

# Mobile and Sensor Systems

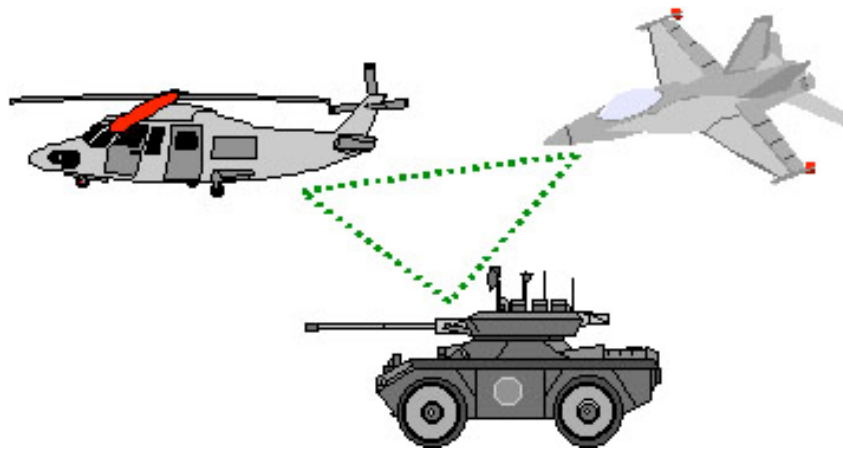
Lecture 2: Infrastructure and  
Opportunistic Mobile Networks

Prof Cecilia Mascolo

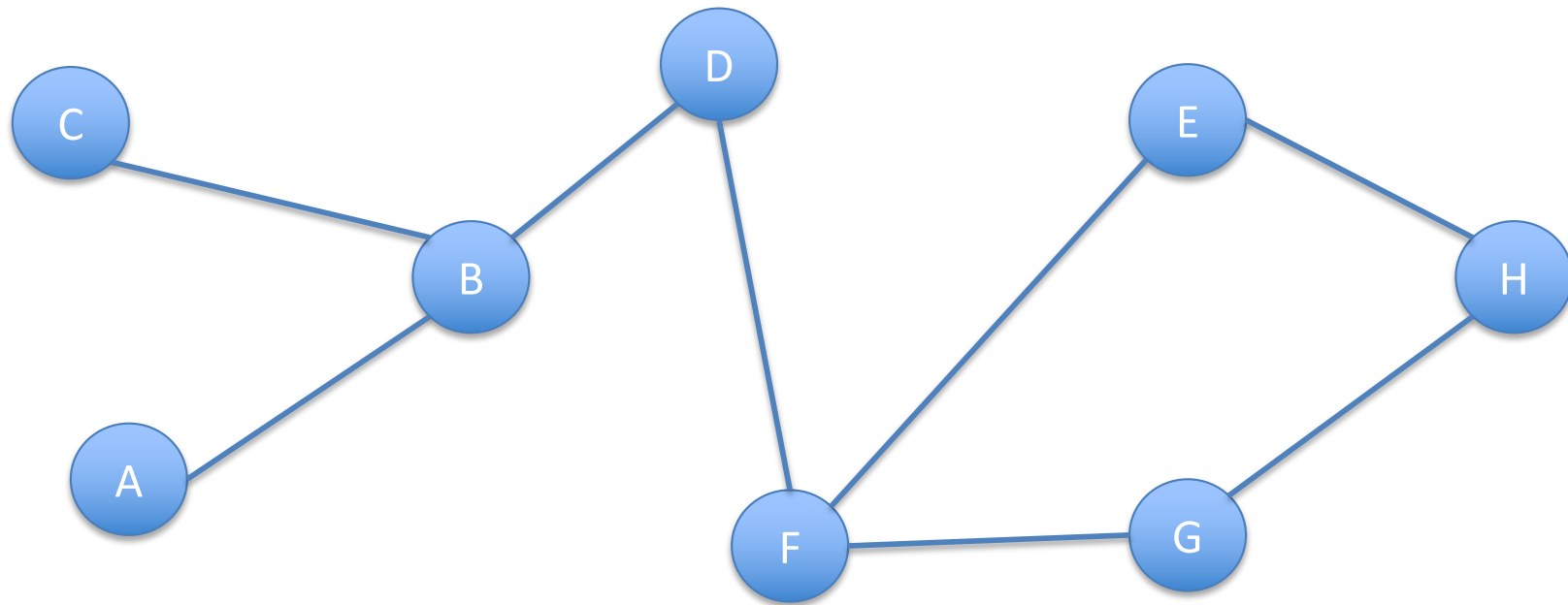
# In this lecture

- We will introduce ad hoc routing protocols.
- We will also introduce disconnected ad hoc networks and opportunistic routing protocols.

# Examples of Multi-hop Ad hoc Networks



# A schematic of an ad hoc network



Note that this is a static snapshot:  
the network will be reconfiguring  
when nodes move.

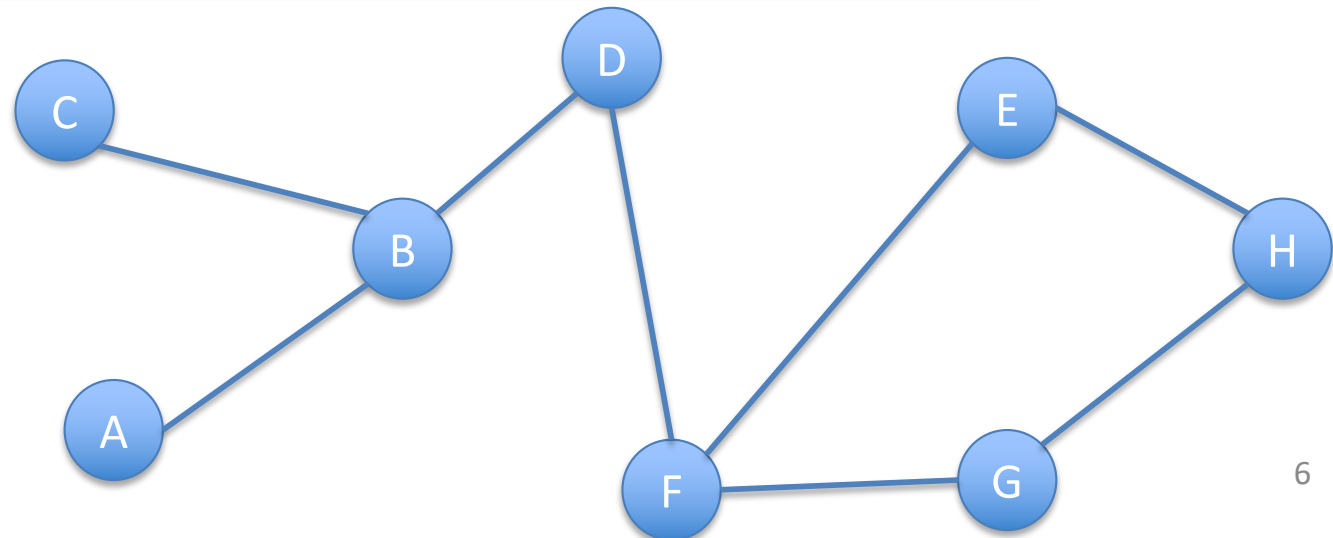
# Examples of Ad-hoc Networking

- Destination Sequenced Distance Vector (**DSDV**) Routing.
- Proactive: routes are maintained also when not needed. Each node maintains a table with a route to every node.
- Each entry of the table has a sequence number assigned by the destination.

Dest. Next Hop, Hops Required, Dest. Seq. number.

# DSDV: Routing Table for Node D

<b>Dest</b>	<b>Nexthop</b>	<b>Hops</b>	<b>SequenceN</b>
<b>A</b>	<b>B</b>	<b>2</b>	<b>406</b>
<b>B</b>	<b>B</b>	<b>1</b>	<b>128</b>
<b>C</b>	<b>B</b>	<b>2</b>	<b>564</b>
<b>D</b>	<b>D</b>	<b>0</b>	<b>710</b>
<b>E</b>	<b>F</b>	<b>2</b>	<b>392</b>
<b>F</b>	<b>F</b>	<b>1</b>	<b>076</b>
<b>G</b>	<b>F</b>	<b>2</b>	<b>128</b>
<b>H</b>	<b>F</b>	<b>3</b>	<b>050</b>

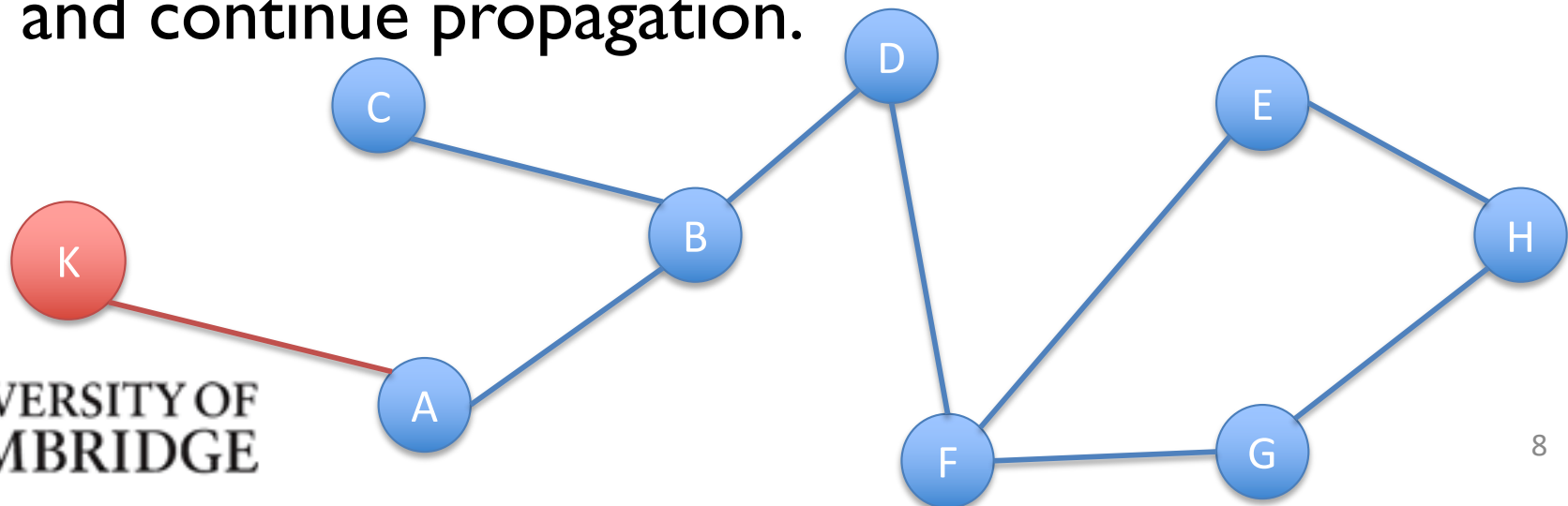


# DSDV Routing Updates

- Each node periodically transmits updates.
- Includes its own sequences number, routing table updates.
- Nodes also send routing table updates for some incremental link changes.
- When two routes to a destination received from two different neighbors, choose the one with *greatest* destination sequence number.
- Sequence numbers help with loops

# DSDV: When a new link appears (ie a node moved closer)

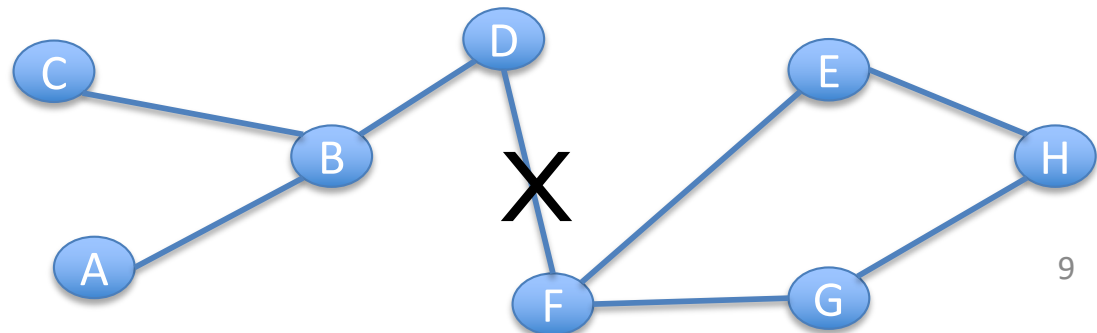
- When K joins it
  - Transmits routing table  $\langle K, K, 0, |0| \rangle$
  - Node A receives it and inserts in routing table:  $\langle K, K, 1, |0| \rangle$
  - Node A propagates the new route to neighbours.
  - Neighbours of A update table with  $\langle K, A, 2, |0| \rangle$  and continue propagation.





# DSDV: When a link breaks (ie a node moves away)

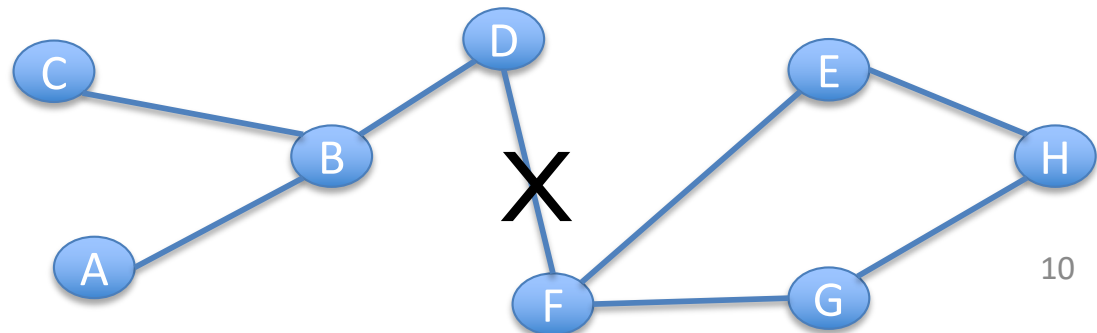
- The link between D and F breaks (the two nodes move apart or the wireless link has no connection for other reasons).
- Node D notices the break (eg lack of table update from F):
  - Updates hop count to F to infinity.
  - Increments the sequence number for F,E,G,H
  - Then:



# DSDV: When a link breaks (ie a node moves away)

– Then Node D sends updates with new route information:

- $\langle F, -, \infty, 077 \rangle$
- $\langle E, -, \infty, 393 \rangle$
- $\langle G, -, \infty, 129 \rangle$
- $\langle H, -, \infty, 051 \rangle$



# DSDV: Limitations

- Circulating and maintaining table updates is expensive for the network.
  - Especially if this is battery powered.
- If the network changes a lot then these updates might be worthwhile but only if all nodes need to communicate to all others
  - Why would a node need to keep updates of routes to nodes it does not need to communicate with?

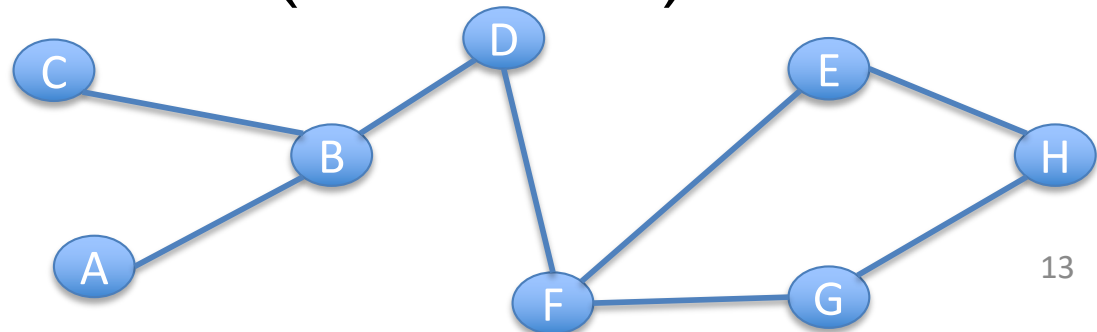
# Another Ad-hoc routing example

## Dynamic Source Routing (DSR)

- DSR is a reactive protocol: routes are searched only when communication with a node is needed.
- When a node needs to communicate it sends a route request packet.
- Nodes receive it and add themselves to the path and propagate the request to their neighbours.
- Eventually the destination node is found and the path is sent back to the source (how?).

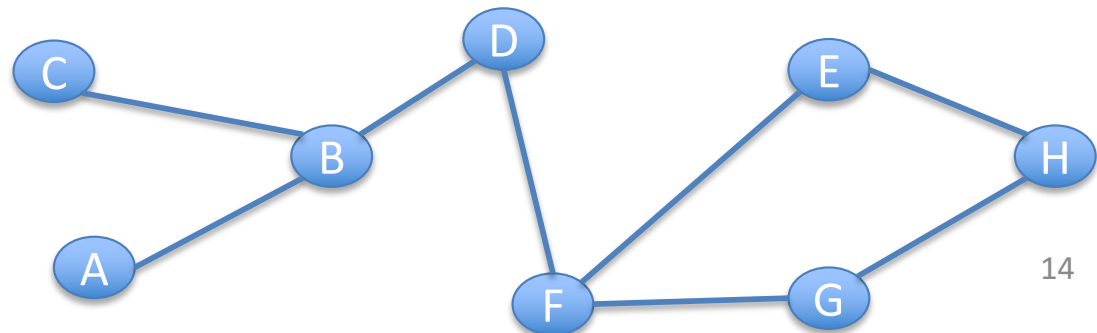
# DSR example

- Node A needs to communicate with F. Sends request  $\langle A, F, [-], |0| \rangle$ .
- Node B receives the request and forwards:
  - $\langle A, F, [B], |0| \rangle$ .
- Node D receives it and forwards:
  - $\langle A, F, [B, D], |0| \rangle$ .
- Node F receives it and ? (continued).



# DSR example (continued)

- Node F receives it and:
  - If links are symmetric sends the path back following the indicated inverse route.
  - Or, if it has cached a path to A uses it.
  - Otherwise it sends a route request for A.



# DSR Comments

- Sequence numbers are used to avoid routing loops.
- Routes are cached for some time to avoid frequent route requests by intermediate nodes.
- Low mobility and stable origin-destinations are the best scenarios for this protocol.

# Hybrid Solutions

## Zone Routing Protocol (ZRP)

- Zone routing is a hybrid protocol which combines proactive with reactive approaches.
- A zone around node N is maintained where routes are collected proactively.
- Beyond the zone an inter zone protocol is responsible to determine the routes in a reactive way.

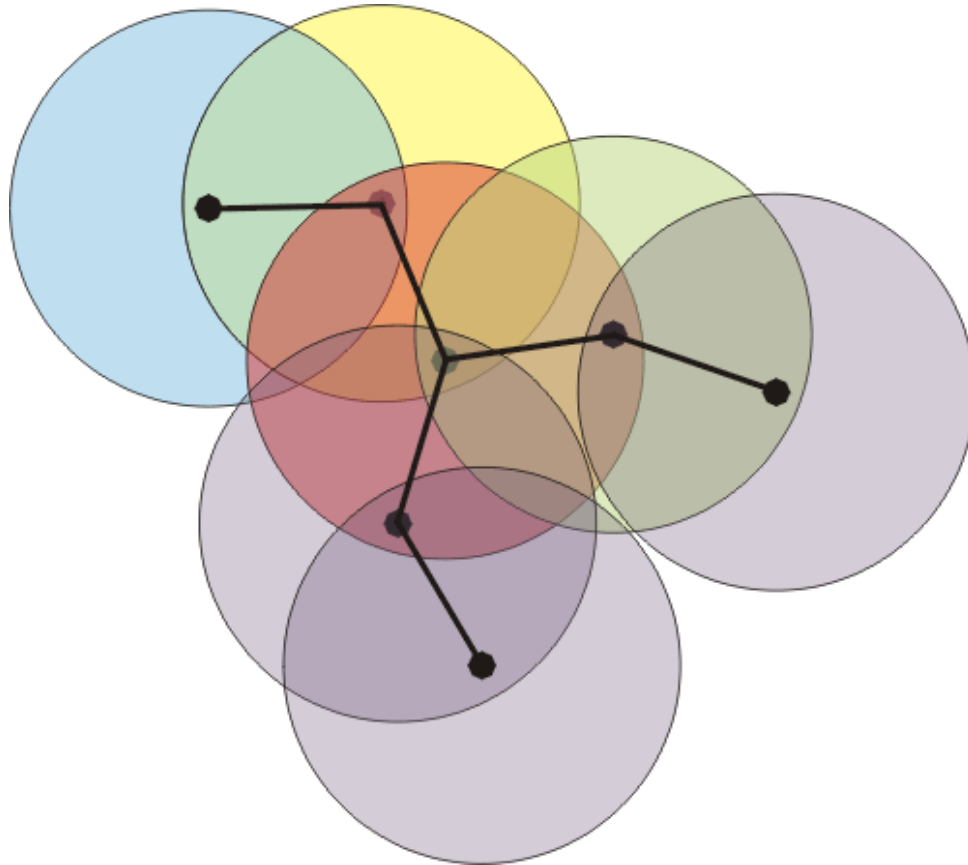


# Disconnected Ad-hoc Networks

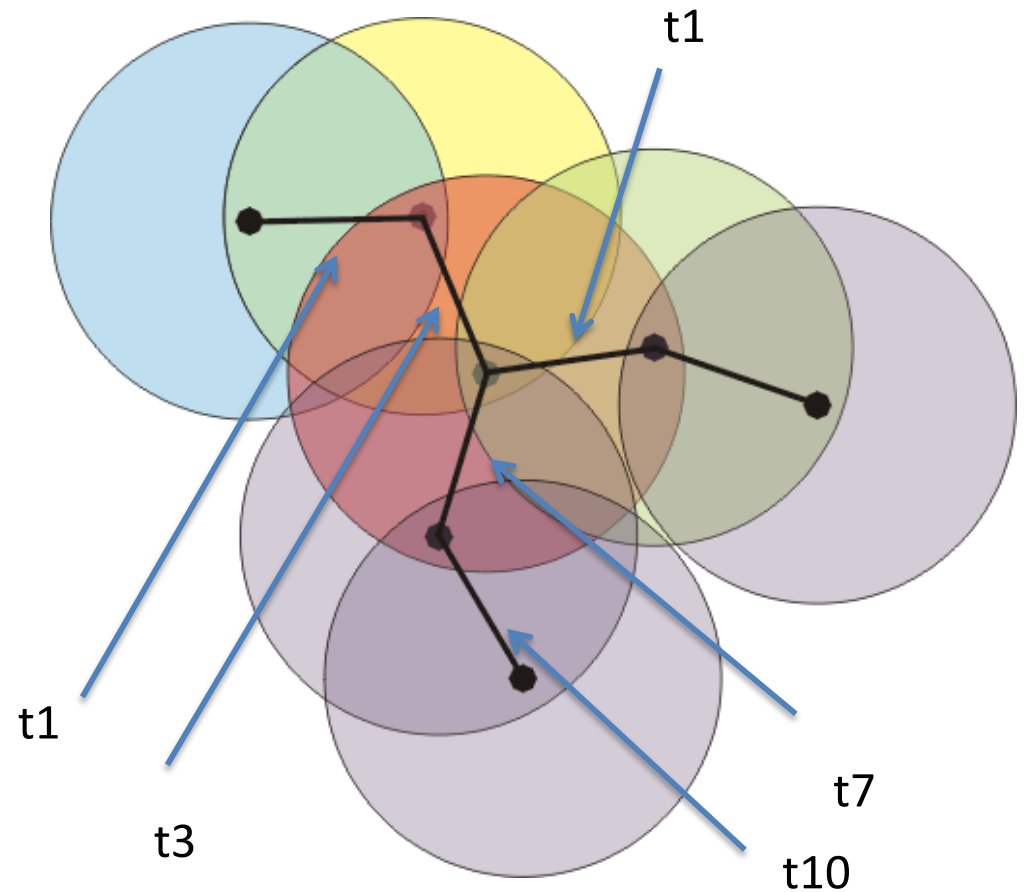
- The protocols we have seen until now only work when there is a “connected path” among the communicating nodes
- I.e. no storage on intermediate nodes is allowed.
- The protocols do not work when this is not the case.

# Connected vs Disconnected Ad-hoc Networks

Connected: there is a connected path  
among each couple



Disconnected: not all links are present  
at the same time



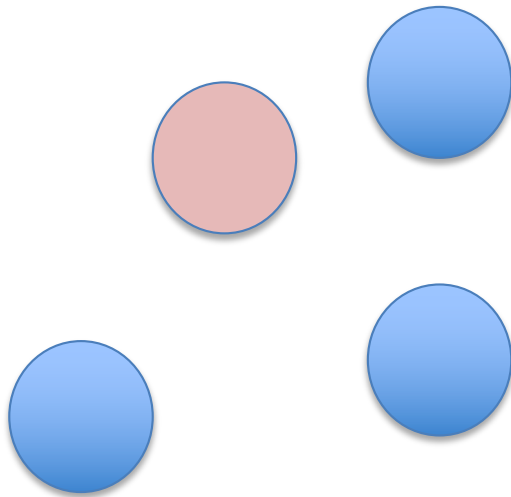
# Delay Tolerant Networks and Protocols

- These protocols do not assume a temporally connected path among the nodes.
- Nodes can accept a packet and deliver it later, after some time storing it.

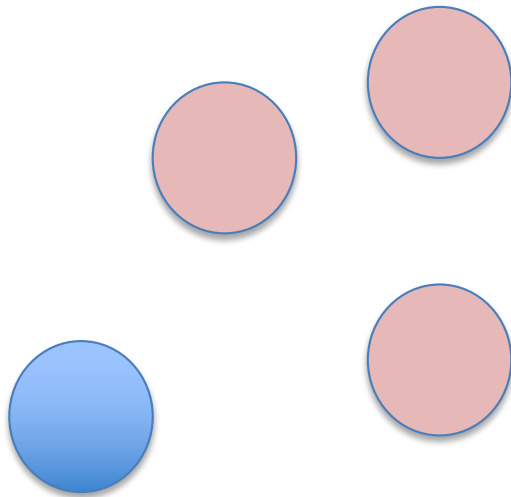
# Example: Epidemic Routing

- A flooding protocol which allows nodes to store packets before forwarding.
- A node accepts a packet, moves while carrying the packet and then forwards at a different time to a different neighbourhood.

# Epidemic Routing Example



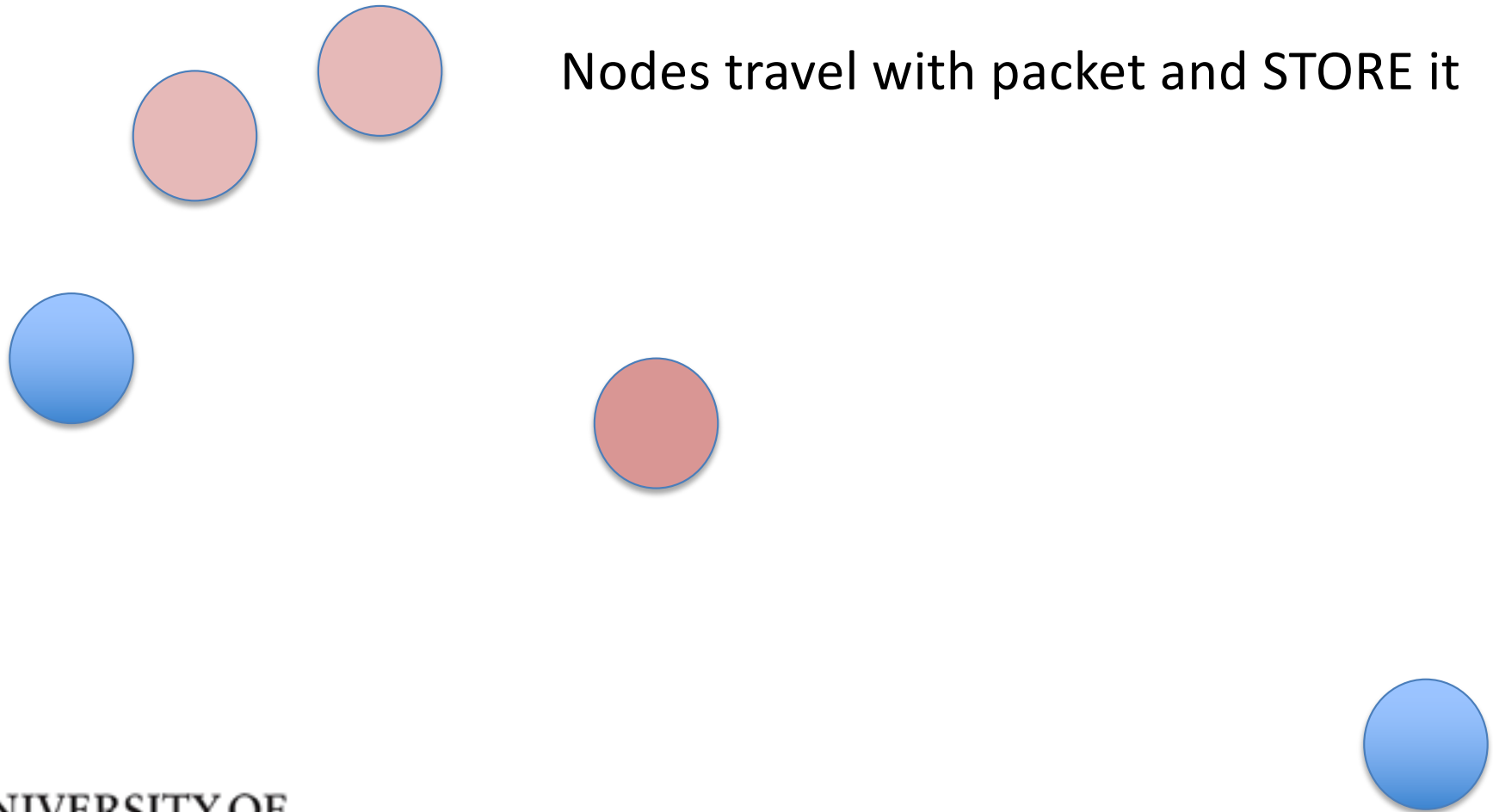
# Epidemic Routing Example



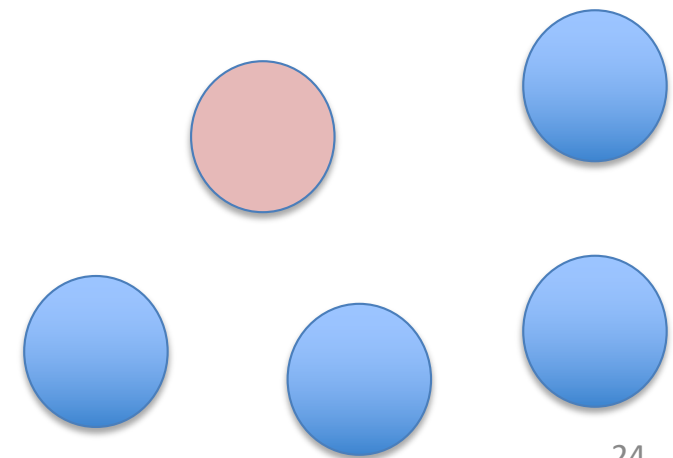
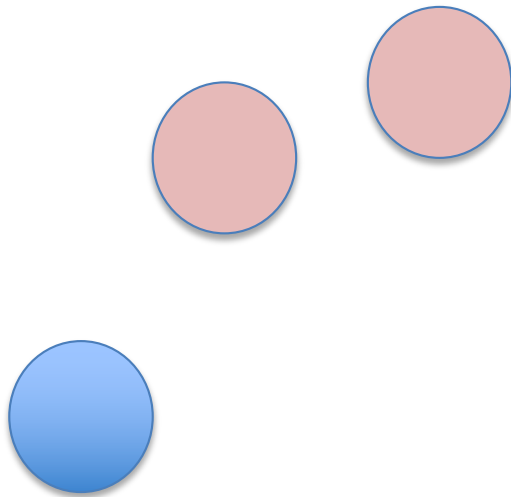
Packet passed to a fraction of neighbouring nodes, which store it



# Epidemic Routing Example

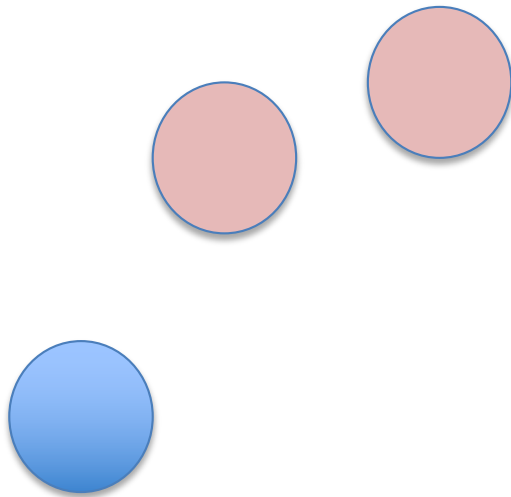


# Epidemic Routing Example

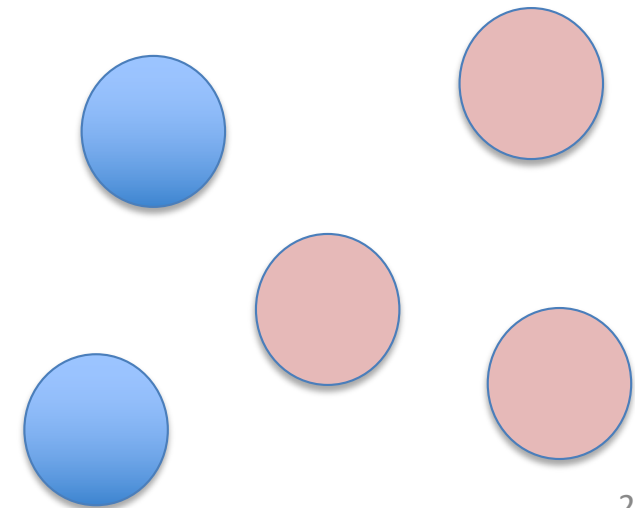




# Epidemic Routing Example



Nodes eventually forward it when in reach of other nodes



# Pros and Cons of Epidemic Routing

- If tuned right it reaches optimal delivery
- Needs large memory on nodes or buffers fill up and packets get ejected and lost
- Forwarding number (epidemic infectivity) needs to be tuned to avoid too many network transmissions (and collision which affect power)

# Can we do better?

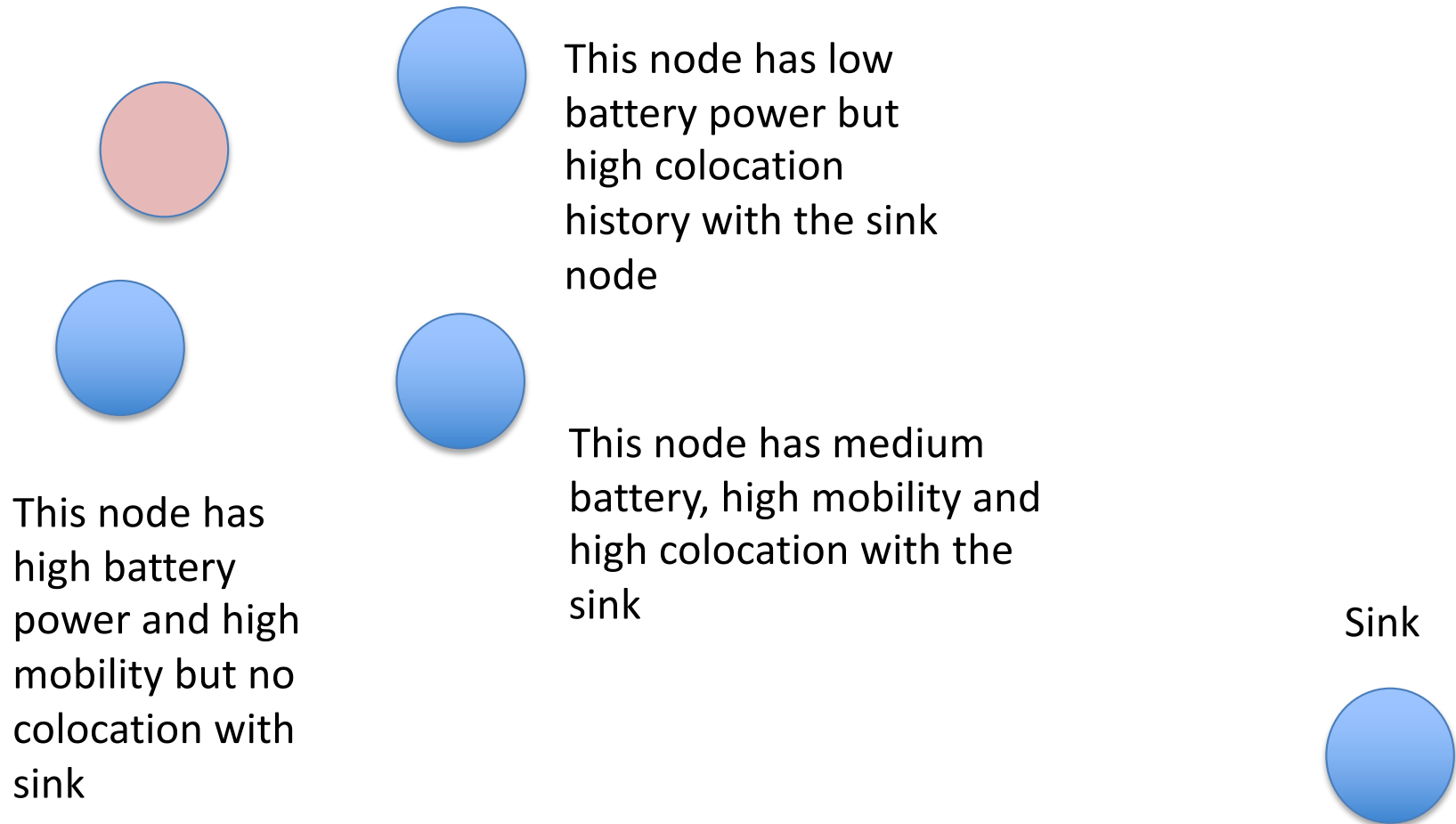
- Exploit the knowledge on the mobility of the nodes
- Is the mobility deterministic (ie. Always on the same path at same times like busses)? Maybe we can even control the mobility of some of the nodes!
- If not fixed, is it at least predictable?
- If not predictable, random...

# When mobility is predictable

- Instead of blindly forwarding packets to all or some neighbours, intermediate nodes estimate the chance, for each outgoing link, of eventually reaching the destination.
- Based on this estimation, the intermediate nodes decide whether to store the packet and wait for a better chance, or decide to which nodes (and the time) to forward.

# Context Aware Routing Example

## which node(s) to choose as carrier?



# Context Aware Routing (CAR)

- A node chooses the best carrier to reach a specific node. How is the best carrier neighbour chosen?
  - Host mobility, host collocation with destination node, battery.
  - A utility function which weights these aspects
  - Kalman Filter is used to predict future host collocation with destination based on previous history.
- The approach is based on local knowledge only.

# Applications...

- Remote areas
  - Developing regions
  - Military
  - Sensor systems
  - Hybrid models (store and forward can give good performance regardless of presence of infrastructure so why not use)
- But also...mesh networks where access points (APs) not connected to the wire select the best relay AP to forward packets

# Summary

- We have talked about ad hoc and opportunistic routing protocols.
  - For connected and disconnected networks
- We have discussed applicability and applications



# Reference

- Agrawal, D. & Zheng, Q. (2006). Introduction to wireless and mobile systems. Thomson.
- A. Vahdat and D. Becker, Epidemic routing for partially-connected ad hoc networks, 2000.
- M. Musolesi and C. Mascolo. CAR: Context-aware Adaptive Routing for Delay Tolerant Mobile Networks. In IEEE Transactions on Mobile Computing. Vol. 8(2). pp. 246-260. February 2009.