Wireless Networks: Routing and Transport

Luciano Bononi

Dipartimento di Informatica

Università di Bologna

e-mail: bononi@cs.unibo.it

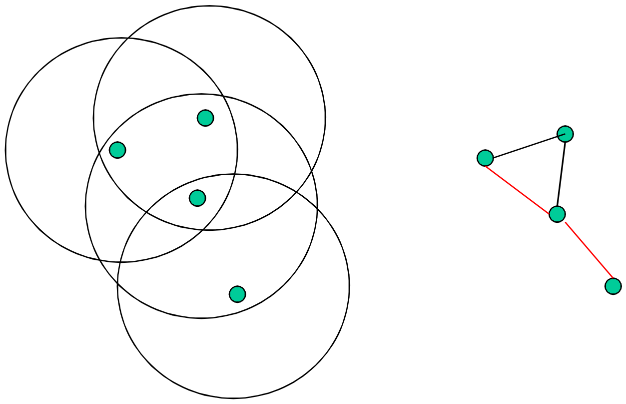
Credits: some slide-content and figures have been taken from slides found on the web by the following authors:
Nitin Vaidya (uiuc), J. Kurose & K. Ross (Computer Networking book)

Mobile Ad Hoc Networks

- Formed by wireless hosts which may be mobile
- Without (necessarily) using a pre-existing infrastructure
- Routes between nodes may potentially contain multiple hops

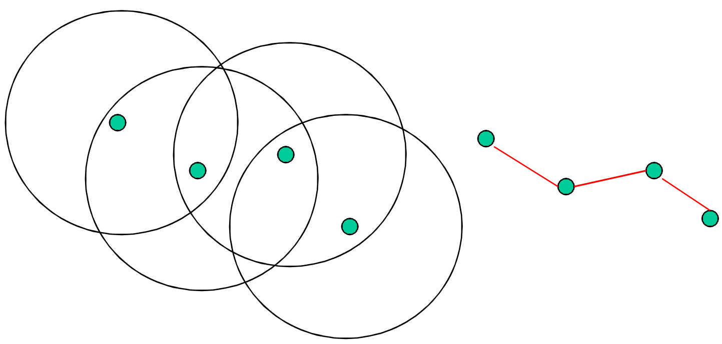
Mobile Ad Hoc Networks

- May need to traverse multiple links to reach a destination



Mobile Ad Hoc Networks (MANET)

- Mobility causes route changes



Unicast Routing
in
Mobile Ad Hoc Networks

Why is Routing in MANET different ?

- **Host mobility**
  - link failure/repair due to mobility may have different characteristics than those due to other causes
- **Rate of link failure/repair may be high**
  - nodes move fast?
- **New performance criteria may be used**
  - route stability vs. mobility
  - energy consumption

Unicast Routing Protocols

- Many protocols have been proposed
  - Some have been invented specifically for MANET
  - Others are adapted from previously proposed protocols for wired networks
- No single protocol works well in all environments
  - some attempts made to develop adaptive protocols

Routing Protocols

- Proactive protocols
  - Determine routes independent of traffic pattern
  - Traditional link-state and distance-vector routing protocols are proactive
- Reactive protocols
  - Maintain routes only if needed
- Hybrid protocols

Trade-Off

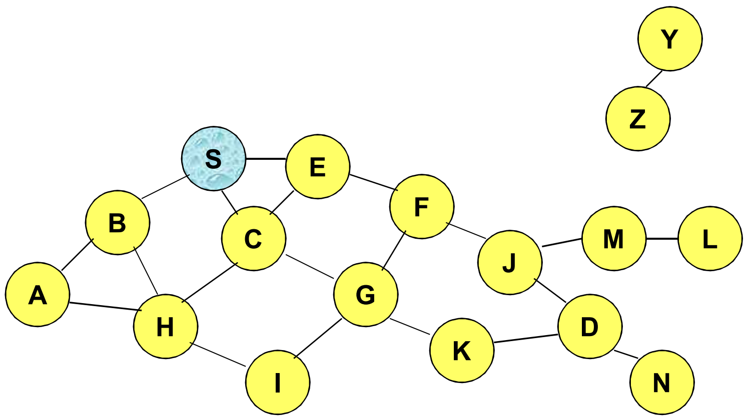
- Latency of route discovery
  - Proactive protocols may have lower latency since routes are maintained at all times
  - Reactive protocols may have higher latency because a route from X to Y will be found only when X attempts to send to Y
- Overhead of route discovery/maintenance
  - Reactive protocols may have lower overhead since routes are determined only if needed
  - Proactive protocols can (but not necessarily) result in higher overhead due to continuous route updating
- Which approach achieves a better trade-off depends on the traffic and mobility patterns

Overview of Unicast Routing Protocols

Flooding for Data Delivery

- Sender S broadcasts data packet P to all its neighbors
- Each node receiving P forwards P to its neighbors
- Sequence numbers used to avoid the possibility of forwarding the same packet more than once
- Packet P reaches destination D provided that D is reachable from sender S
- Node D does not forward the packet

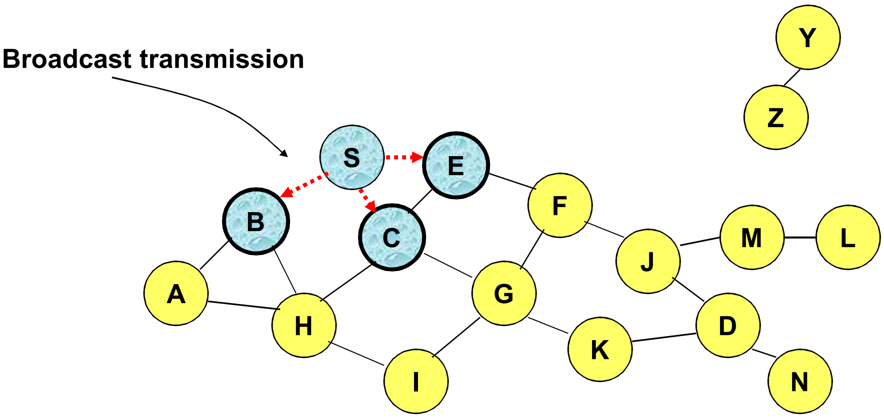
Flooding for Data Delivery



Represents a node that has received packet P

Represents that connected nodes are within each other's transmission range

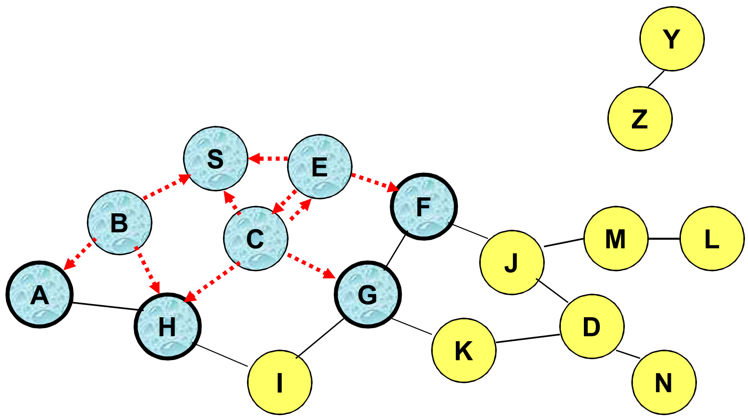
Flooding for Data Delivery



Represents a node that receives packet P for the first time

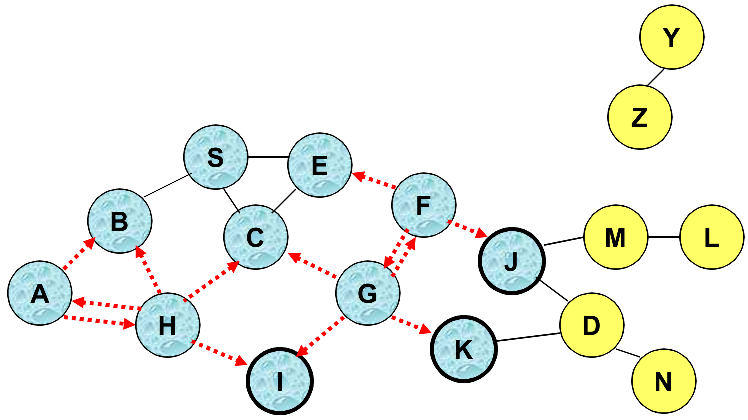
Represents transmission of packet P

Flooding for Data Delivery



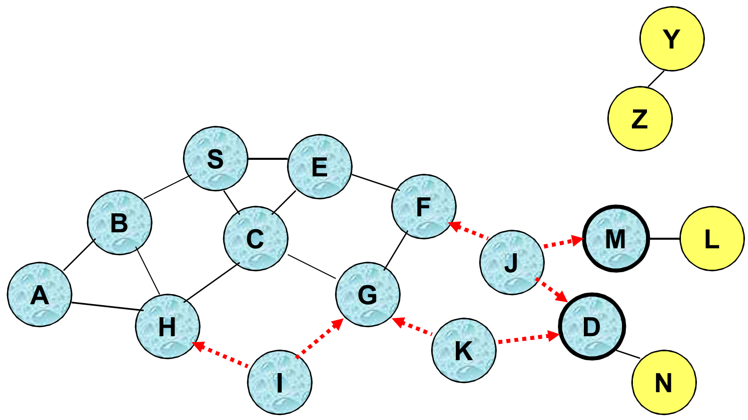
- Node H receives packet P from two neighbors: potential for collision

Flooding for Data Delivery



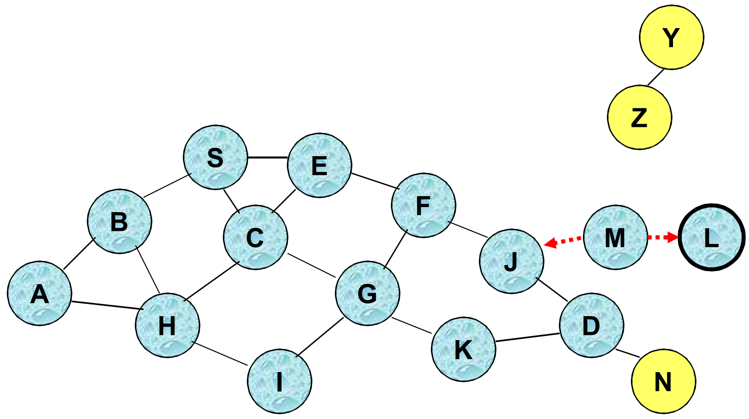
- Node C receives packet P from G and H, but does not forward it again, because node C has already forwarded packet P once

Flooding for Data Delivery



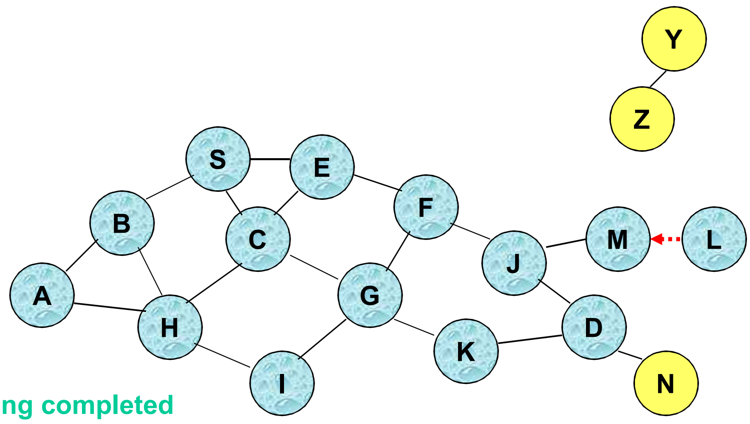
- Nodes J and K both broadcast packet P to node D
- Since nodes J and K are hidden from each other, their transmissions may collide
  => Packet P may not be delivered to node D at all, despite the use of flooding

Flooding for Data Delivery



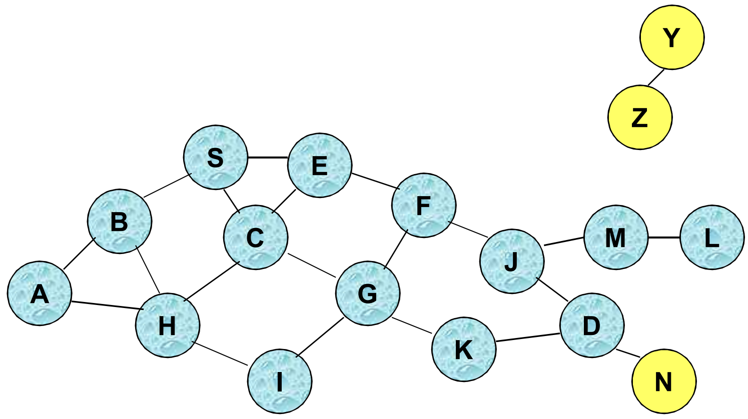
- Node D does not forward packet P, because node D is the intended destination of packet P

Flooding for Data Delivery



- Flooding completed
- Nodes unreachable from S do not receive packet P (e.g., node Z)
- Nodes for which all paths from S go through the destination D also do not receive packet P (example: node N)

Flooding for Data Delivery



• Flooding may deliver packets to too many nodes (in the worst case, all nodes reachable from sender may receive the packet)

Flooding for Data Delivery: Advantages

- Simplicity
- May be more efficient than other protocols when rate of information transmission is low enough that the overhead of explicit route discovery/maintenance incurred by other protocols is relatively higher
  - this scenario may occur, for instance, when nodes transmit small data packets relatively infrequently, and many topology changes occur between consecutive packet transmissions
- Potentially higher reliability of data delivery
  - Because packets may be delivered to the destination on multiple paths

Flooding for Data Delivery: Disadvantages

- Potentially, very high overhead
  - Data packets may be delivered to too many nodes who do not need to receive them
- Potentially lower reliability of data delivery
  - Flooding uses broadcasting -- hard to implement reliable broadcast delivery without significantly increasing overhead
    - Broadcasting in IEEE 802.11 MAC is unreliable
  - In our example, nodes J and K may transmit to node D simultaneously, resulting in loss of the packet
    - in this case, destination would not receive the packet at all

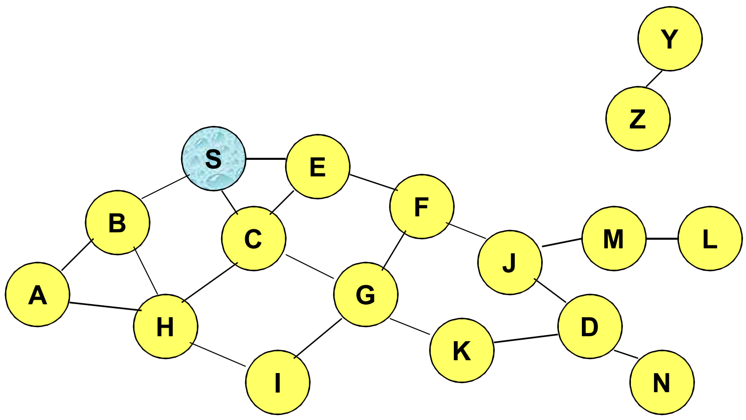
Flooding of Control Packets

- Many protocols perform (potentially *limited*) flooding of control packets, instead of data packets
- The control packets are used to discover routes
- Discovered routes are subsequently used to send data packet(s)
- Overhead of control packet flooding is amortized over data packets transmitted between consecutive control packet floods

Dynamic Source Routing (DSR) [Johnson96]

- When node S wants to send a packet to node D, but does not know a route to D, node S initiates a route discovery
- Source node S floods Route Request (RREQ)
- Each node appends own identifier when forwarding RREQ

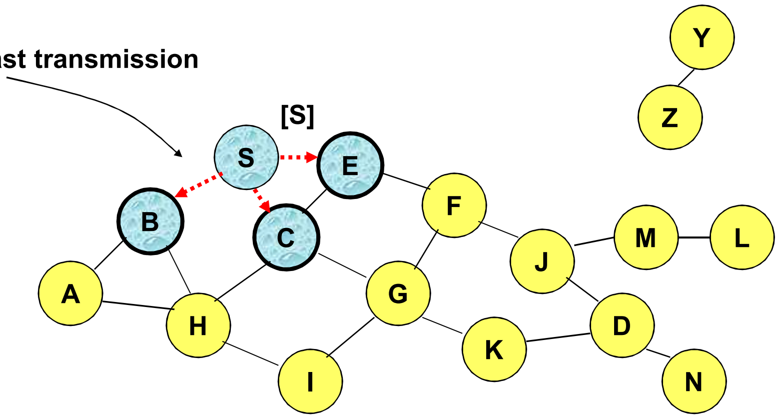
Route Discovery in DSR




Represents a node that has received RREQ for D from S

Route Discovery in DSR

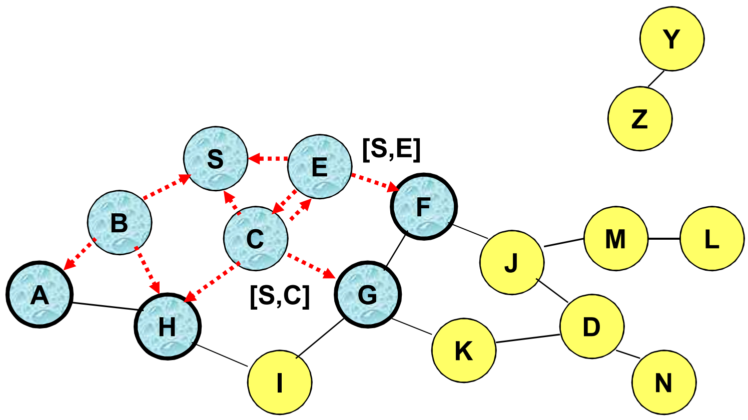
Broadcast transmission



Represents transmission of RREQ

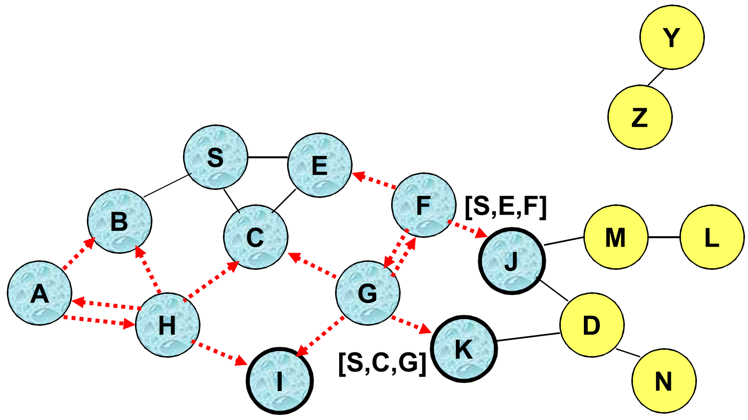
[X,Y] Represents list of identifiers appended to RREQ

Route Discovery in DSR



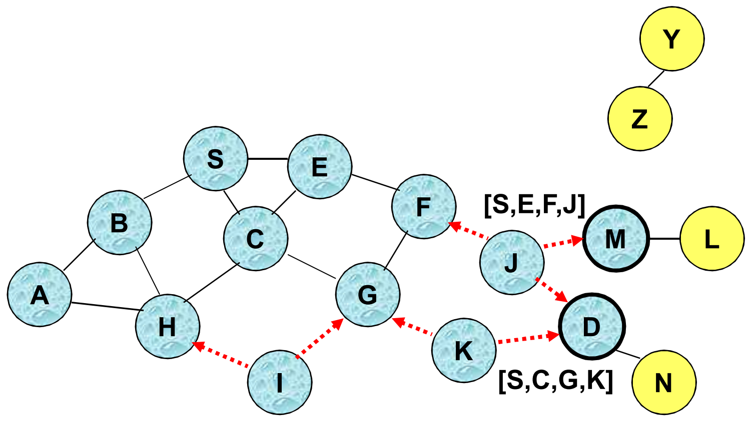
- Node H receives packet RREQ from two neighbors: potential for collision

Route Discovery in DSR



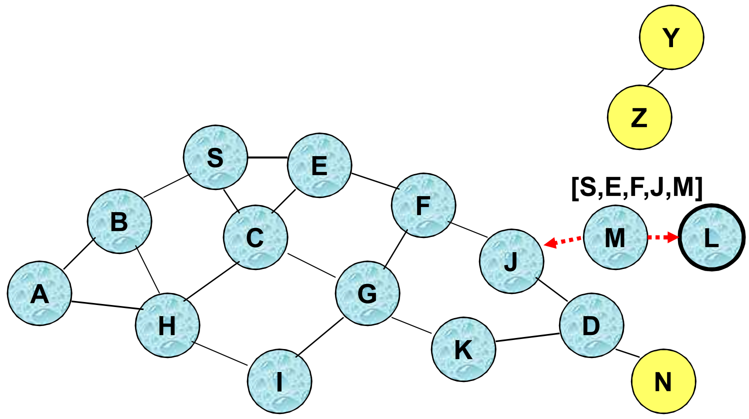
- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

Route Discovery in DSR



- Nodes J and K both broadcast RREQ to node D
- Since nodes J and K are hidden from each other, their transmissions may collide

Route Discovery in DSR

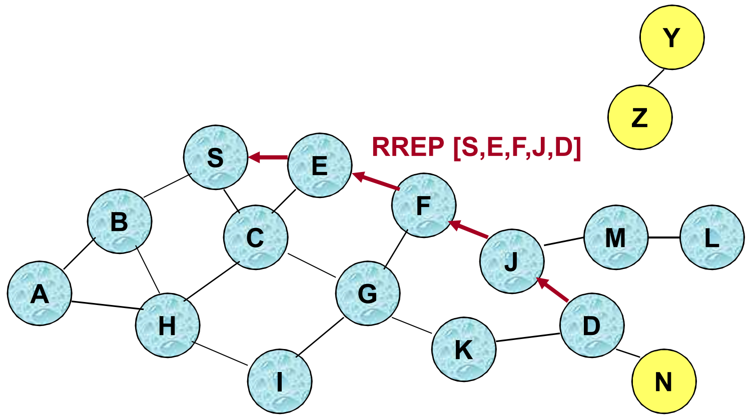


• Node D does not forward RREQ, because node D is the intended target of the route discovery

Route Discovery in DSR

- Destination D on receiving the first RREQ, sends a Route Reply (RREP)
- RREP is sent on a route obtained by reversing the route appended to received RREQ
- RREP includes the route from S to D on which RREQ was received by node D

Route Reply in DSR



Represents RREP control message

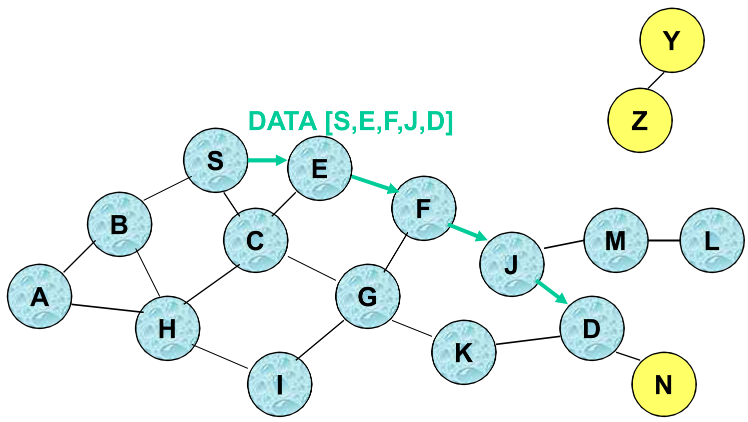
Route Reply in DSR

- Route Reply can be sent by reversing the route in Route Request (RREQ) only if links are guaranteed to be bi-directional
  - To ensure this, RREQ should be forwarded only if it received on a link that is known to be bi-directional
- If unidirectional (asymmetric) links are allowed, then RREP may need a route discovery for S from node D
  - Unless node D already knows a route to node S
  - If a route discovery is initiated by D for a route to S, then the Route Reply is piggybacked on the Route Request from D.
- If IEEE 802.11 MAC is used to send data, then links have to be bi-directional (since Ack is used)

Dynamic Source Routing (DSR)

- Node S on receiving RREP, caches the route included in the RREP
- When node S sends a data packet to D, the entire route is included in the packet header
  - hence the name source routing
- Intermediate nodes use the source route included in a packet to determine to whom a packet should be forwarded

Data Delivery in DSR



Packet header size grows with route length

When to Perform a Route Discovery

- When node S wants to send data to node D, but does not know a valid route node D

DSR Optimization: Route Caching

- Each node caches a new route it learns by *any means*
- When node S finds route [S,E,F,J,D] to node D, node S also learns route [S,E,F] to node F
- When node K receives Route Request [S,C,G] destined for node, node K learns route [K,G,C,S] to node S
- When node F forwards Route Reply RREP [S,E,F,J,D], node F learns route [F,J,D] to node D
- When node E forwards Data [S,E,F,J,D] it learns route [E,F,J,D] to node D
- A node may also learn a route when it overhears Data packets

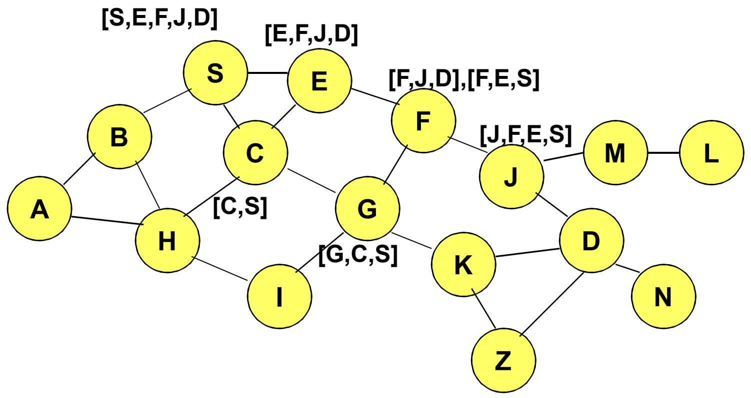
Use of Route Caching

- When node S learns that a route to node D is broken, it uses another route from its local cache, if such a route to D exists in its cache. Otherwise, node S initiates route discovery by sending a route request

- Node X on receiving a Route Request for some node D can send a Route Reply if node X knows a route to node D

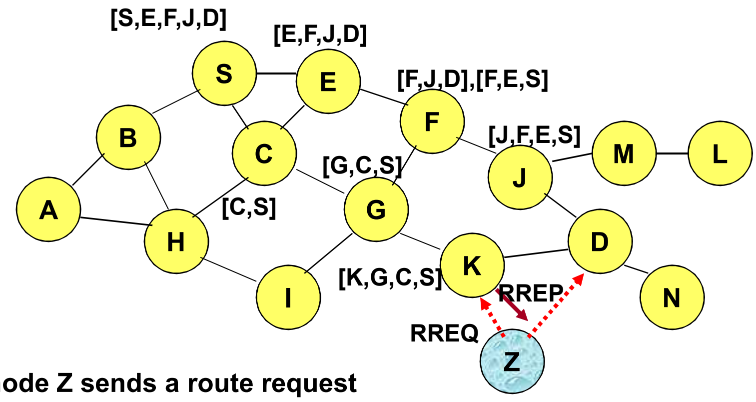
- Use of route cache
  - can speed up route discovery
  - can reduce propagation of route requests

Use of Route Caching



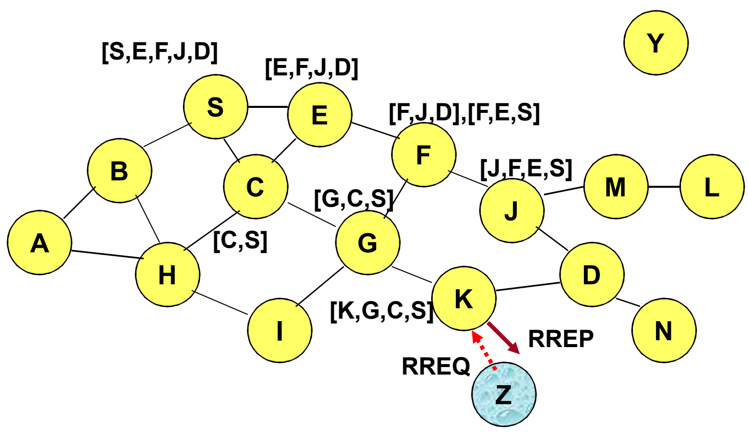
[P,Q,R] Represents cached route at a node
(DSR maintains the cached routes in a tree format)

Use of Route Caching:
Can Speed up Route Discovery



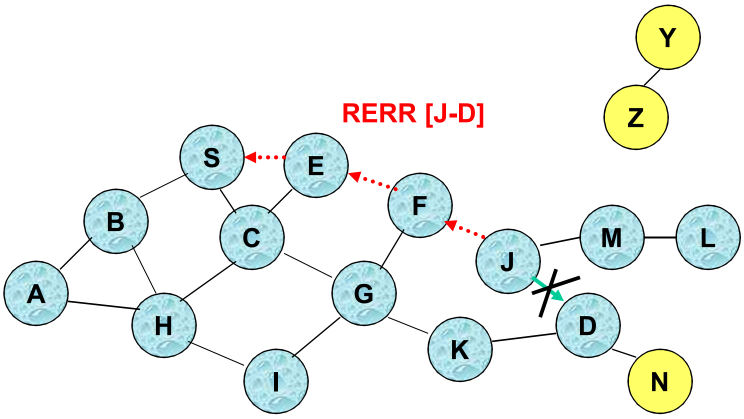
When node Z sends a route request for node C, node K sends back a route reply [Z,K,G,C] to node Z using a locally cached route

Use of Route Caching:
Can Reduce Propagation of Route Requests



Assume that there is no link between D and Z.
Route Reply (RREP) from node K limits flooding of RREQ.
In general, the reduction may be less dramatic.

Route Error (RERR)



J sends a route error to S along route J-F-E-S when its attempt to forward the data packet S (with route SEFJD) on J-D fails

Nodes hearing RERR update their route cache to remove link J-D

Route Caching: Beware!

- Stale caches can adversely affect performance
- With passage of time and host mobility, cached routes may become invalid
- A sender host may try several stale routes (obtained from local cache, or replied from cache by other nodes), before finding a good route
- An illustration of the adverse impact on TCP will be discussed later in the tutorial [Holland99]

Dynamic Source Routing: Advantages

- Routes maintained only between nodes who need to communicate
  - reduces overhead of route maintenance
- Route caching can further reduce route discovery overhead
- A single route discovery may yield many routes to the destination, due to intermediate nodes replying from local caches

Dynamic Source Routing: Disadvantages

- Packet header size grows with route length due to source routing
- Flood of route requests may potentially reach all nodes in the network
- Care must be taken to avoid collisions between route requests propagated by neighboring nodes
  - insertion of random delays before forwarding RREQ
- Increased contention if too many route replies come back due to nodes replying using their local cache
  - Route Reply *Storm* problem
  - Reply storm may be eased by preventing a node from sending RREP if it hears another RREP with a shorter route

Dynamic Source Routing: Disadvantages

- An intermediate node may send Route Reply using a stale cached route, thus polluting other caches
- This problem can be eased if some mechanism to purge (potentially) invalid cached routes is incorporated.
- For some proposals for cache invalidation, see [Hu00Mobicom]
  - Static timeouts
  - Adaptive timeouts based on link stability

Flooding of Control Packets

- How to reduce the scope of the route request flood ?
  - LAR [Ko98Mobicom]
  - Query localization [Castaneda99Mobicom]
- How to reduce redundant broadcasts ?
  - The Broadcast Storm Problem [Ni99Mobicom]

Location-Aided Routing (LAR) [Ko98Mobicom]

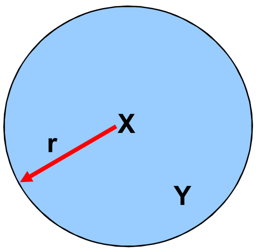
- Exploits location information to limit scope of route request flood
  - Location information may be obtained using GPS
- *Expected Zone* is determined as a region that is expected to hold the current location of the destination
  - Expected region determined based on potentially old location information, and knowledge of the destination's speed
- Route requests limited to a *Request Zone* that contains the Expected Zone and location of the sender node

Expected Zone in LAR

X = last known location of node D, at time t0

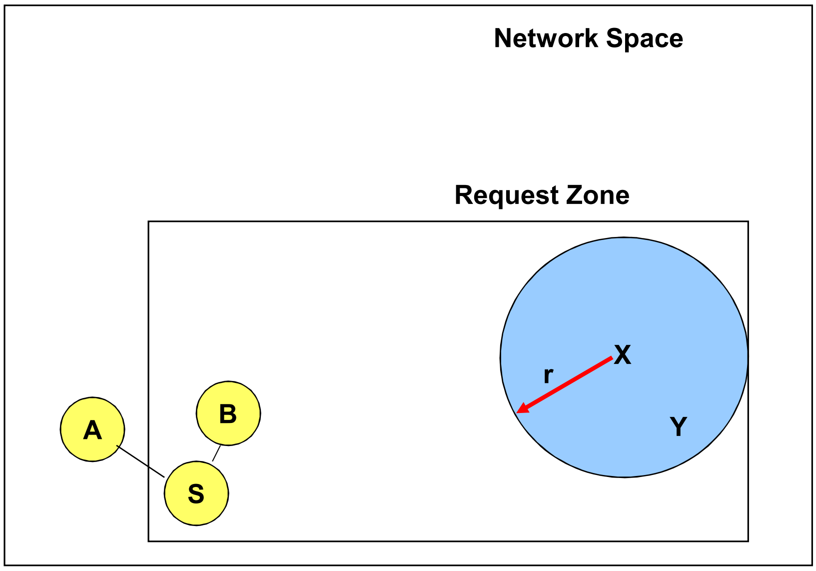
Y = location of node D at current time t1, unknown to node S

r = (t1 - t0) * estimate of D's speed



Expected Zone

Request Zone in LAR



LAR

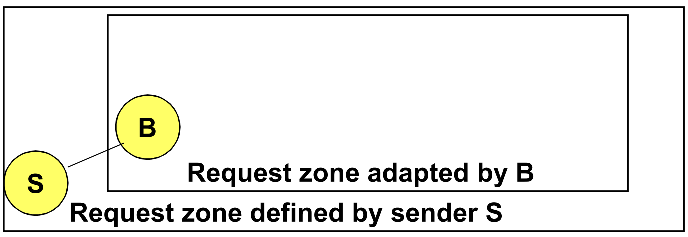
- Only nodes within the request zone forward route requests
  - Node A does not forward RREQ, but node B does (see previous slide)
- Request zone explicitly specified in the route request
- Each node must know its physical location to determine whether it is within the request zone

LAR

- Only nodes within the request zone forward route requests
- If route discovery using the smaller request zone fails to find a route, the sender initiates another route discovery (after a timeout) using a larger request zone
  - the larger request zone may be the entire network
- Rest of route discovery protocol similar to DSR

LAR Variations: Adaptive Request Zone

- Each node may modify the request zone included in the forwarded request
- Modified request zone may be determined using more recent/accurate information, and may be smaller than the original request zone



LAR Variations: Implicit Request Zone

- In the previous scheme, a route request explicitly specified a request zone
- Alternative approach: A node X forwards a route request received from Y if node X is deemed to be closer to the expected zone as compared to Y
- The motivation is to attempt to bring the route request physically closer to the destination node after each forwarding

Location-Aided Routing

- The basic proposal assumes that, *initially*, location information for node X becomes known to Y 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

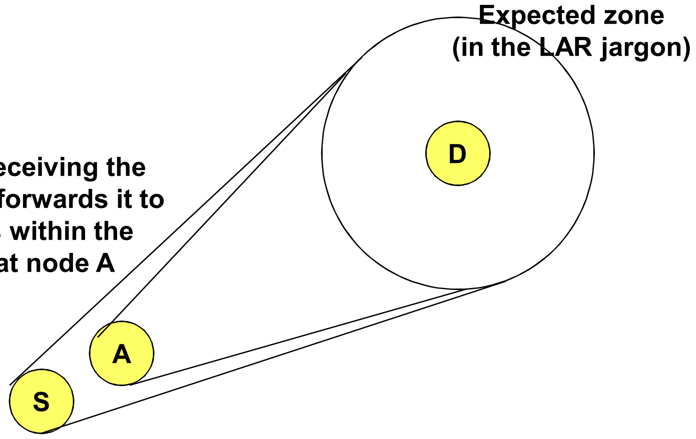
- Location information can also be piggybacked on any message from Y to X
- Y may also proactively distribute its location information
  - Similar to other protocols discussed later (e.g., DREAM, GLS)

Location Aided Routing (LAR)

- Advantages
  - reduces the scope of route request flood
  - reduces overhead of route discovery
- Disadvantages
  - Nodes need to know their physical locations
  - Does not take into account possible existence of obstructions for radio transmissions

Distance Routing Effect Algorithm for Mobility (DREAM)

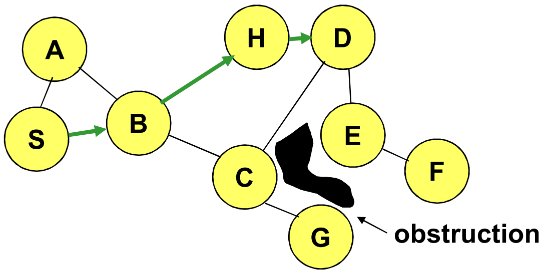
Node A, on receiving the data packet, forwards it to its neighbors within the cone rooted at node A



S sends *data packet* to all neighbors in the cone rooted at node S

Geographic Distance Routing (GEDIR) [Lin98]

- Location of the destination node is assumed known
- Each node knows location of its neighbors
- Each node forwards a packet to its neighbor closest to the destination
- Route taken from S to D shown below



Routing with Guaranteed Delivery [Bose99Dialm]

- Improves on GEDIR [Lin98]
- Guarantees delivery (using location information) provided that a path exists from source to destination
- Routes around obstacles if necessary
- A similar idea also appears in [Karp00Mobicom]

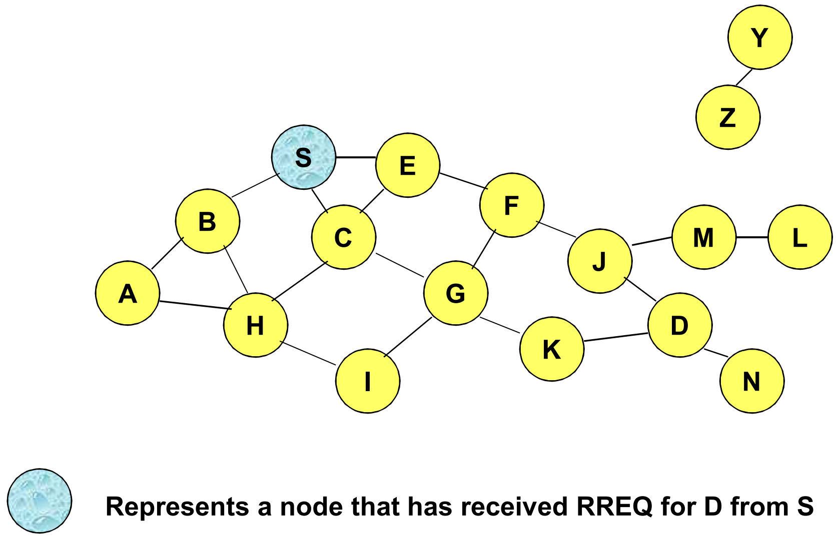
Ad Hoc On-Demand Distance Vector Routing (AODV) [Perkins99Wmcsa]

- DSR includes source routes in packet headers
- Resulting large headers can sometimes degrade performance
  - particularly when data contents of a packet are small
- AODV attempts to improve on DSR by maintaining routing tables at the nodes, so that data packets do not have to contain routes
- AODV retains the desirable feature of DSR that routes are maintained only between nodes which need to communicate

AODV

- Route Requests (RREQ) are forwarded in a manner similar to DSR
- When a node re-broadcasts a Route Request, it sets up a reverse path pointing towards the source
  - AODV assumes symmetric (bi-directional) links
- When the intended destination receives a Route Request, it replies by sending a Route Reply
- Route Reply travels along the reverse path set-up when Route Request is forwarded

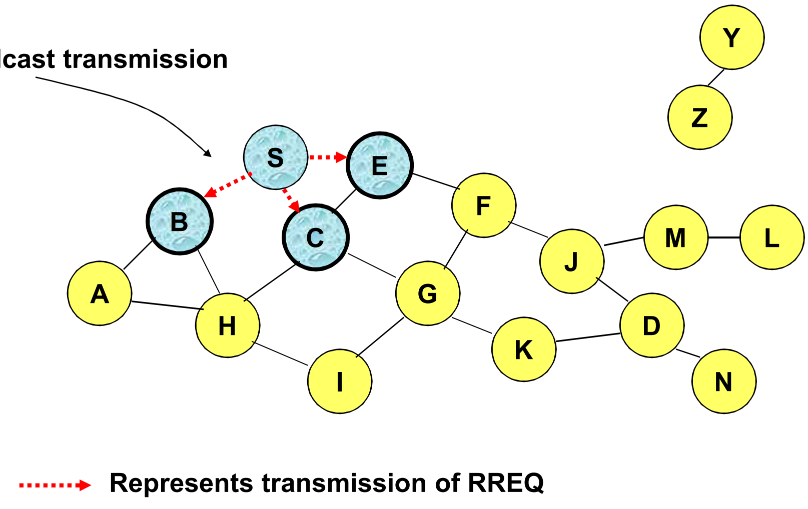
Route Requests in AODV



Represents a node that has received RREQ for D from S

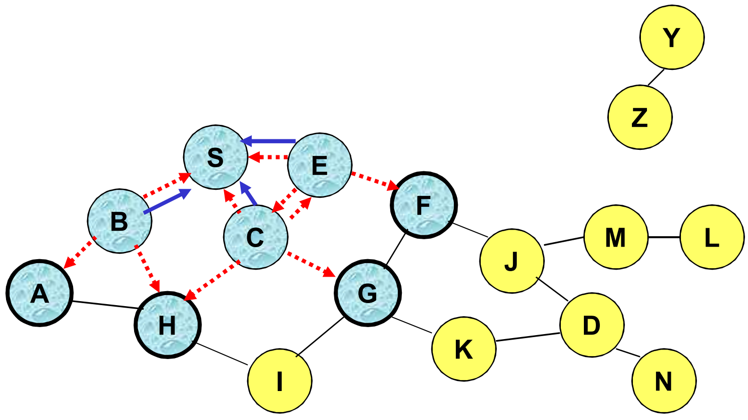
Route Requests in AODV

Broadcast transmission



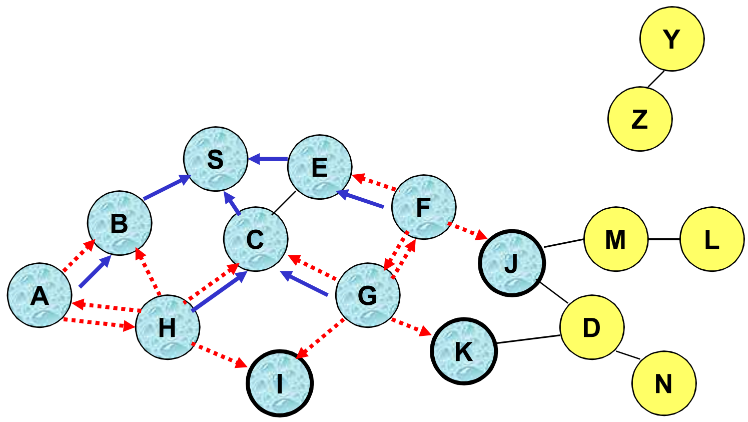
Represents transmission of RREQ

Route Requests in AODV



63

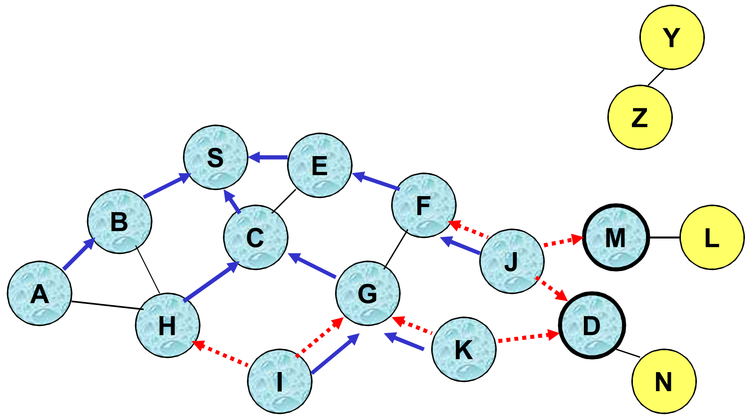
Reverse Path Setup in AODV



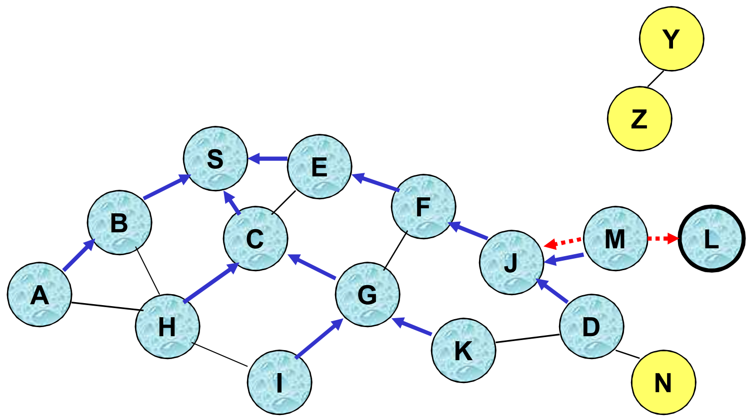
- Node C receives RREQ from G and H, but does not forward it again, because node C has already forwarded RREQ once

64

Reverse Path Setup in AODV

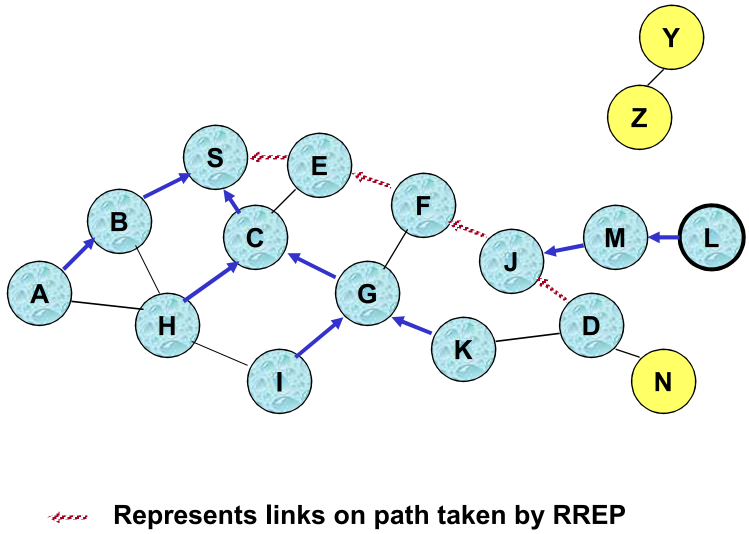


Reverse Path Setup in AODV



- Node D does not forward RREQ, because node D is the intended target of the RREQ

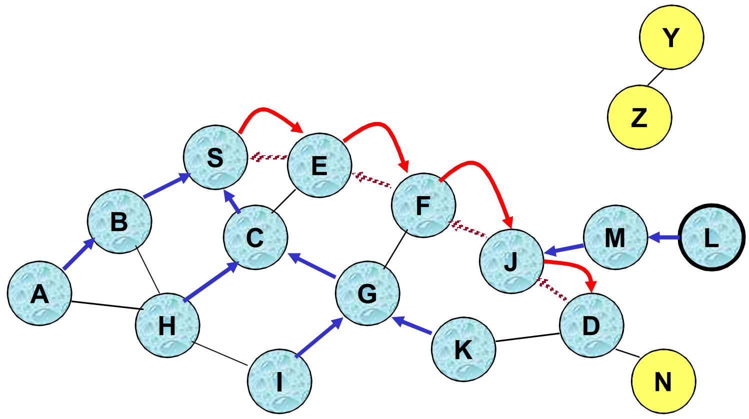
Route Reply in AODV



Route Reply in AODV

- An intermediate node (not the destination) may also send a Route Reply (RREP) provided that it knows a more recent path than the one previously known to sender S
- To determine whether the path known to an intermediate node is more recent, *destination sequence numbers* are used
- The likelihood that an intermediate node will send a Route Reply when using AODV not as high as DSR
  - A new Route Request by node S for a destination is assigned a higher destination sequence number. An intermediate node which knows a route, but with a smaller sequence number, cannot send Route Reply

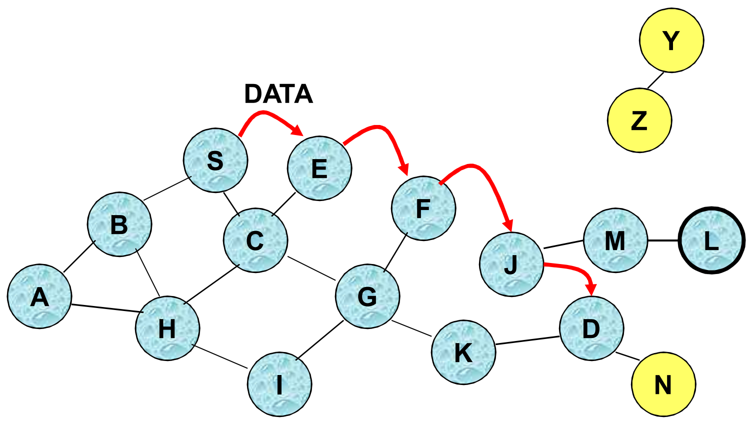
Forward Path Setup in AODV



Forward links are setup when RREP travels along the reverse path


Represents a link on the forward path

Data Delivery in AODV



Routing table entries used to forward data packet.

Route is *not* included in packet header.

Timeouts

- *A routing table entry maintaining a reverse path is purged after a timeout interval*
  - timeout should be long enough to allow RREP to come back
- *A routing table entry maintaining a forward path is purged if **not used** for a **active_route_timeout** interval*
  - if no is data being sent using a particular routing table entry, that entry will be deleted from the routing table (even if the route may actually still be valid)

Summary: AODV

- Routes need not be included in packet headers
- Nodes maintain routing tables containing entries only for routes that are in active use
- *At most one next-hop per destination maintained at each node*
  - DSR may maintain several routes for a single destination
- Unused routes expire even if topology does not change

So far ...

- All protocols discussed so far perform some form of flooding
- Now we will consider protocols which try to reduce/avoid such behavior

Proactive Protocols

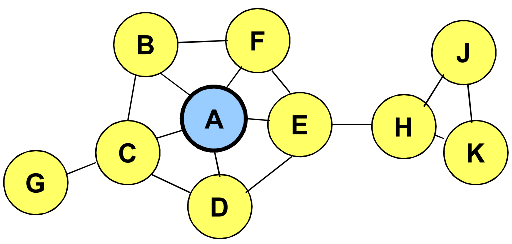
- Most of the schemes discussed so far are reactive
- Proactive schemes based on distance-vector and link-state mechanisms have also been proposed

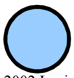
Link State Routing [Huitema95]

- Each node periodically floods status of its links
- Each node re-broadcasts link state information received from its neighbor
- Each node keeps track of link state information received from other nodes
- Each node uses above information to determine next hop to each destination

Optimized Link State Routing (OLSR)

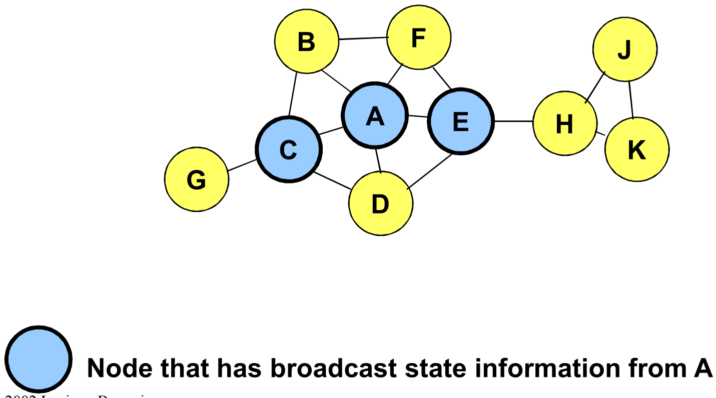
- Nodes C and E are multipoint relays of node A




Node that has broadcast state information from A

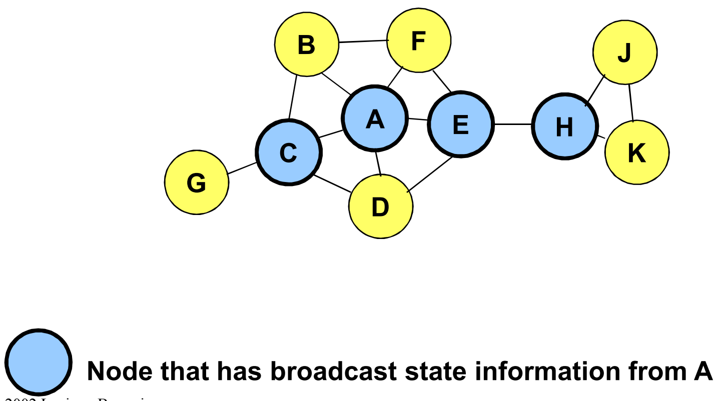
Optimized Link State Routing (OLSR)

- Nodes C and E forward information received from A



Optimized Link State Routing (OLSR)

- Nodes E and K are multipoint relays for node H
- Node K forwards information received from H
  - E has already forwarded the same information once



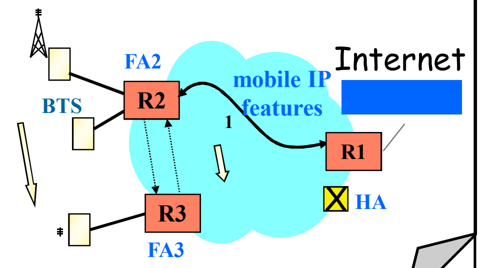
OLSR

- OLSR floods information through the multipoint relays
- The flooded itself is fir links connecting nodes to respective multipoint relays
- Routes used by OLSR only include multipoint relays as intermediate nodes

Mobile IP

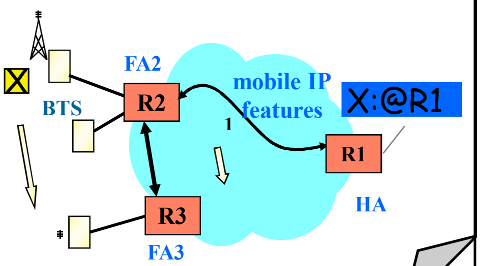
Mobile IP in Wireless Infrastructure Networks

- **Mobile IP:**
  - •X Home Agent (HA) is located in Router R1
  - •X moves to R2, then to R3, IP domains...
    - Foreign Agents FA2 and FA3 dynamically created by mobile IP in R2 and R3
    - FA2 informs HA about new IP for X
    - HA tunnels IP(x) to FA2



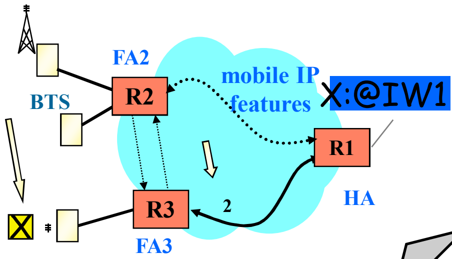
Mobile IP in Wireless Infrastructure Networks

- **Mobile IP:**
  - •X Home Agent (HA) is located in Router R1
  - •X moves to R3, from R2 IP domains...
    - FA3 informs FA2 about new IP for X
    - FA2 tunnels IP(x) to FA3
  - IP tunnel-in-IP tunnel



Mobile IP in Wireless Infrastructure Networks

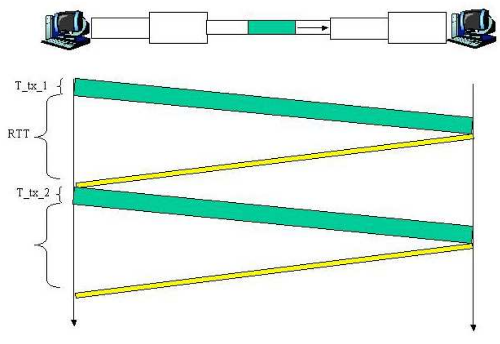
- **Mobile IP:**
  - eventually FA3 <-> HA?
    - avoids tunnel-in-tunnel
    - avoids IP triangulation



TCP on Mobile Ad Hoc Networks

Overview of
Transmission Control Protocol / Internet Protocol
(TCP/IP)

Stop & Wait protocols



Stop & Wait protocols:

Perdita e alterazione dei dati ricevuti gestita mediante:

- feedback Ack/Nak dal ricevente
- ritrasmissione immediata (se Ack/NAK è ricevuto)
- timeout per gestione feedback implicito (perdita)
- Numeri di sequenza (per disambiguare ritrasmissione in seguito a perdita dell'Ack)

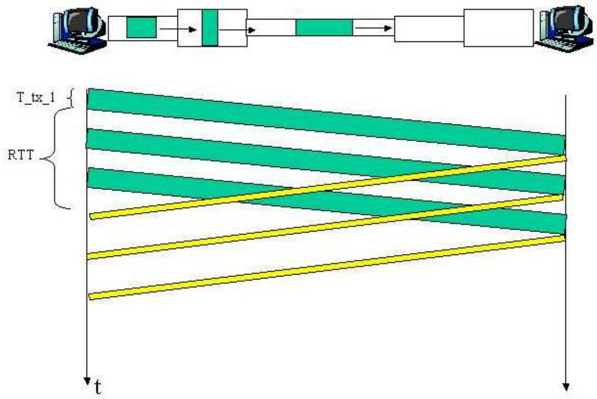
Ma quali sono le prestazioni del sistema?
RTT = network Round Trip Time
T_tx_i = Size(packet i) / channel bitrate
Channel Utilization = T_tx_i / (RTT + T_tx_i)

es. invio segmenti di 1000 Byte, rete a 1 Gbps, con RTT 30 ms
T_tx_i = 8000 bit / 2^30 bps = 8 microSec
Channel Utilization = 8 / (8+30000) = 266 Kbps (basso utilizzo!)

Prodotto (Bandwidth * Delay) della rete: (effetto memoria)
rappresenta la quantità di dati "in transito" sulla rete.

Idealmente si dovrebbe sfruttare del tutto per rendere massimo il throughput del sistema (effetto pipeline). Se la pipeline è piena al destinatario vengono recapitati 1 Gbps di dati.

Gestione del canale a Pipeline



89

Gestione del canale a Pipeline:

non si attende di ricevere l'ACK dei segmenti precedenti prima di inviare i segmenti successivi (se disponibili):

- aumentano i numeri di sequenza (Ack non ambigui)
- necessità di buffer su mittente e destinatario

Idealmente si dovrebbero trasmettere i segmenti al massimo ritmo di invio sostenibile dalla rete (prodotto Bandwidth*Delay). La rete funziona come una spugna al massimo dell'assorbimento dei dati.

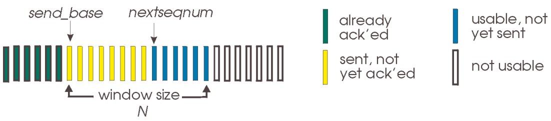
Q: Ma se ci sono errori o perdita di segmenti o Ack/NAK?
A: esistono due tecniche per la gestione dei problemi di trasmissione in canali gestiti con protocolli a Pipeline:
protocollo Go-Back-N (GBN) e protocollo Selective Repeat (SR).

90

Protocollo Go-Back-N

- k bit usati per numerare la sequenza di segmenti
- finestra massima (scorrevole) di N segmenti in sospeso (trasmessi ma non confermati)
- Ack cumulativi: Ack(n) vale sul mittente anche per tutti i segmenti sospesi in [send_base…n-1]
- destinatario invia Ack solo se il segmento è quello atteso (oppure ripete ultimo Ack valido) e non inserisce in buffer segmenti fuori ordine
- timer per la ritrasmissione (solo segmento send_base)

in caso di timeout: ritrasmissione di tutti i segmenti successivi della finestra

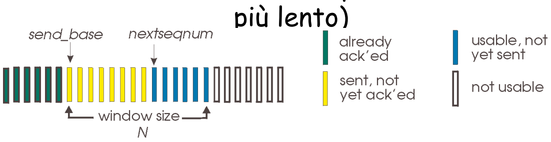


91

Aspetti critici di Go-Back-N:

- Perché scartare i dati fuori sequenza, anche se corretti?
  - buffer limitato, semplice gestione (Expected_seq#)
- Se prodotto (Bandwidth*Delay) è grande, allora N dovrebbe essere grande
  - N grande => alta probabilità di errore… ma in caso di errore ritrasmetto N segmenti? saturazione della rete!
- Perché N deve essere limitato? e a quale valore è opportuno limitare N?

controllo di flusso e controllo di congestione:
N = min(capacità buffer destinatario, capacità di smaltimento del router più lento)



92

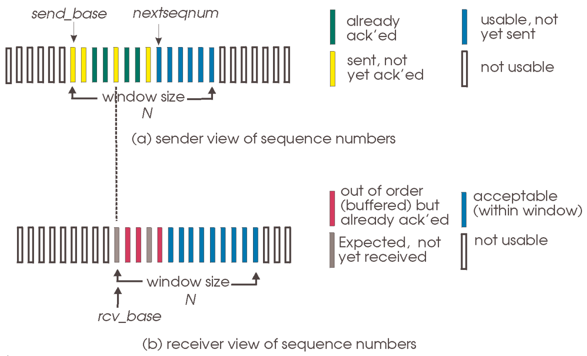
Protocollo Selective Repeat

Il destinatario che implementa Selective Repeat:

- gestisce una finestra di ricezione non superiore al suo buffer
- invia Ack specifici e bufferizza segmenti anche fuori ordine di ricezione, purchè entro la finestra di ricezione.
- invia Ack in caso di segmenti ripetuti anche precedenti nel range [Expected_seq#-N… Expected_seq#].

Il mittente che implementa Selective Repeat:

gestisce un timer (logico) per ogni segmento in sospeso



 93

Internet Protocol (IP)

- Packets may be delivered out-of-order
- Packets may be lost
- Packets may be duplicated

 94

Transmission Control Protocol (TCP)

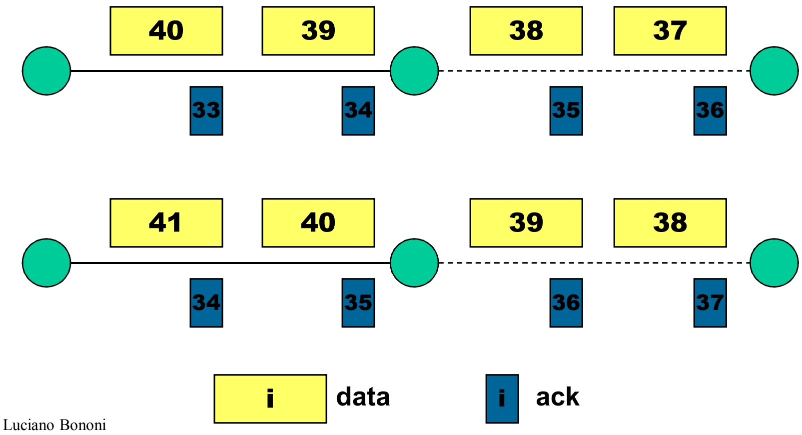
- Reliable ordered delivery
- Implements congestion avoidance and control
- Reliability achieved by means of retransmissions if necessary
- End-to-end semantics
  - Acknowledgements sent to TCP sender confirm delivery of data received by TCP receiver
  - Ack for data sent only **after** data has reached receiver

TCP Basics

- Cumulative acknowledgements
  - An acknowledgement ack's all contiguously received data
- TCP assigns byte sequence numbers
  - For simplicity, we will assign packet sequence numbers
- Also, we use slightly different syntax for acks than normal TCP syntax
  - In our notation, *ack i* acknowledges receipt of packets through packet *i*

Cumulative Acknowledgements

- A new cumulative acknowledgement is generated only on receipt of a new in-sequence packet

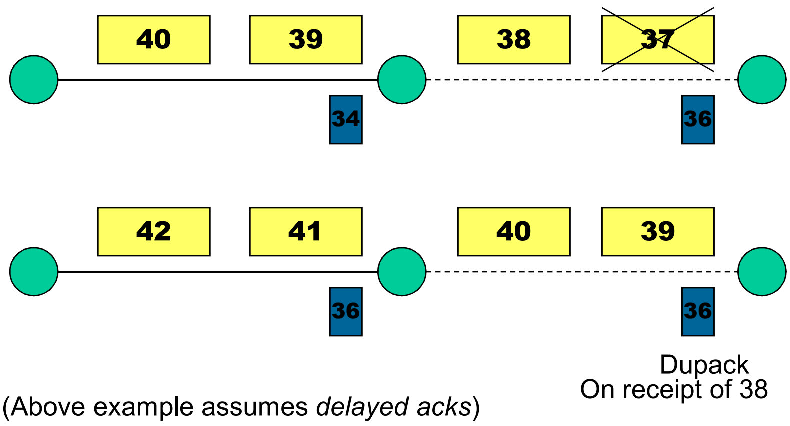


97

Duplicate Acknowledgements

- A dupack is generated whenever an

  out-of-order segment arrives at the receiver

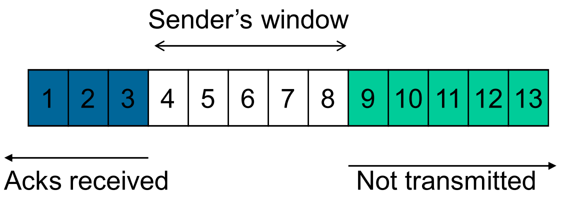


(Above example assumes *delayed acks*)

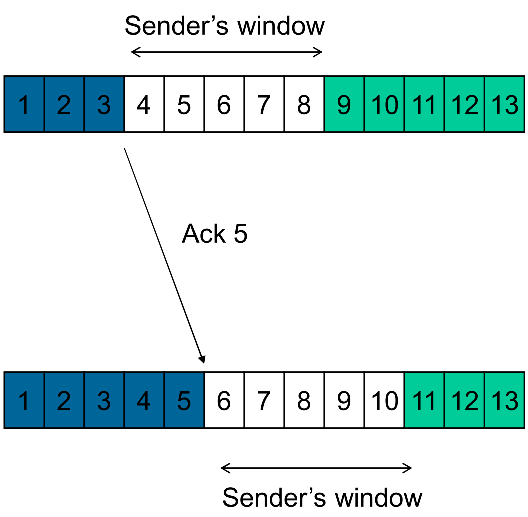
98

Window Based Flow Control

- Sliding window protocol
- Window size minimum of
  - receiver's advertised window - determined by available buffer space at the receiver
  - congestion window - determined by the sender, based on feedback from the network

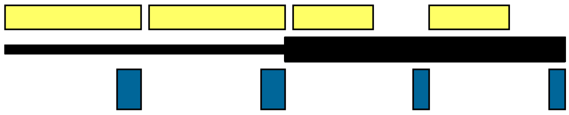


Window Based Flow Control



Ideal Window Size

- Ideal size = delay * bandwidth
  - delay-bandwidth product



- What if window size < delay*bw ?
  - Inefficiency (wasted bandwidth)
- What if > delay*bw ?
  - Queuing at intermediate routers
    - increased RTT due to queuing delays
  - Potentially, packet loss

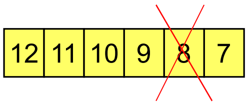
Detecting Packet Loss Using Retransmission Timeout (RTO)

- At any time, TCP sender sets retransmission timer for only one packet
- If acknowledgement for the timed packet is not received before timer goes off, the packet is assumed to be lost
- RTO dynamically calculated

Detecting Packet Loss Using Dupacks: Fast Retransmit Mechanism

- Dupacks may be generated due to
  - packet loss, or
  - out-of-order packet delivery

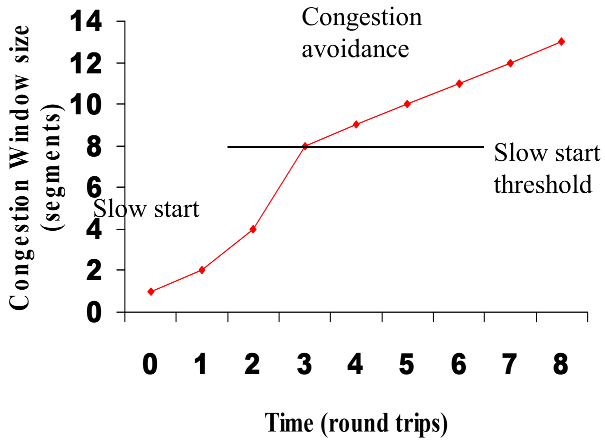
- TCP sender assumes that a packet loss has occurred if it receives three dupacks consecutively



Receipt of packets 9, 10 and 11 will each generate a *dupack* from the receiver. The sender, on getting these dupacks, will *retransmit* packet 8.

Congestion Avoidance and Control

- Slow Start: **cwnd** grows exponentially with time during slow start
- When **cwnd** reaches slow-start threshold, congestion avoidance is performed
- Congestion avoidance: **cwnd** increases linearly with time during congestion avoidance
  - Rate of increase could be lower if sender does not always have data to send



Example assumes that acks are not delayed

Congestion Control

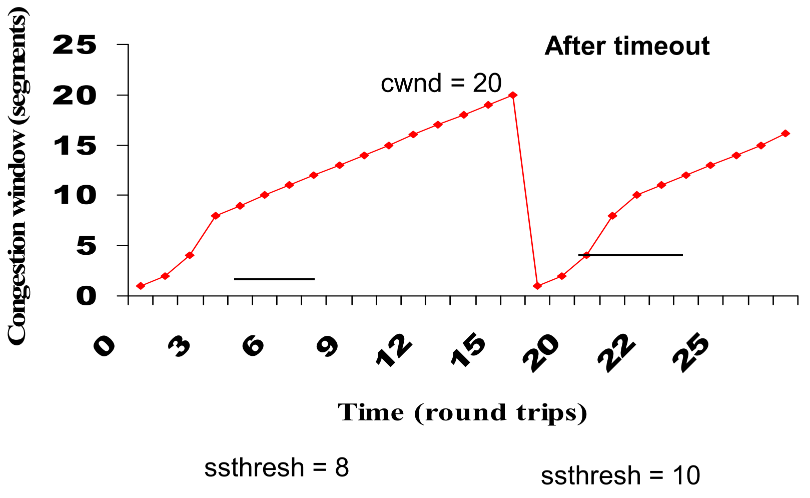
- On detecting a packet loss, TCP sender assumes that network congestion has occurred
- On detecting packet loss, TCP sender drastically reduces the congestion window
- Reducing congestion window reduces amount of data that can be sent per RTT

Congestion Control -- Timeout

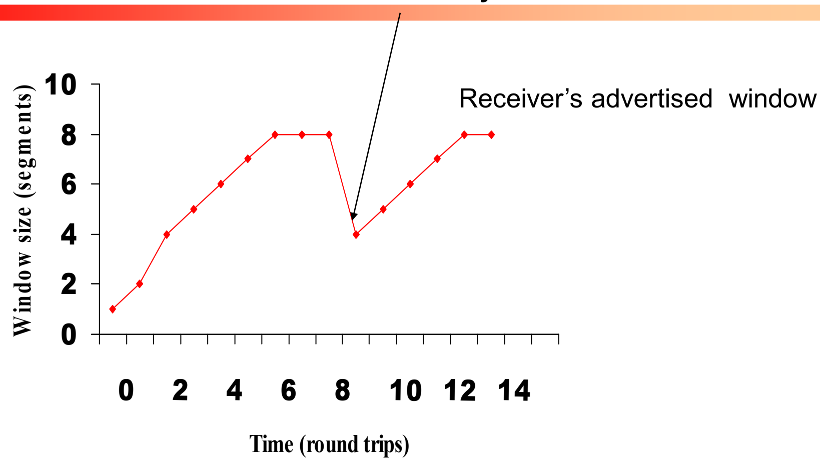
- On a timeout, the congestion window is reduced to the initial value of 1 MSS
- The slow start threshold is set to half the window size before packet loss
  - more precisely,

    ssthresh = maximum of min(cwnd,receiver's advertised window)/2 and 2 MSS
- Slow start is initiated

Timeout effect on CWND



After fast recovery



After fast retransmit and fast recovery window size is reduced in half.

Possibili argomenti per seminario finale (generali)

multiple access techniques (including the impact of multiple antennas)
cellular system design
ad-hoc wireless networking
multiuser information theory
capacity of ad hoc networks
access techniques in wireless networks
dynamic resource allocation in wireless networks
cross layer design in wireless networks
adaptive modulation/coding in multiuser systems
power control in wireless networks
space-time processing for mobile communications
MIMO techniques for multiuser systems
multiuser multicarrier /OFDM systems
CDMA systems
interference cancellation / multiuser detection in CDMA
coding/spreading tradeoffs in CDMA
CDMA vs. OFDM
user location strategies in WiNet
multirate/multimedia over wireless networks

Possibili argomenti per seminario finale

smart antennas
traffic models for multimedia data
energy efficient protocols for ad hoc and sensor systems
routing for ad hoc wireless networks
routing for vehicular wireless networks
performance of TCP/IP and/or ATM over wireless channels
performance of TCP/IP over multihop wireless networks
software radios
multiuser ultra wide band (UWB) systems
HW constraints in wireless systems
wireless system services and killer applications
innovative and visionary wireless-enabled services
RFID technologies
wireless frameworks' implementations
service frameworks for wireless devices synchronization
Mobile IP
Security issues in wireless systems
Vehicular Network Technologies
Wireless Monitoring
IEEE 802.[11-22] and related special task groups