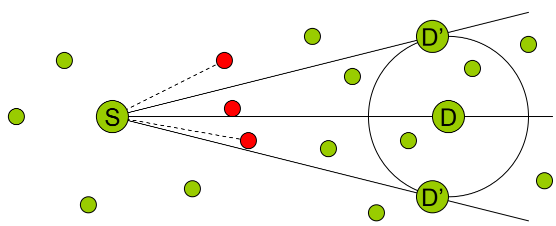
Restricted Directional Flooding (2)

- Ko, Vaidya: location aided routing (LAR)
  - Originally intended to support topology-based reactive routing protocols
  - Restrict flooding needed to discover new routes (request zone)
  - LAR1: flooding restricted to rectangular region containing expected zone of destination
  - LAR2: restrict to nodes at most some delta δ more afar from D



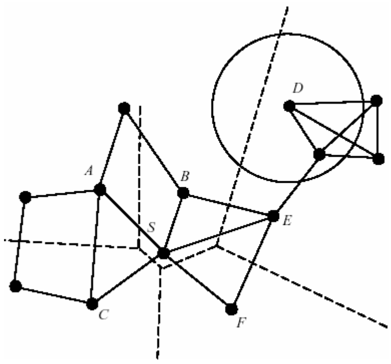
Restricted Directional Flooding (3)

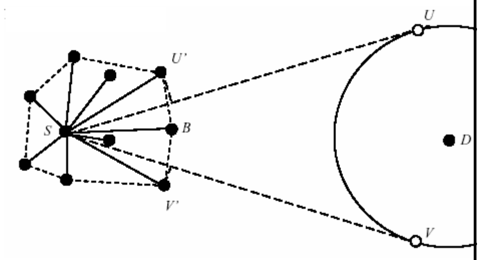
- Stojmenovic et al.: Concept of restricted directional flooding can be generalized for distance-, progress- and direction-based greedy routing
- Determine all possible "best" next hop nodes for each possible destination position within expected area
- R-DIR: direction-based, angular range



Restricted Directional Flooding (4)

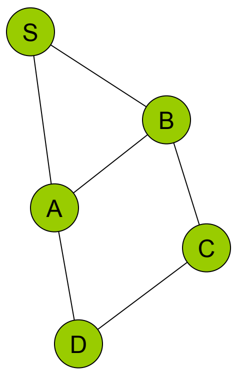
- V-GEDIR: distance-based, Voronoi diagram
- CH-MFR: progress-based, convex hull





Multipath Strategies

- Source node initially forwards message to c best neighbors
- Receiving node performs one of the known greedy strategies
- Message duplicates are either ignored or handled as in alternate and disjoint methods
- Examples:
  - Original 2-greedy: SAD, SBA
  - Alternate 2-greedy: SAD, SBABCD
  - Disjoint 2-greedy: SAD, SBCD



Energy Aware Greedy Routing (1)

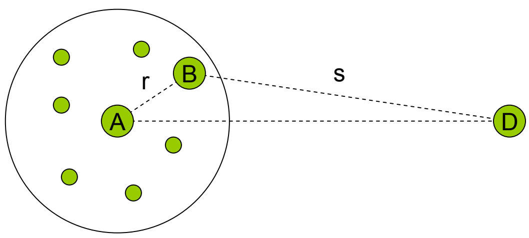
- Idea: choose nodes within optimal transmission range
- It holds the following theorem:

Theorem: Let d be the distance between the source and destination. The power needed for direct transmission is $u(d)=ad^b+c$ which is optimal if $d<=(c/(a(1-2^{(1-b)})))^{(1/b)}$. Otherwise, n-1 equally spaced nodes can be selected for retransmissions, where $n=d(a(b-1)/c)^{(1/b)}$, producing minimal power consumption of about $v(d)=dc(a(b-1)/c)^{(1/b)}+da(a(b-1)/c)^{((1-b)/b)}$.

- Nodes can't be placed arbitrary, but result can be used to select the optimal next hop node

Energy Aware Greedy Routing (2)

- Power aware greedy routing to reach destination D:
  - Node A selects node B that minimizes $u(r)+v(s)$, where $r=|AB|$, and $s=|BD|$



- Principle can be used to define other routing schemes
  - Cost-aware: function proportional to remaining battery power (Increases Lifetime of the whole network)
  - Power-cost-aware: Combine both metrics

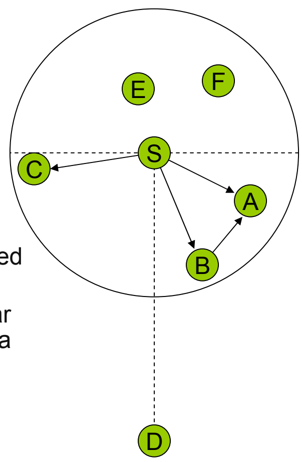
UBICOMP (C) 2004 AG SYSOFT – UNIVERSITY OF TRIER

Beacon-less Routing (1)

- Traditional greedy routing need information about all one-hop neighbors
  - Periodic hello messages
  - Transmitted with maximum signal strength
  - Independently of current data traffic
  - Problem of bidirectional connections
- Heissenbüttel, Brown: Beacon-less routing (BLR)
  - Node is unaware of its neighbors
  - Just broadcast a message to all unknown neighbors
  - Receiving node introduces a small timeout before forwarding
  - Node located at the “best” position introduces the fewest delay
  - Nodes hearing of retransmission cancel the scheduled packet

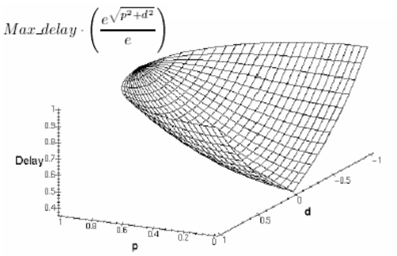
Beacon-less Routing (2)

- Problem: Message duplicates
  - E and F are in backward direction
  - E.g.: B introduces fewest delay
  - A removes scheduled packet
  - C does not hear transmission from B and forwards the packet too
- Avoiding message duplicates
  - only nodes in a certain forwarding area allowed as candidate nodes
  - Nodes in forwarding area are able to overhear retransmission of each other node in that area
- Active selection method: control Message instead of full packet
  - Forwarding node sends unicast to “winning” node
  - Large packet can be sent with reduced transmission power

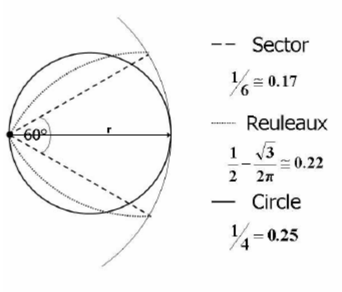


Beacon-less Routing (3)

- Possible delay functions (r=radius, p=progress, d= distance)
  - Basically MFR: Max_delay(r-p)/r
  - Slightly modified NFP: Max_delay(p/r)
  - An advanced delay function



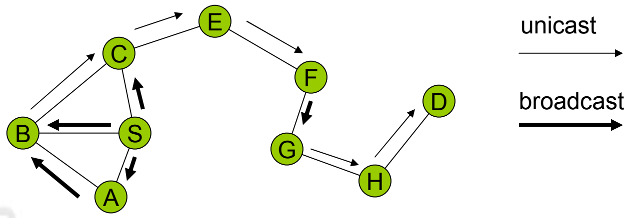
- Possible forwarding areas
  - Circle good forwarding area regarding progress and successful hops



BLR Example

Guaranteed Delivery based on Memorization (1)

- Stojmenovic, Lin: Partial flooding to guarantee delivery (f-GEDIR, f-MFR, f-DIR)
  - Intermediate nodes handle packet according to GEDIR, MFR, …
  - Concave node broadcasts packet to all neighbors
  - To avoid message loops: concave node rejects further copies of the message, concave nodes are removed from the list of candidate nodes
- Example: Message from S to D



Guaranteed Delivery based on Memorization (2)

- When there is a path from source to destination then one of the neighbors lies on the path → guaranteed delivery
- Improvement: Component routing
  - Flooding produces many redundant message transmissions
  - Unit graph model assumed (see next chapter)
  - Concave node may determine connected components in the subgraph of its neighbors
  - Forward message to only the best neighbor in each component
  - Number of message transmissions reduced significantly: concave node has at most four connected components

Guaranteed Delivery based on Memorization (3)

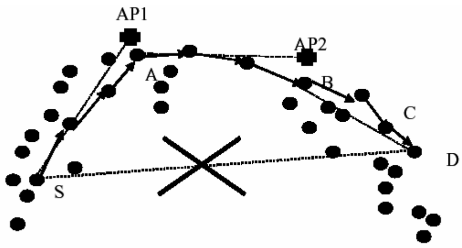
- Jain et al.: Geographic Routing Algorithm (GRA)
  - Intermediate node handles message greedily
  - Concave node maintains route to destination node
  - Start route discovery for outdated routing tables
  - Stuck packet is routed to destination after successful route discovery
- How to perform route discovery?

Guaranteed Delivery based on Memorization (4)

- Standard graph traversal algorithms: DFS and BFS
- Breadth first search: Equivalent to flooding
- Depth first search from concave node S
  - Yields an acyclic path from S to D
  - Node X puts its address on route discovery packet p
    - Forward to neighbor who has not seen p before
    - Select neighbor Y which minimizes |XY|+|YD|
  - If no possible neighbor exists, remove address from p and send it back to the node from which p was originally received
- Other metrics may be applied on next neighbor selection
  - Quality-of-service paths (delay and bandwidth criteria, connection time, …)

The Approach of Terminodes

- Terminodes Project
- Blazevic et al.: Terminode routing
- Combination of two components: TLR, TRR
- Terminodes Local Routing
  - Reach nodes in vicinity
  - No location information used → topology based routing
  - Rationale behind TLR → imprecise location information
- Terminodes Remote Routing
  - Reach remote nodes
- Anchored Geodesic Packet Forwarding (AGPF) major novelty of TRR
  - Anchored path comprised of a list of fixed geographic points
  - Packet loosely follows anchors in greedy mode
  - Next anchor used when arriving at a node with current anchor in transmission radius



Planar Graph Routing

A Simplified Network Model

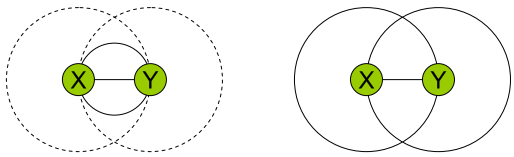
- Unit Disk Graph (V,E) defined on point set V, UDG(V):

$$(v, w) \in E \Leftrightarrow |vw| \leq R$$

  - R reflects the sending radius equal for each device
- Generalizations
  - Minpower-graph (different sending radii)
  - Subset of unit disk graph (obstacles)
- Following described algorithms need planar graph
  - Does UDG suffice?
  - Extracting planar graph locally

Planar Graph Construction Based on β-Skeletons

- Gabriel graph GG, Relative Neighborhood Graph RNG



- General class β-skeleton
  - GG and RNG forming extreme cases for planar graph construction based on this class
- How to obtain GG and RNG locally?

Localized Gabriel Graph Construction

- Lemma: If $UDG(S)$ is connected then $GG(S) \cap UDG(S)$ is connected.
- Lemma: Let u and v be points of $UDG(S)$ such that $(u,v) \notin GG(S)$ and let w be a witness to this. Then $(u,w) \in UDG(S)$ and $(v,w) \in UDG(S)$.
  - Proof: …
- Algorithm for each node v

$$\forall u \in N(v) : DISK(u,v) \cap (N(v) \setminus \{u,v\}) \neq \emptyset$$
$$\Rightarrow DELETE(u,v)$$

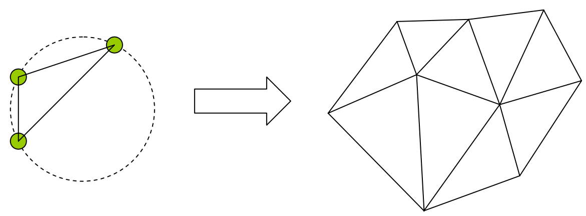
- Connectivity and planarity of localized RNG construction?

Excursion: Spanning Ratio

- Spanning Ratio: maximum ratio over all node pairs (X,Y) of Euclidean length of shortest path connecting (X,Y) and their direct Euclidean distance.
  - Used to rate graph construction
  - E.g. fully connected graph
- Spanning ratio of GG and RNG for n nodes
  - GG: spanning ratio of O(sqrt(n)) in the worst case
  - RNG: spanning ratio of O(n) possible
  - Thus, both have ratios depending on number of nodes

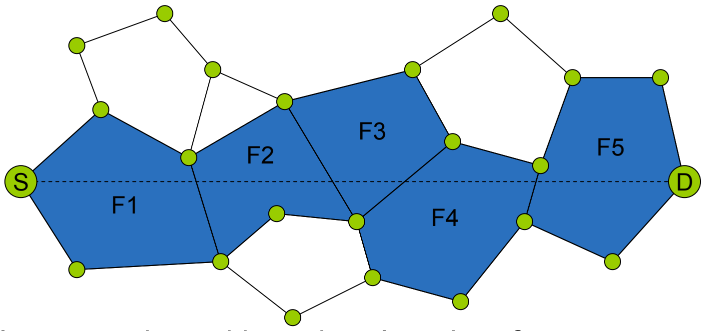
Other Localized Planar Graph Constructions

- Delaunay Triangulation



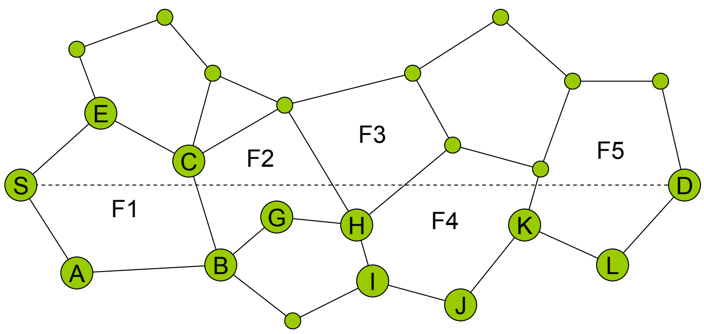
- Delaunay Triangulation has a constant spanning ratio
- Arbitrary long edges (no local construction)
- Existing local methods
  - Constant spanning ratio
  - Increased communication overhead opposed to GG and RNG
  - Alternative: 2-hop information

The Face Routing Principle



- Planar graph partitions the plane into faces
- Main idea: route packets along faces intersected by straight line connecting S and D
- Example: packet visits face sequence F1, …, F5

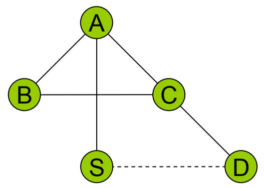
Face Traversal



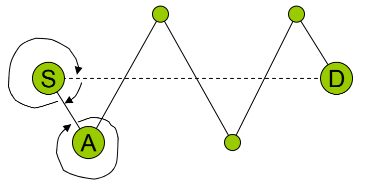
- Right/left hand rule
  - Forward packet along next edge clockwise/counterclockwise from the edge where it arrived (e.g. right hand rule in F1: SABCESAB…)
- When packet arrives at an edge intersecting the straight line SD, the next face intersected by this line is handled the same way
- Example: Right hand rule leads to path SABGHIJKLD

Loop-free Operation of Face Routing

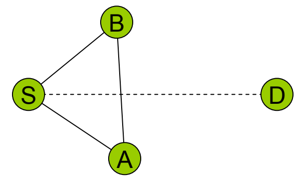
- An example where face routing leads to a packet loop when planarity is not provided: SABCABC…



- The last intersection point has to be remembered: SASA…



- The first edge traversed has to be remembered: ASSBASBA…



The Complete Face Routing Algorithm

- Extract a planar graph
- P := S
- REPEAT
  - Determine Face F with P on its boundary that intersects line PD
  - Let E be the first edge traversed along F
  - Traverse F until reaching an edge that intersects PD at some point Q different from P or until E is traversed twice in the same direction
  - P := Q
- UNTIL P=D or E is traversed twice in the same direction

FACE

Properties of Face Routing Algorithm

- FACE algorithm is loop-free and guarantees delivery in static connected planar geometric graphs (UDG needed for planar graph construction)
- Path length produced by FACE increases opposed to shortest path length, when average network degree increases (due to limited degree of planar graph)
- Remember: Greedy routing has performance close to shortest path when successful
- Bose et al.: Combination of FACE and Greedy Algorithm: GFG
  - Packet routed in FACE mode only until reaching a node closer to destination than the node where greedy routing failure occurred
- Karp, Kung: GPSR (the same as GFG, focused on medium access layer, and moving nodes)
- Sooner-back-method: consider also each neighbor during face traversal

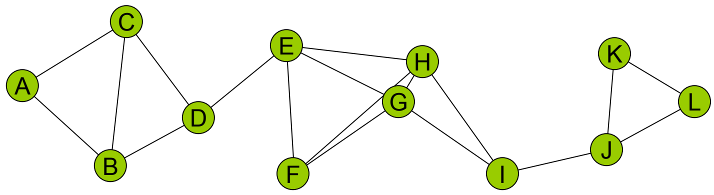
GFG

Internal Node Based Face Routing (GFG-I)

- Idea: Reduce hop count by reducing number of network nodes used by face traversal
- Definition: A subset S of network nodes G is termed a *dominating set (DS)* if each node is either element of S or has at least one neighbor in S
- Datta et al.: Perform FACE algorithm only on internal nodes defined by a connected dominating set
  - Gabriel graph construction performed on DS only
  - If concave node is no internal node forward to neighbor in DS
  - Route along Gabriel graph until
    - Local minimum handled
    - Or node with destination in its neighbor list reached

How to Construct a Dominating Set Locally

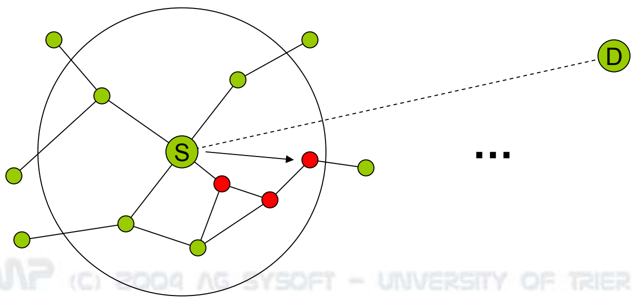
- Wu, Li: Preserve nodes which have two unconnected neighbors (no message exchange needed for UDG)



- Two additional refinements: A is covered by B if each neighbor of A is also neighbor of B and Key(A) < Key(B)
  - Only preserve those nodes not covered by any neighbor (inter-gateway nodes)
  - (Preserve only inter-gateway nodes not covered by any pair of connected neighboring nodes)

Shortcut-Based Routing (GFG-S)

- Possibly more neighbor nodes along path produced by face traversal
- Locally construct planar graph used by all neighbor nodes → 2-hop neighbor information needed!
- Perform a local planar graph traversal until reaching the last node in view and send packet to that node directly

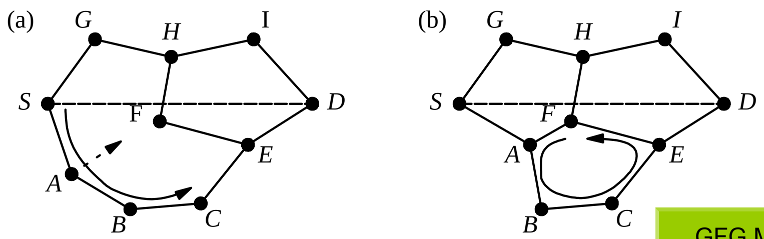


Energy-Aware Face Routing (PFP-I-S)

- Stojmenovic, Datta: GFG used in combination with an energy-aware greedy routing mechanism
- How to make face traversal energy efficient too?
  - Internal nodes → shorter paths
  - Shortcut-based routing→ Select power-optimal next hop node
- Internal nodes may fail earlier → shorter network lifetime
- Wu et al.: Apply cost metric to dominating set construction
  - Roughly: Basic local DS construction with an additional rule for removing redundant nodes with low remaining battery power

Planar Graph Routing and Mobility

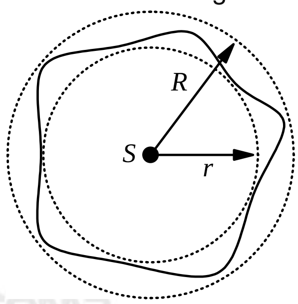
- Face routing proved to be loop for static UDG
- Mobility may lead to packet loop (also in combination with greedy routing)
- Loop frequency depends on mobility rate
- Has to be concerned in case of: increased device mobility, low bandwidth, long forwarding paths, or long queuing times
- Solution: Timestamps?
  - Decreasing connectivity
  - Loops due to removal



GFG Mobile

Unstable Transmission Ranges (robust-GFG)

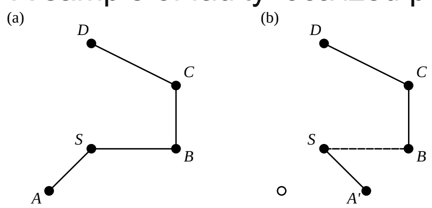
- FACE+GABRIEL provides guaranteed delivery and proper operation on UDG (Bidirectional links are not sufficient)
- Barriere et al.: Restricting the variation of sending radius



- Restriction guarantees, that each node inside U(X,Y) of connected X and Y is seen by at least one of them
- Construction of a virtual supergraph → virtual edges
- Exctract Gabriel graph locally on supergraph
- Routing along virtual edges
- Problem: Tower-like Construction

The Effect of Localization Errors

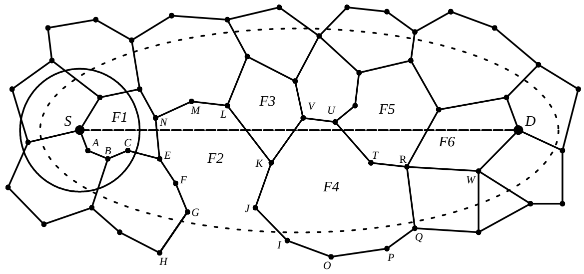
- A sample of faulty localized planar graph construction



- Loop construction also possible
- Simulation: Error frequently caused due to disconnection
- Simple problem fix: Remove on acknowledged request
  - Does not guarantee delivery

Restricting the Searchable Area (GOAFR)

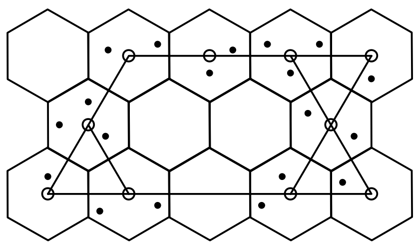
- Efficient operation of face routing depends on initial face traversal direction (e.g. right hand rule for outer face F2)



- Kuhn et al: Limit face traversal on ellipse containing the optimal path → reverse traversal when hitting ellipse
- Size of ellipse not predictable → adapt ellipse size

Geographic Cluster Routing (GCR)

- Face traversal restricts next hop selection to a subset of possible next hop nodes
- Frey et al: Perform Face Traversal along edges of adjacent geographical clusters defined by an infinite mesh of polygons



- Any interests to participate in ongoing research?

UBICOMP

Summary on Geographic Routing

Summary on presented Greedy Methods (1)

- Delivery rates for DIR, GEDIR, MFR comparable and greatly depend on network degree
  - Sparse networks (degree 4): only about 50%
  - Dense networks: over 90%
- 2-hop neighbor information provides minor improvement (10% in sparse networks)
- GEDIR and MFR
  - select same paths in most cases
  - When successful competitive with Dijkstra
  - Paths from DIR tend to be slightly longer
- NC improves NFP which has low success rates due to greedy routing failures
  - Preserve energy only in large networks
- Power-routing competitive with MFR, DIR, GEDIR
  - Outperforms all known greedy methods regarding power consumption
  - Competitive with Dijkstra applied on power, cost and power-cost metric

Summary on presented Greedy Methods (2)

- Known Beacon-less routing algorithms cover at most 0.25 of total transmission range
  - Leads sooner to greedy routing failure compared to traditional greedy routing (covering at most 0.5)
  - Performance of BLR comparable in dense networks with low or no mobility
  - BLR outperforms conventional position-based routing (suffering from outdated position information) under high mobility
- Simulation shows superiority of V-GEDIR, CH-MFR, and R-DIR over DREAM and LAR
  - Higher delivery rates with reduced flooding rate
- Disjoint scheme has higher success rate compared to alternate scheme
- Delivery rates of multipath strategies comparable to best existing restricted directional flooding algorithms
  - Linear communication overhead reduced to O(sqrt(n))
  - Experimentally observed: more than three paths do not compensate for additional flooding rate

Summary on presented Greedy Methods (3)

- For networks with uniform two-dimensional node distribution single-path greedy routing creates O(sqrt(n)) packets in average for random routing task
- Single-path methods differ in amount of traffic produced and memory needed to keep neighbor information up to date (beaconless, 1-hop, 2-hop)
- It holds for average number of neighbors of d
  - Locality of 1: message complexity O(n), state volume per device O(d)
  - Locality of 2: message complexity O(nd), state volume per device remains O(d)

Discussion of Planar Graph Routing (1)

- Average dilation: hop count compared to shortest path
  - in sparse networks (4 nodes) GFG degrades significantly
  - Dense population GFG comparable to shortest path (why?)
- Internal nodes and shortcuts drastically reduce AD
  - In sparse networks mainly by internal nodes concept
  - In dense networks performance gain due to shortcuts
- Power consumption significantly reduced by PFP-I-S
  - Advantage of shortcuts more notable
- Mobility caused loops not solved for high mobility rate

Discussion of Planar Graph Routing (2)

- Path lengths produced by robust-GFG
  - What means routing along virtual edges?
  - Number of hops produced depends on variation in transmission range and minimum node distance
  - For any k<n a node configuration can be constructed with virtual path length > k
- Localized algorithms performing routing with the square of the cost needed for the shortest path are asymptotically optimal → GOAFR is asympt. optimal
- Advantages of GCR
  - Arbitrary next hop selection by using one-hop information only
  - Graph structure remains more stable when network nodes move

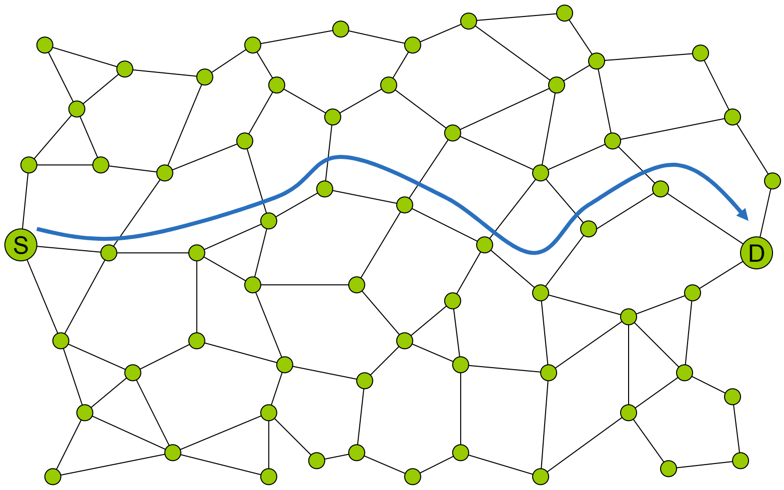
Open research issues

- Path lengths produced by face routing on other planar graphs and communication cost needed to construct them locally
- Face routing in subsets of arbitrary minpower graphs
- Location inaccuracies
- Loops due to dynamically changing network topologies
- QoS, congestion control, end-to-end delay, 3D

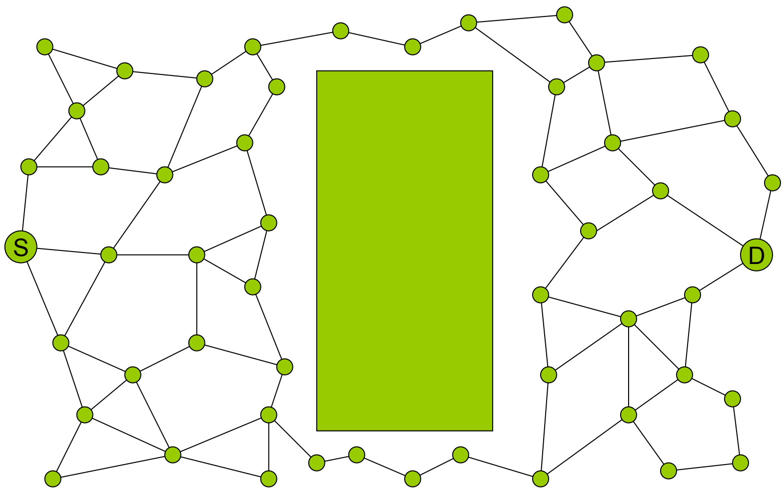
UBICOMP

Supplement: Geographic Cluster Routing GCR

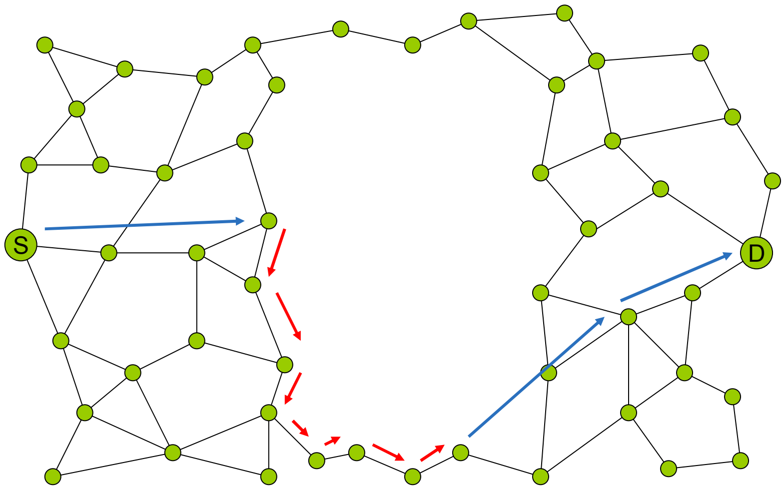
GFG applied in dense uniform node distribution



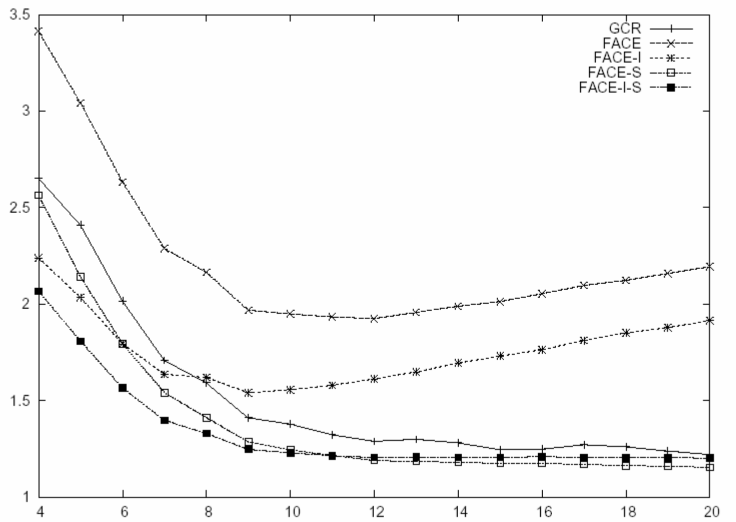
What if spatial distribution is nonuniform?



Face component increases hop count

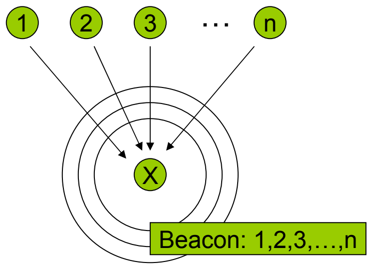


The impact of Internal nodes and shortcuts

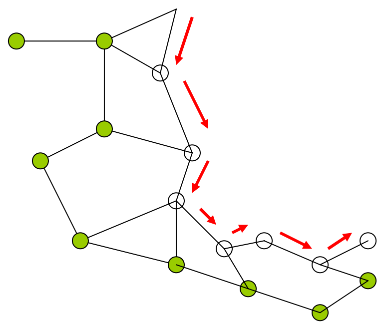


Is shortcut procedure the best solution?

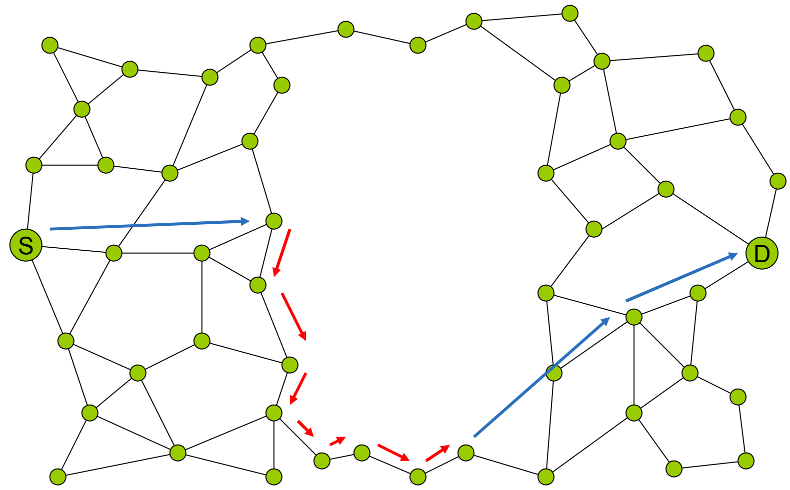
- Size of a beacon message does not scale with number of neighbors



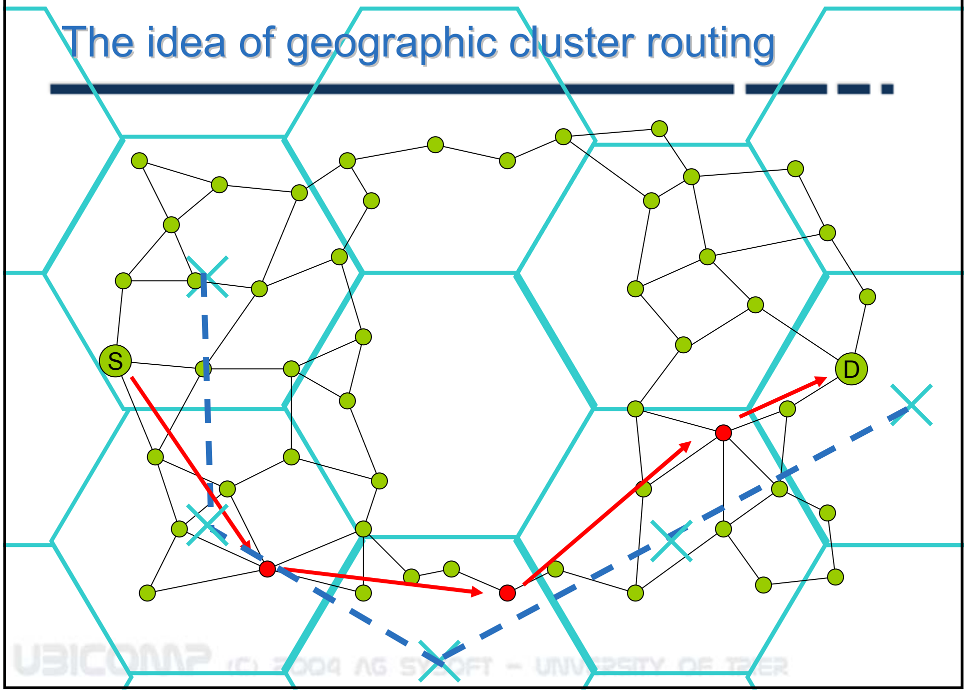
- Free node selection restricted along the traversed path
  - Additional energy consideration not possible



How to achieve shortcut with one-hop information?

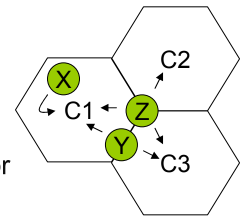


The idea of geographic cluster routing

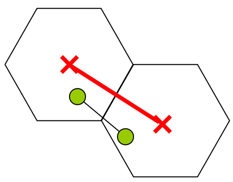


The Concept of Node Aggregation

- Definition: Geographical Cluster
- Device position maps device on exactly one cluster
  - What if device located on a cluster boundary or cluster edge
  - Define well defined mapping by using a total ordering:
    (x1,x2) ↔ (x1<x2) or (y1<y2 if x1=x2)

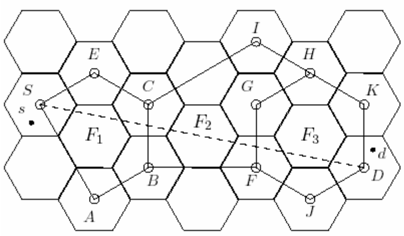


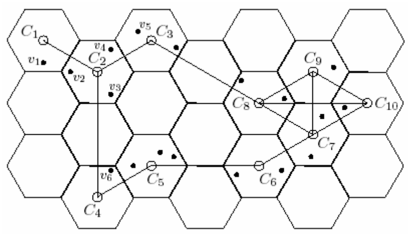
- Definition: Adjacent Clusters
- Graph defined by adjacent clusters of an aggregated unit disk graph forms an geometric graph



Planar Graph Routing along Adjacent Clusters

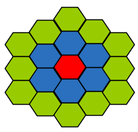
- Principle of face routing can be applied on overlay graph
- However, the graph fined by adjacent clusters is not necessarily planar
- Setting the right cluster size for uniform sending radius r
  - <= r: to perform routing inside a cluster (see next)
  - >2/sqrt(7)*r: for planar graph construction (see next)
  - <= 2/sqrt(7)*r ???

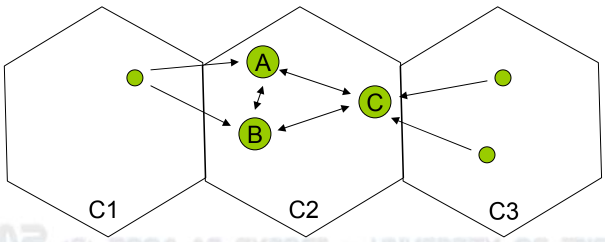




Constructing the aggregated graph locally

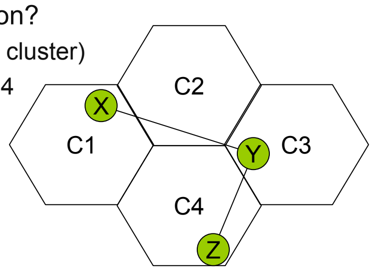
- Global construction straight forward
- Local construction
  - Each node announces its local view to all other nodes inside its cluster
  - Collect all announcements from all other cluster members
  - Construct global view of all adjacent clusters
  - Remove cluster edges if all devices announcing this edge have dissapeared
- Beacon message size remains O(1)
  - Number of adjacent clusters limited to c (currently 18)
  - Each cluster can be represented by one Bit





Extracting a Planar Graph

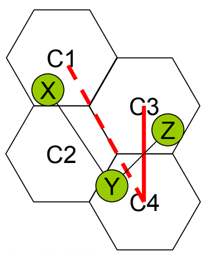
- Simply applying Gabriel graph construction?
  - X preserves edge C1→C3 (sees no other cluster)
  - Y removes edge C3→C1 due to cluster C4
  - Inconsistent view→ Loops

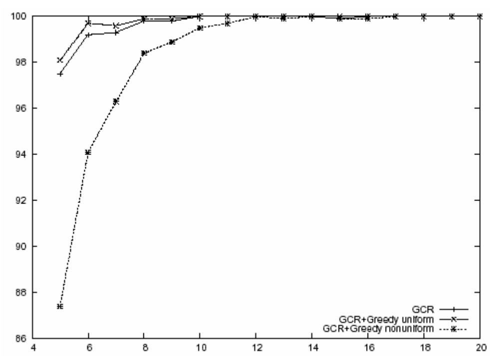


- Improved Gabriel Graph Construction: Preserve edge only if preserved also in C
  - Beacon message increased by additional 18+18 Bits
    - Announce all outgoing planar graph edges to all adjacent clusters
    - Announce all incoming planar graph edges to all cluster members
  - Beacon size remains O(1)

Correctness of Planar Graph Construction

- For cluster diameter d within (2/sqrt(7)*r,r] one can prove that the method produces planar subgraph of aggregated unit disk graph (with radius r)
- (sketch of the prove)
- Graph may be disconnected even when there is a path from source to destination!



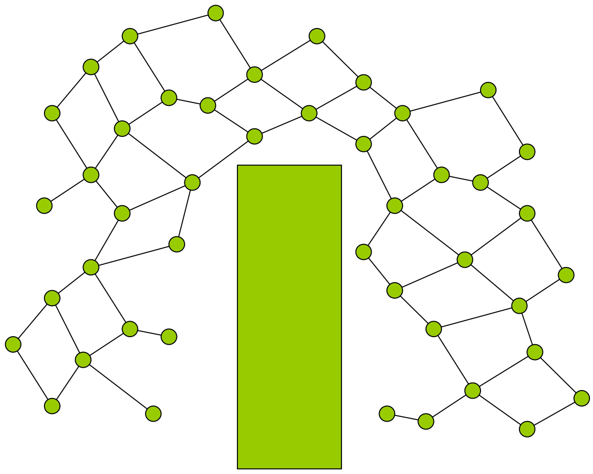


Success rate of GCR

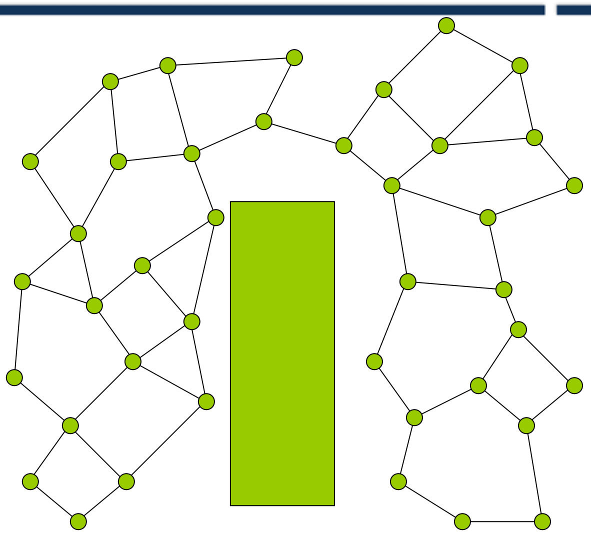
Current research issues

- Assumption: Planarity might also be proved for d within (0,r]
- Efficient local planar graph construction without disconnections
- Applying the concept of internal nodes
- The impact of mobility (see next)

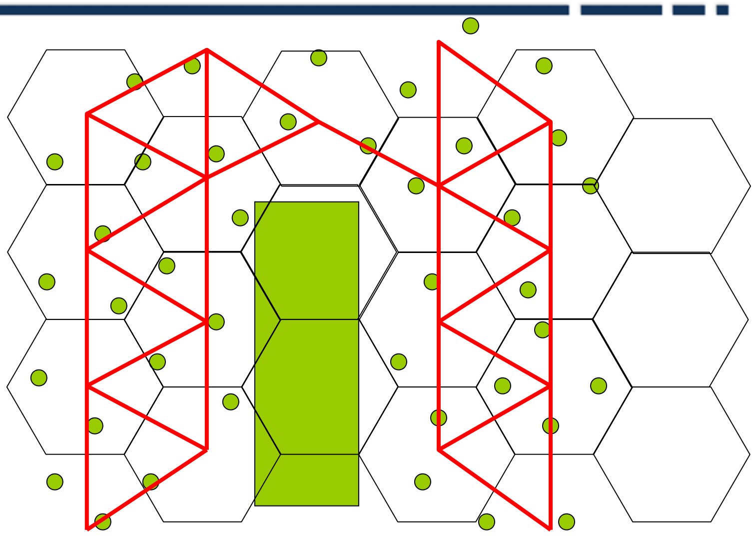
The impact of mobility: Standard Gabriel Graph



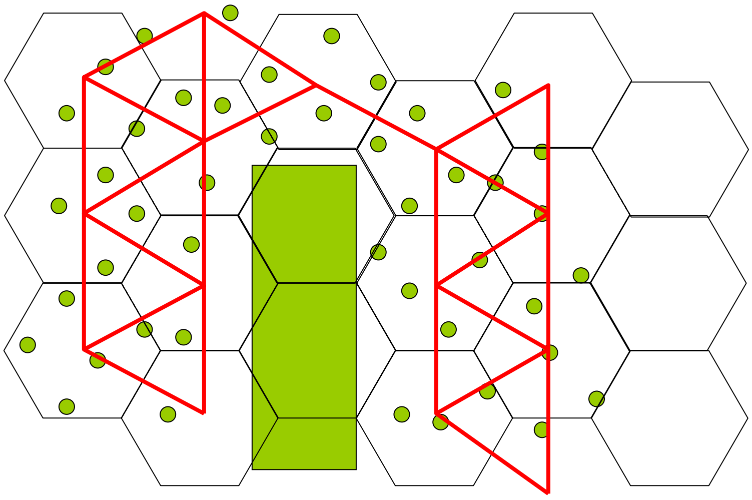
The impact of mobility: Standard Gabriel Graph



The impact of mobility: CFR



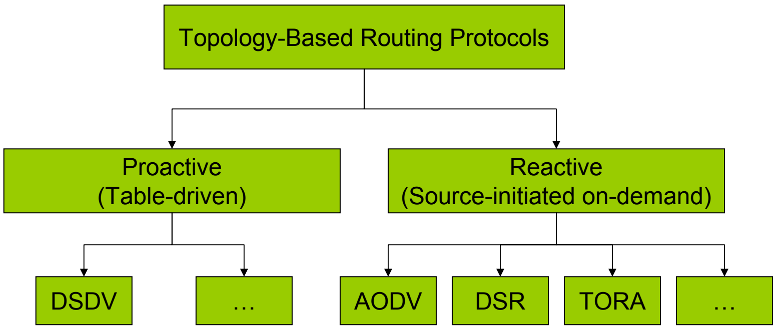
The impact of mobility: CFR



UBICOMP

Topology-Based Routing

Topology-Based Routing Overview

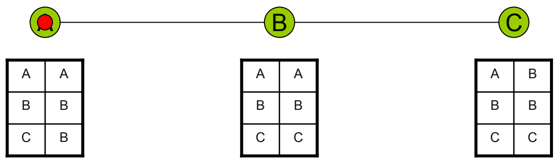


UBICOMP

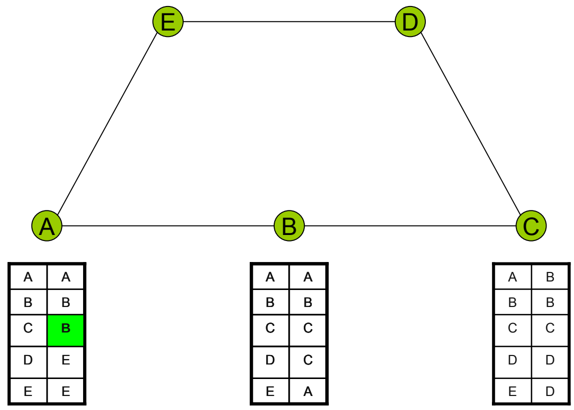
DSDV

Destination-Sequenced Distance-Vector Routing

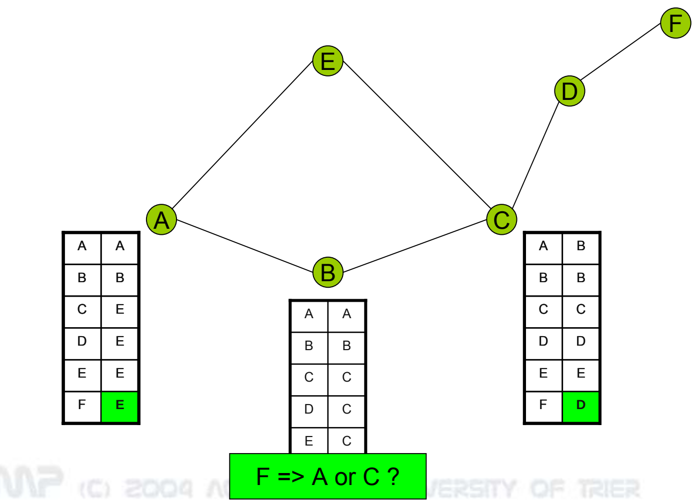
DSDV: Main Idea



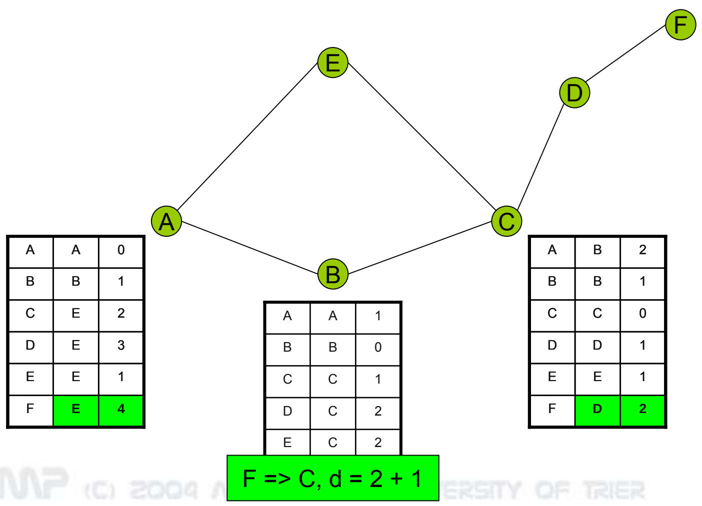
DSDV: Table Maintenance



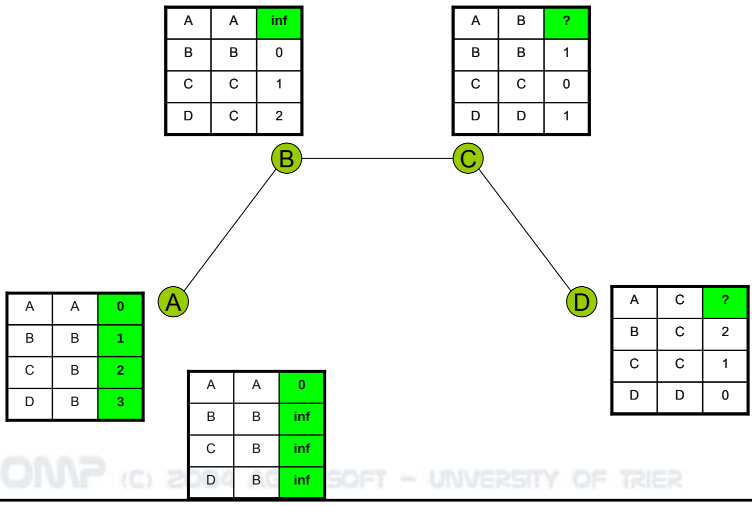
DSDV: Shortest Paths



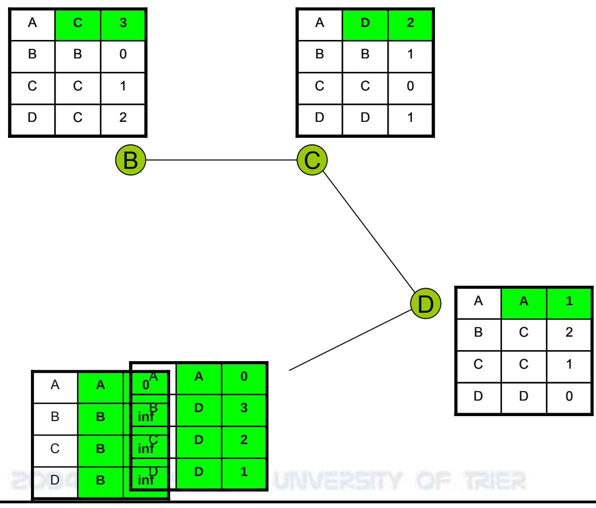
DSDV: Shortest Paths



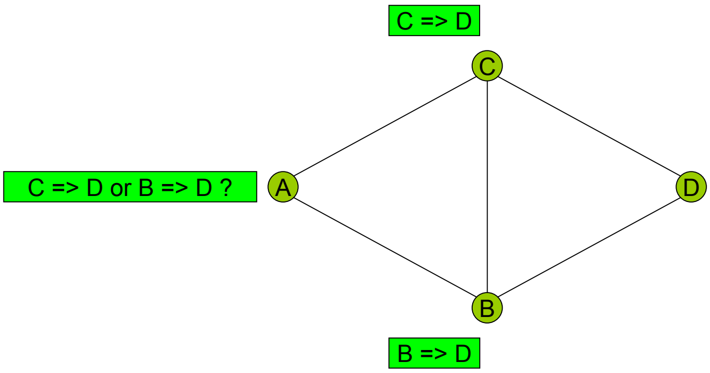
DSDV: Topology Changes



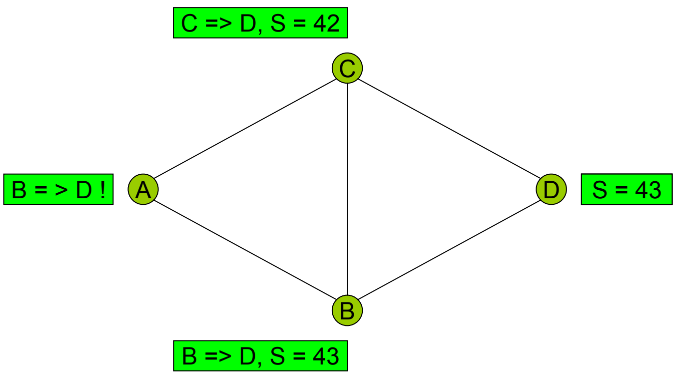
DSDV: Topology Changes



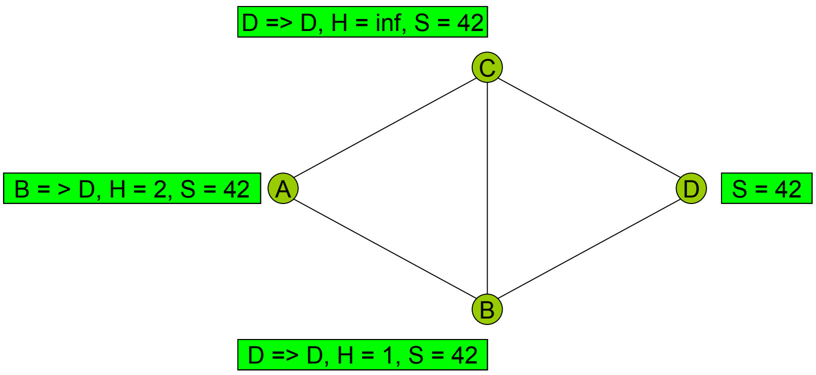
DSDV: Sequence Numbers



DSDV: Sequence Numbers

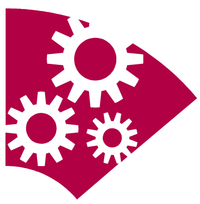


DSDV: Sequence Numbers

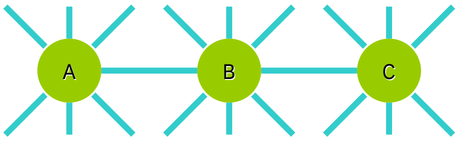


DSDV: More Facts

- Optimizations
  - Updates: Full Dump, Incremental Dump
  - Stability: Observe average transient phase
- Properties
  - Simple algorithm
  - Table stores all nodes => O(n)
  - All nodes involved
  - Sequence numbers => Loop-freedom, no *Count to Infinity Problem*



Count to Infinity



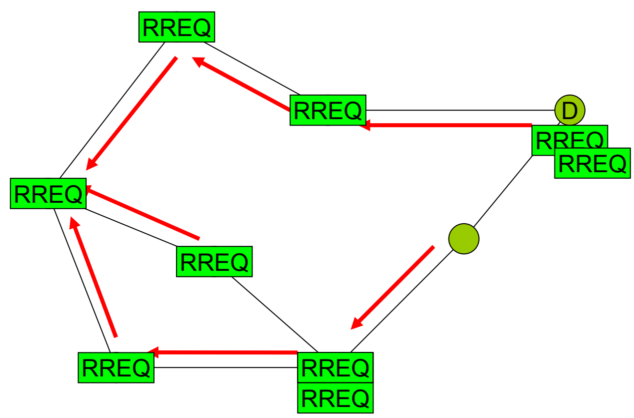
- B→C: 1 , A→C: 2, Connection from B to C disappears
- B removes DV of C, B retains DV of A
  - B concludes B→C: 3, B passes DV to all neighbors
- A receives new DV of B
  - A conjectures A→C: 4, A passes DV to all neighbors
- Etc …

UBICOMP

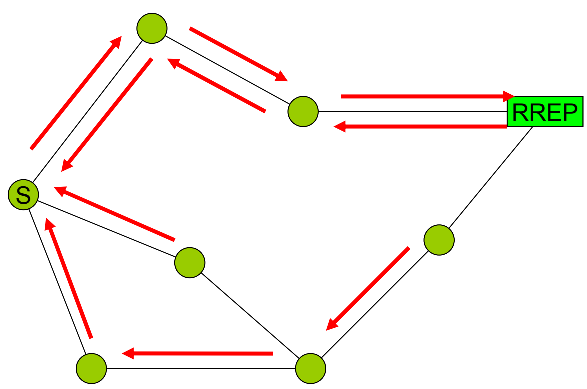
AODV

Ad-Hoc On-Demand Distance Vector Routing

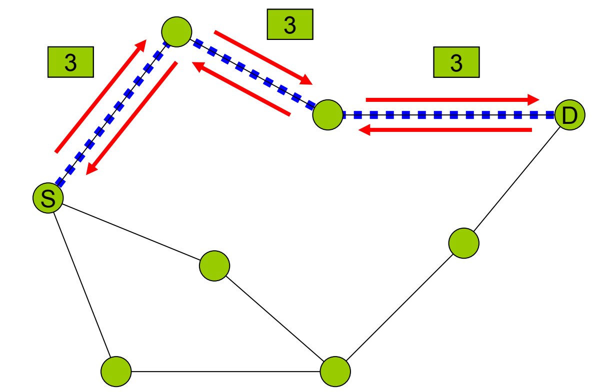
AODV: Main Idea



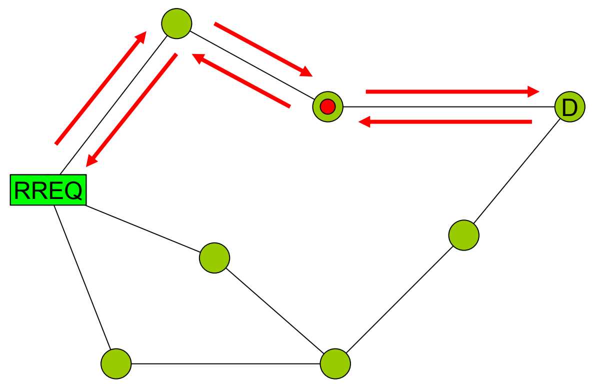
AODV: Main Idea



AODV: Main Idea

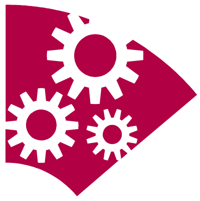


AODV: Node Mobility



AODV: More Facts

- Additional Components
  - Use of existing Paths
  - Sequence Numbers like used for DSDV
    - Loop-free
    - Up-to-date
- Advantage opposed to DSDV



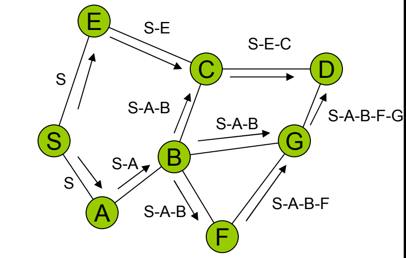
UBICOMP

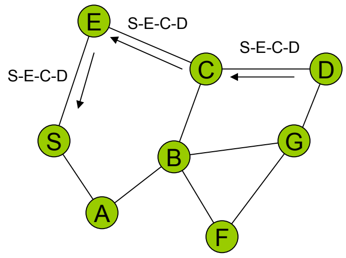
DSR

Dynamic Source Routing

DSR: Route Discovery

- Initiator broadcasts route request
- Accumulate intermediate hops in route request
- Each node processes the following steps
  - If request received before → discard
  - If host listed in request → discard
  - If destination reached → return route reply (reverse route, known path, or piggyback on route request)
  - Otherwise, append node id and rebroadcast
- Intermediate nodes cache all paths they overhear



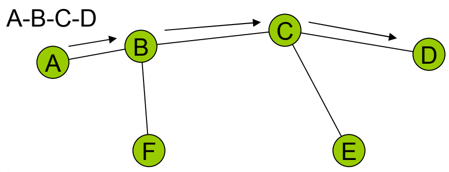


DSR: Route Maintenance

- If hop-by-hop acknowledgement provided
  - Node at broken link replies route error packet to initiator
  - Intermediate nodes remove hop in error and truncate paths at that point
- If not provided
  - Promiscuous mode (passive acknowledgment)
  - Upper layer acknowledges
- Returning the error packet
  - Cached route
  - Path reversal
  - Piggybacking
  - Explicit route discovery
- End-to-end acknowledgment if links are asymmetric

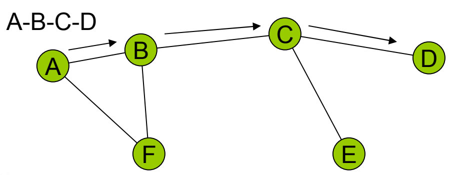
DSR: Route Cache (1)

- Route Cache organized as a tree originating in current node
  - E.g. route discovery to from A to D also reveals paths to B and C
- Node cache increases with every new learned route
  - Intermediate nodes inspect the full path stored in forwarded packet or route reply packet
    - E.g. node B learns path B-C-D from A when forwarding packet to D
  - Overhear transmissions of neighbor nodes in promiscuous mode
- Route cache used for premature route reply
  - E.g. node F will receive cache entry of node B



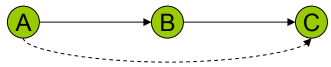
DSR: Route Cache (2)

- Problem Collision due to multiple replies
  - E.g. Node F receives cache entries from both node A and B
  - Solution delay depending on path length → short paths are favored
- Avoiding loops
  - E.g. B asking A for a path to D
- Optimization nonpropagating route requests



DSR: More Facts

- Further Optimizations
  - Piggybacking of route replies and error packets
  - Reflecting shorter routes using promiscuous mode



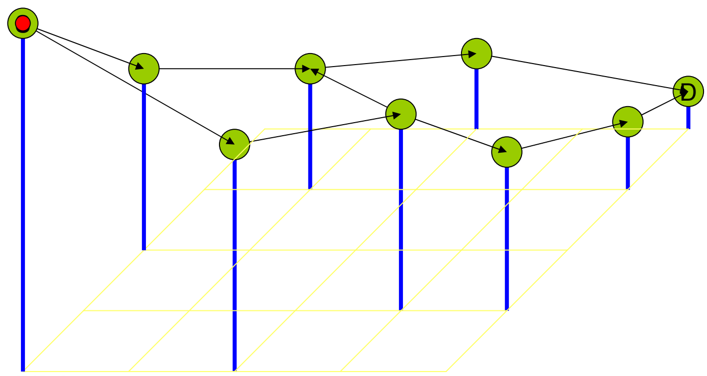
  - Improved error handling: promiscuous mode, forward error message from source back to the error node
- Advantage over hop-by-hop routing
  - Nodes do not need to maintain up-to-date routing information
- Drawback of packet carrying complete ordered list of forwarding nodes
  - scalability

UBICOMP

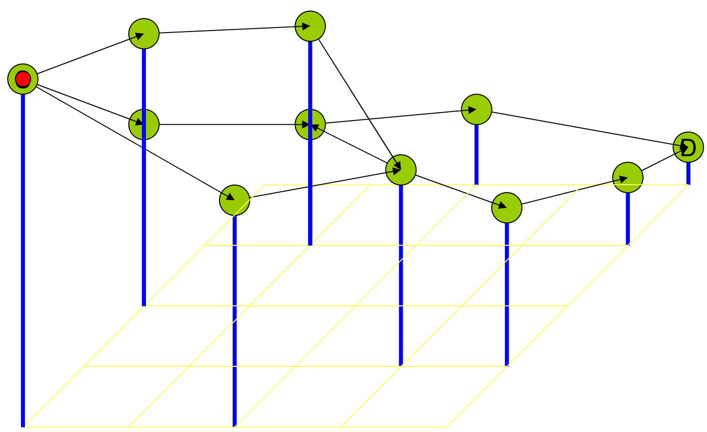
TORA

Temporally Ordered Routing Algorithm

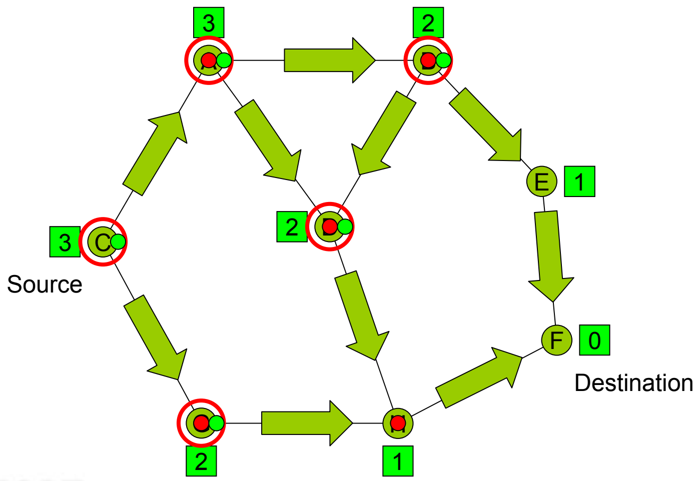
TORA: Main Idea



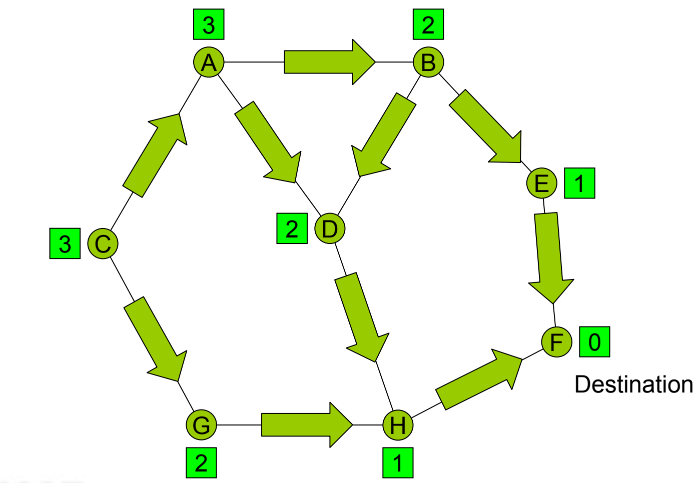
TORA: Main Idea



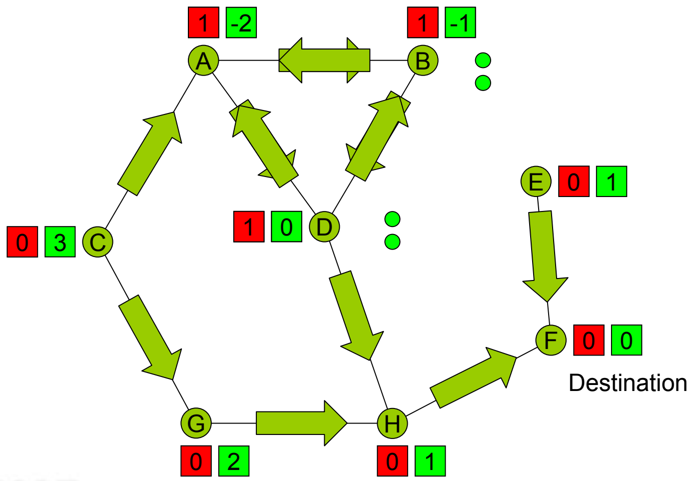
TORA: Route Construction



TORA: Link Error Without any Effect

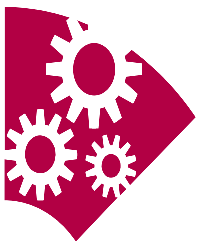


TORA: Link Error having an Effect



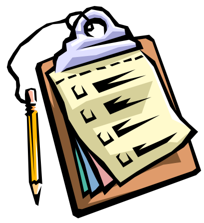
TORA: More Facts

- Removal of routes possible when network gets partitioned
- Complete metric for one node:
  - Reference level
    - Logical Clock Value of the link failure
    - ID of reference level initiator
    - Reflection Indicator
  - Delta
    - Propagation Ordering Parameter
    - ID of node



Conclusion

- Protocol overhead depends on topology changes
- Proactive vs. reactive
  - Message latency
  - Number of messages
  - Involved nodes
- Link reversal vs. distance vector
  - Involved nodes
  - Alternative paths
  - Effect of disappearing links



UBICOMP

Summary

Geographic vs. Topology-Based Routing

- Advantage opposed to topology based routing
  - Stateless routing possible (expect of neighbor list)
  - May work even for highly dynamic network topology
- Problems
  - Localization of own position → GPS
  - Localization of other nodes → location service
  - Greedy routing failures