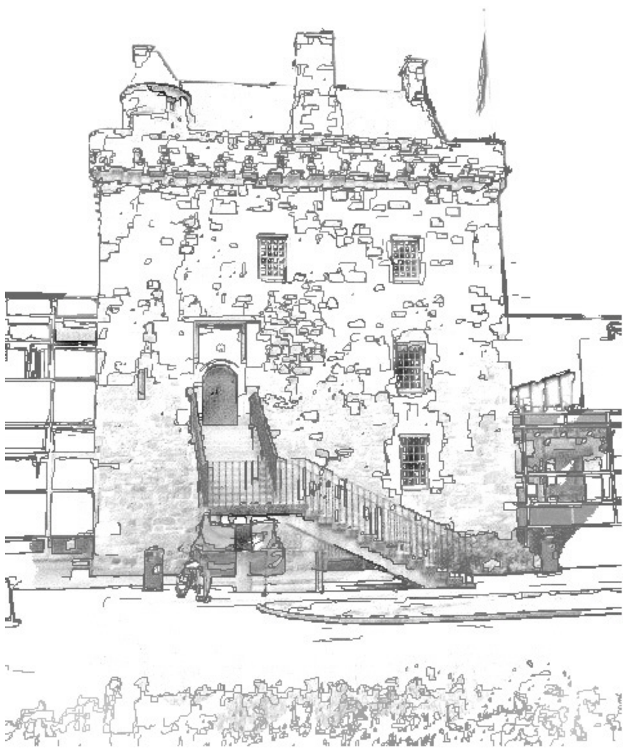
BCS SGAI Symposium/Colloquium on Mobile and Complex Agent Systems

Nikos Migas

Bill Buchanan



BCS SGAI Symposium/Colloquium on Mobile and Complex Agent Systems

The main objective of the symposium is to provide up-to-date research and applications of mobile and complex agent systems from leading researchers in the UK, and in Europe.

Dr W. Browne,
Dept of Cybernetics,
University of Reading.

J.Tain,
Glasgow Caledonian

Professor Wiebe van der Hoek,
University of Liverpool

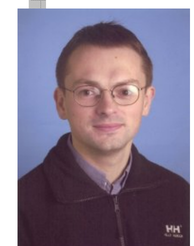
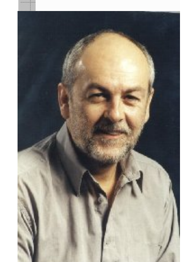
Christain
Erfurth

Prof W.Rossak,
Friedrich-Schiller-Universität,
Jena

S.Robles,
Universitat
Autonoma
de Barcelona.

Programme

10:00-10:30	Agent-based Systems for Ad-hoc Routing, W.Buchanan, N.Migas and K.McArtney, Napier University.
10:30-11:00	Social Laws in Alternating Time, Wiebe van der Hoek, University of Liverpool.
11:00-11:30	Cybernetic Intelligence: How Feedback Can Enhance the Behaviour of Mobile Robots, W. Browne and V.Becerra, University of Reading.
11:30-12:00	Mobile Personality Rich Agents, S. Cringean, D. Benyon and G. LePlatre, Napier University.
11:30-12:00	An Evolutionary Intelligent Agent Approach to the Prevention of Covert Channels in the TCP/IP Protocol Suite, D.Llamas, Napier University.
12:30-13:30	Lunch


NAPIER UNIVERSITY
EDINBURGH SCOTLAND

BCS

Mobile Agents

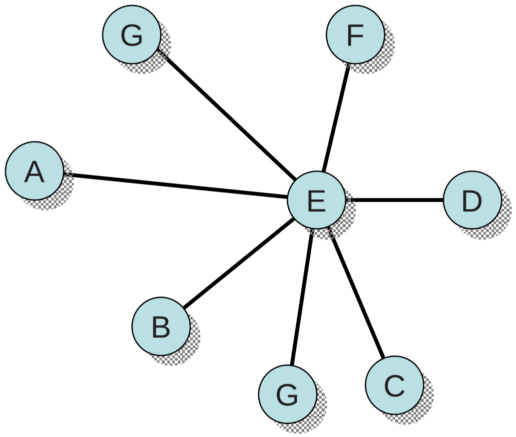
Mobile and Complex Agent Systems

13:30-14:00	On Multi-Agent Based Simulation of Economic Systems, H.Tianfield and J.Tian, Glasgow Caledonian University.
14:00-14:30	Mobile Devices for Ad-hoc Routing using Mobile Agents, N Migas, W.Buchanan and K.McArtney, Napier University.
14:30-15:00	Dynamic Routing of Autonomous Mobile Agents, W.Rossak and C.Erfurth, Friedrich Schiller University Jena.
15:00-15:30	Viral-Net System Via Autonomous Agent Technology , A.Abimbola, J.Munoz and W.Buchanan, Napier University.
15:30-16:00	Security Solutions for Killer Mobile Agent Applications, S.Robles, Universitat Autonoma de Barcelona.

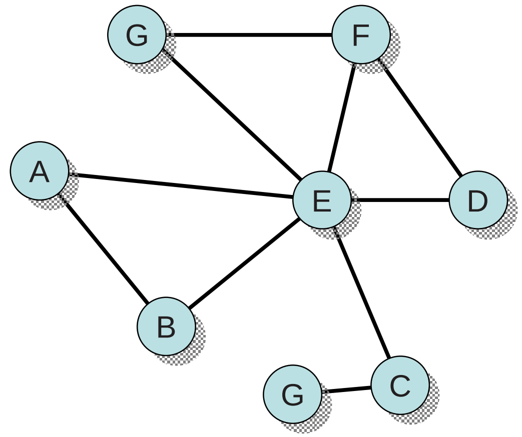
Agent-based Systems for Ad-hoc Routing

W.Buchanan
N.Migas
K.McArtney

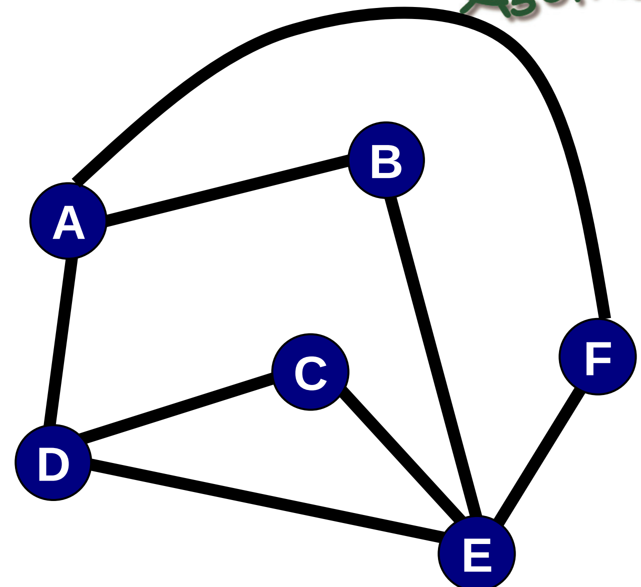
TCP, IP and RIP

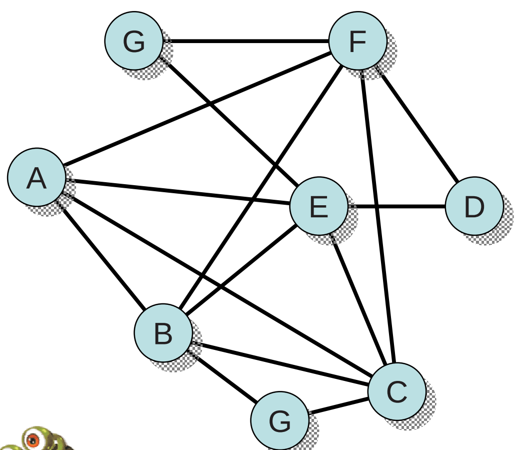


Centralised

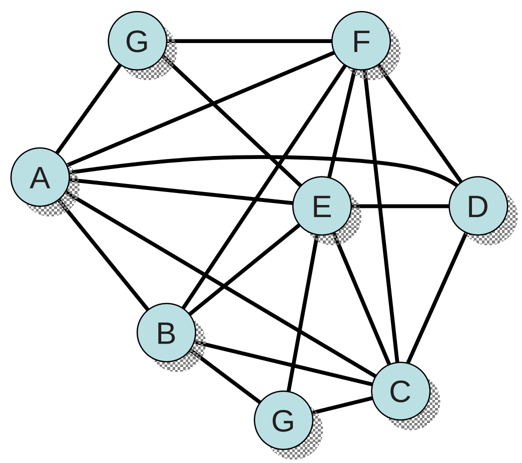


Decentralised



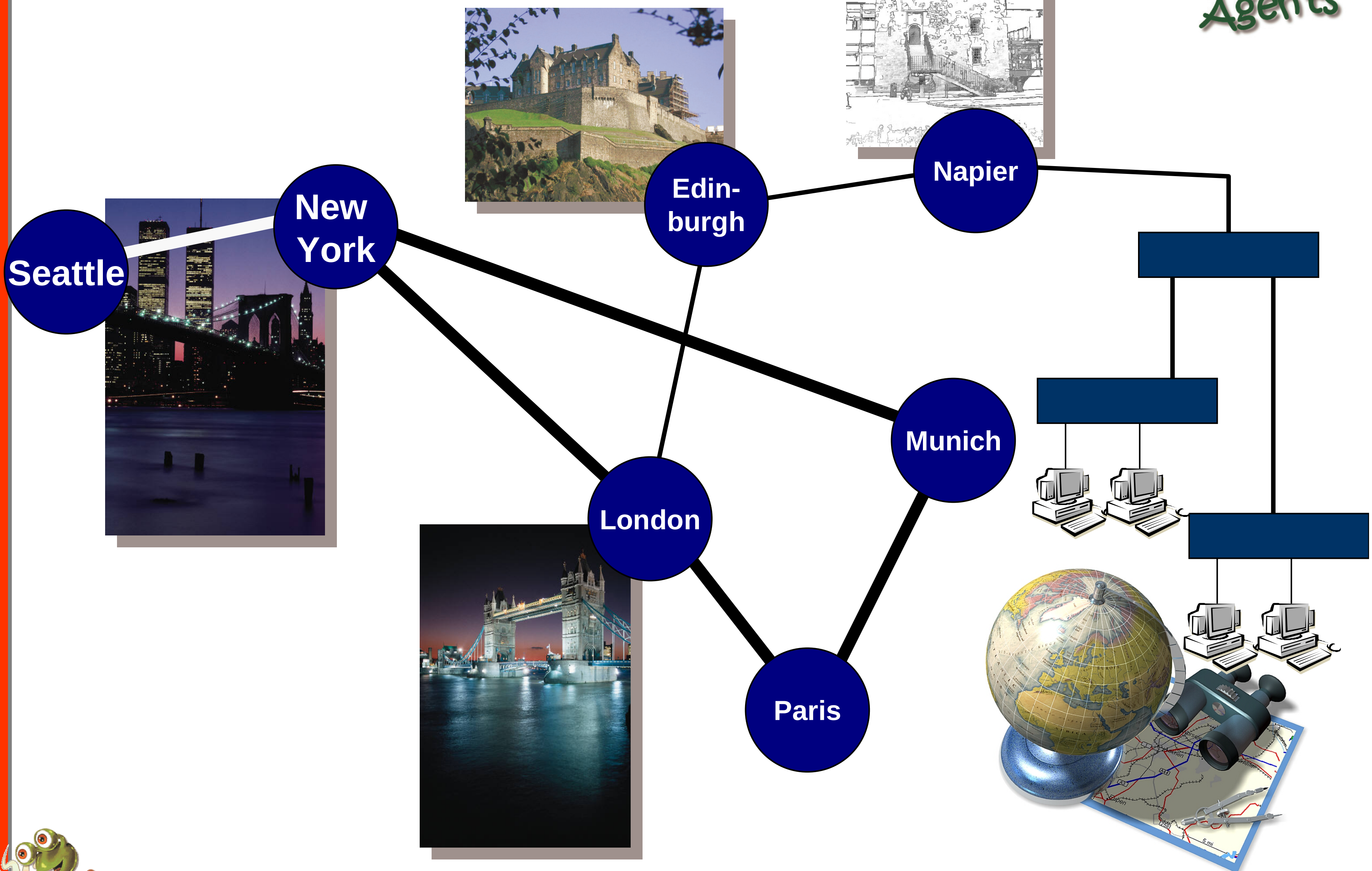


Distributed

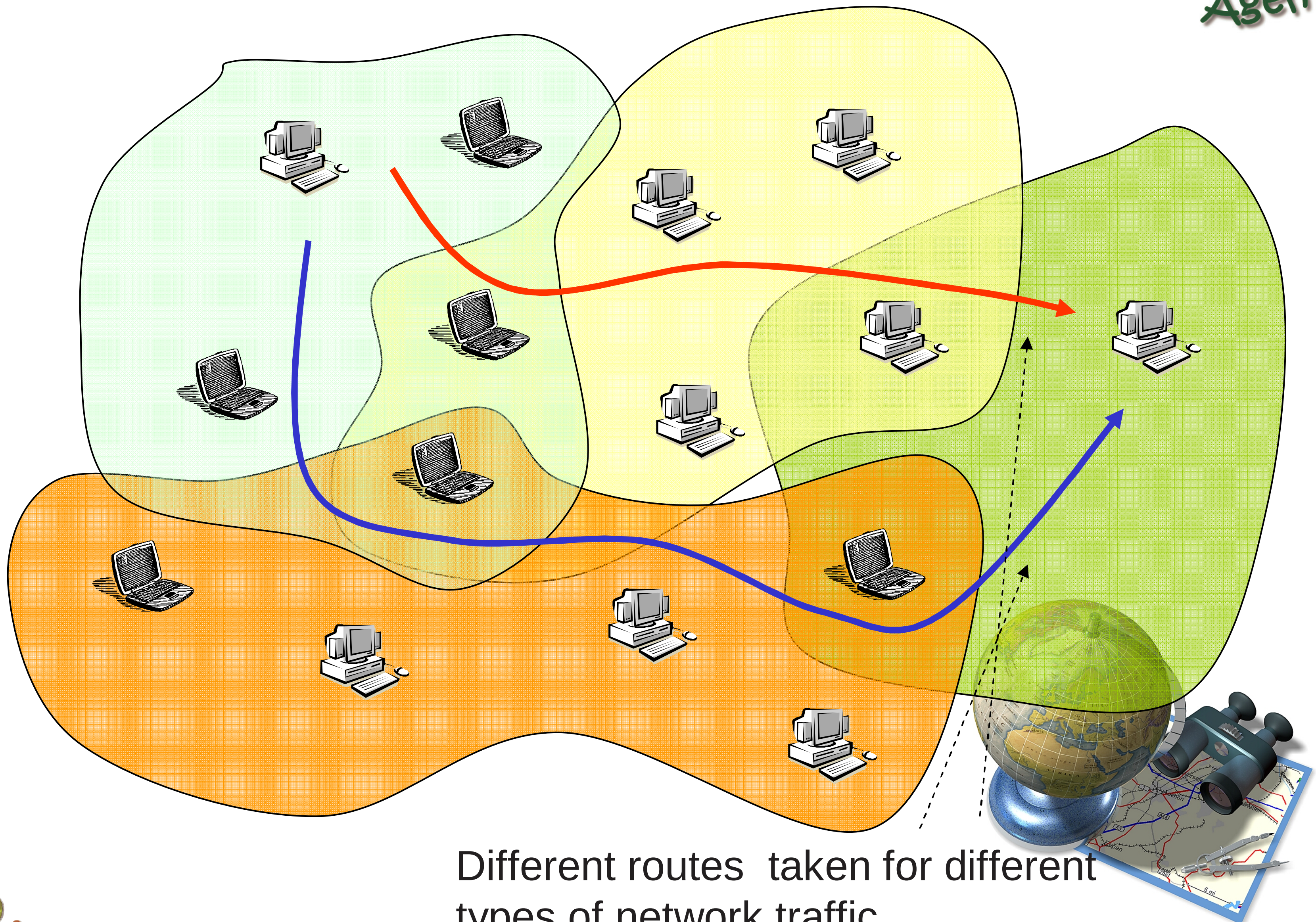


More fully distributed

The Internet at its conception



The Internet we have...

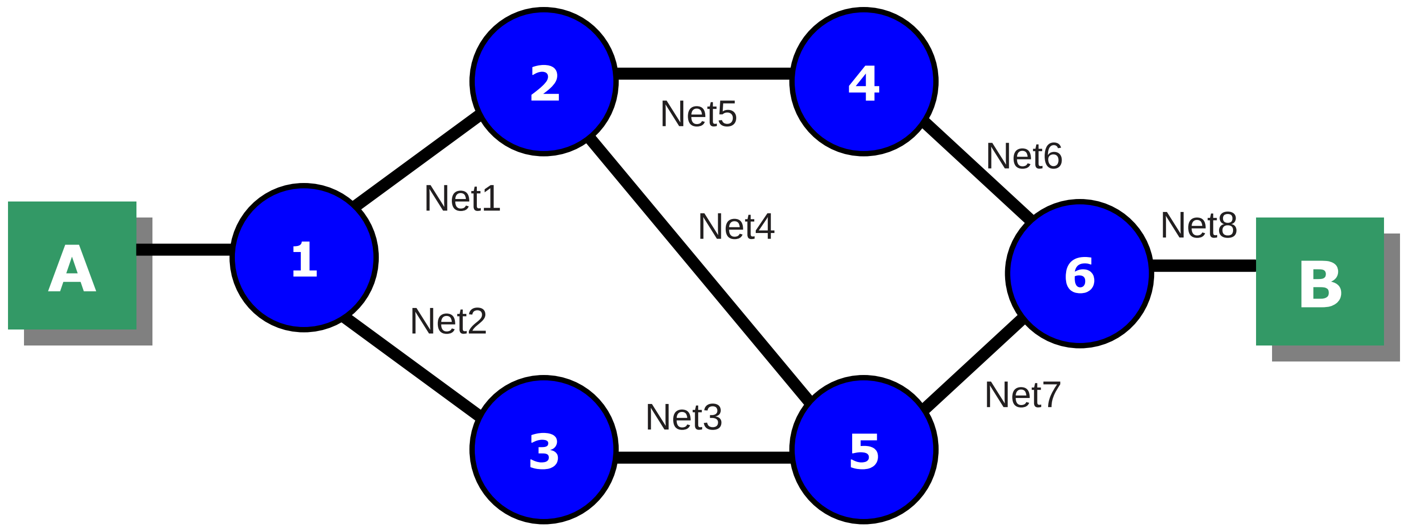


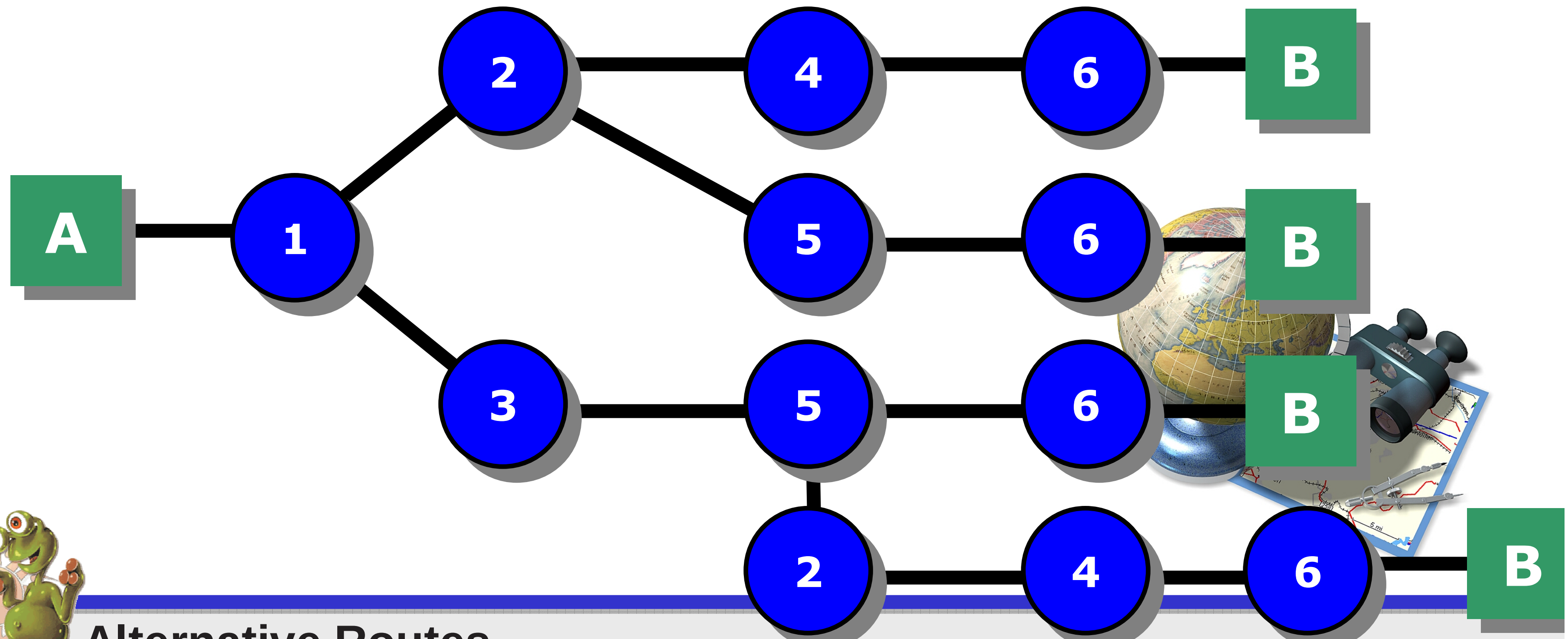
Different routes taken for different types of network traffic

Distributed Routing

Routing Protocols

Background





Alternative Routes

Best route?

Routing based on **hops**:

Route (1,3,5,6) = 4 hops [**BEST**]
Route (1,3,5,2,4,6) = 6 hops

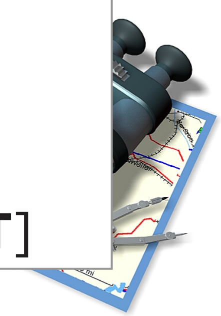
Routing based on **delay** (latency):

Route(2,4,6) = 1.5+1.25 = 2.75
Route(2,5,6) = 1.1+1.3 = 2.4 [**BEST**]

Routing based on **error probability**:

$P_e(2 - 5)=0.01$
$P_e(2 - 4)=0.05$

$P_e(5 - 6)=0.15$
$P_e(4 - 6)=0.1$



$P_{noerror}(2,5,6) =(1 - 0.01) \times (1 - 0.15) = 0.8415$
$P_{noerror}(2,4,6) =(1 - 0.05) \times (1 - 0.1) = 0.855$ [**BEST**]

Bandwidth. The data capacity of a link, which is typically defined in bps.

Delay. The amount of time that is required to send a packet from the source to a destination.

Load. A measure of the amount of activity on a route.

Reliability. Relates to the error rate of the link.

Hop count. Defined by the number of routers that it takes between the current router and the destination.

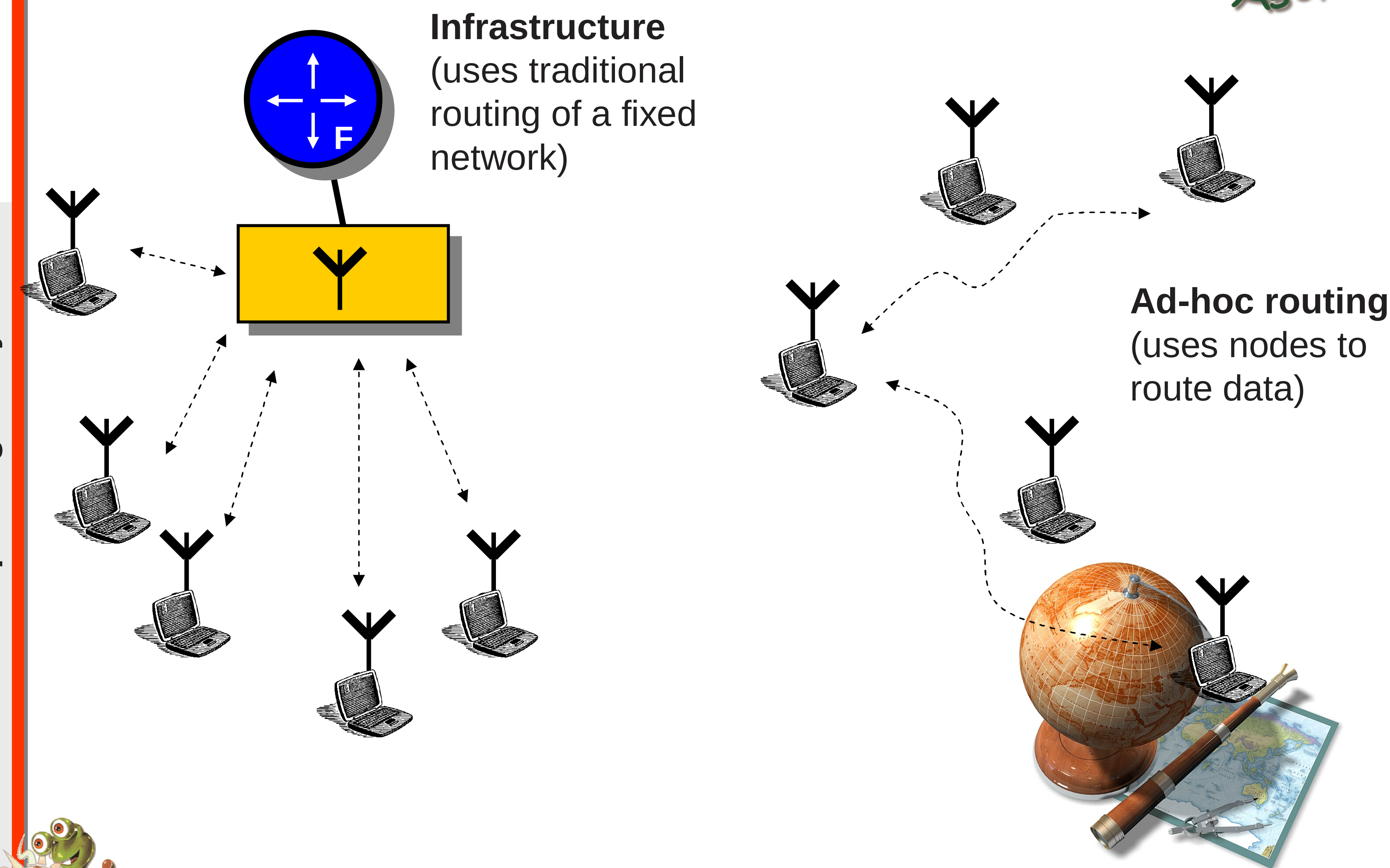
Ticks. Defines the delay of a link by a number of ticks of a clock.

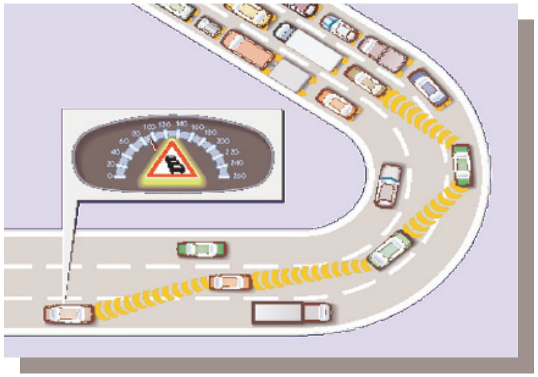
Cost. An arbitrary value which defines the cost of a link, such as financial expense, bandwidth, and so on.

Best Route Parameters?

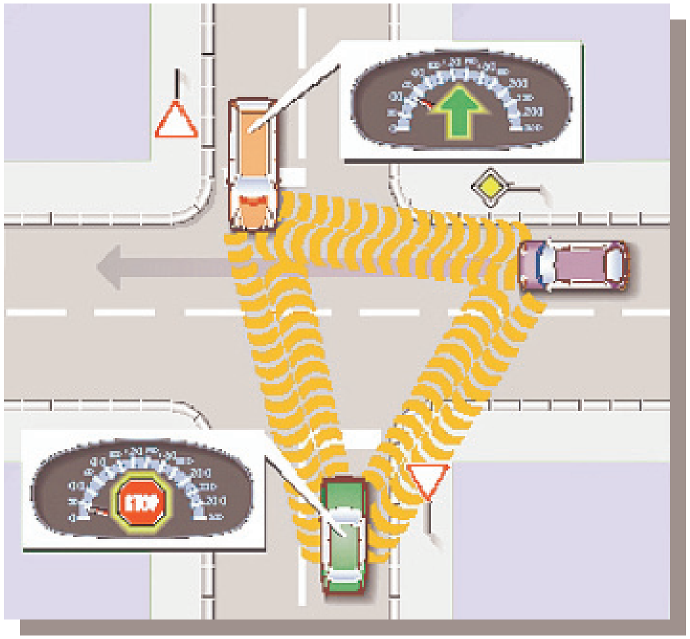
Ah-hoc Networks

W.Buchanan
N.Migas
K.McArtney


F
Infrastructure
(uses traditional
routing of a fixed
network)
Ad-hoc routing
(uses nodes to
route data)



- **Military and rescue operations**
  - Battlefield
  - Evacuation of a building on fire
- **Terrorism & Rescue Operations**
- **Hospitals**
  - Retrieve patient's information from hospital's database while in surgery
- **Conference meetings**
  - Share information quickly
  - Schedule meetings
- **Networking while on the road**
  - Inter-vehicle communication



Applications of Ad-hoc networks

Classifications of Ad-hoc Routing

- **Table-driven**. This is where the route is defined by continually updating routing tables. Examples include:
  - DSDV (Destination-Sequenced Distance-Vector).
  - CGSR (Clusterhead Gateway Switch Routing).
  - WRP (Wireless Routing Protocol).
- **Demand-driven (source-initiated)**. This is where the source initiates a route for a given connection. Examples include:
  - AODV (Ah-hoc On-Demand Distance Vector).
  - DSR (Dynamic Source Routing).
  - LMR (Lightweight Mobile Routing).
  - TORA (Temporally Ordered Routing Algorithm).
  - ABR (Associativity-based Routing).
  - SSR (Signal Stability Routing).

Table-based Ad-hoc Routing

DSDV
CGSR
WRP

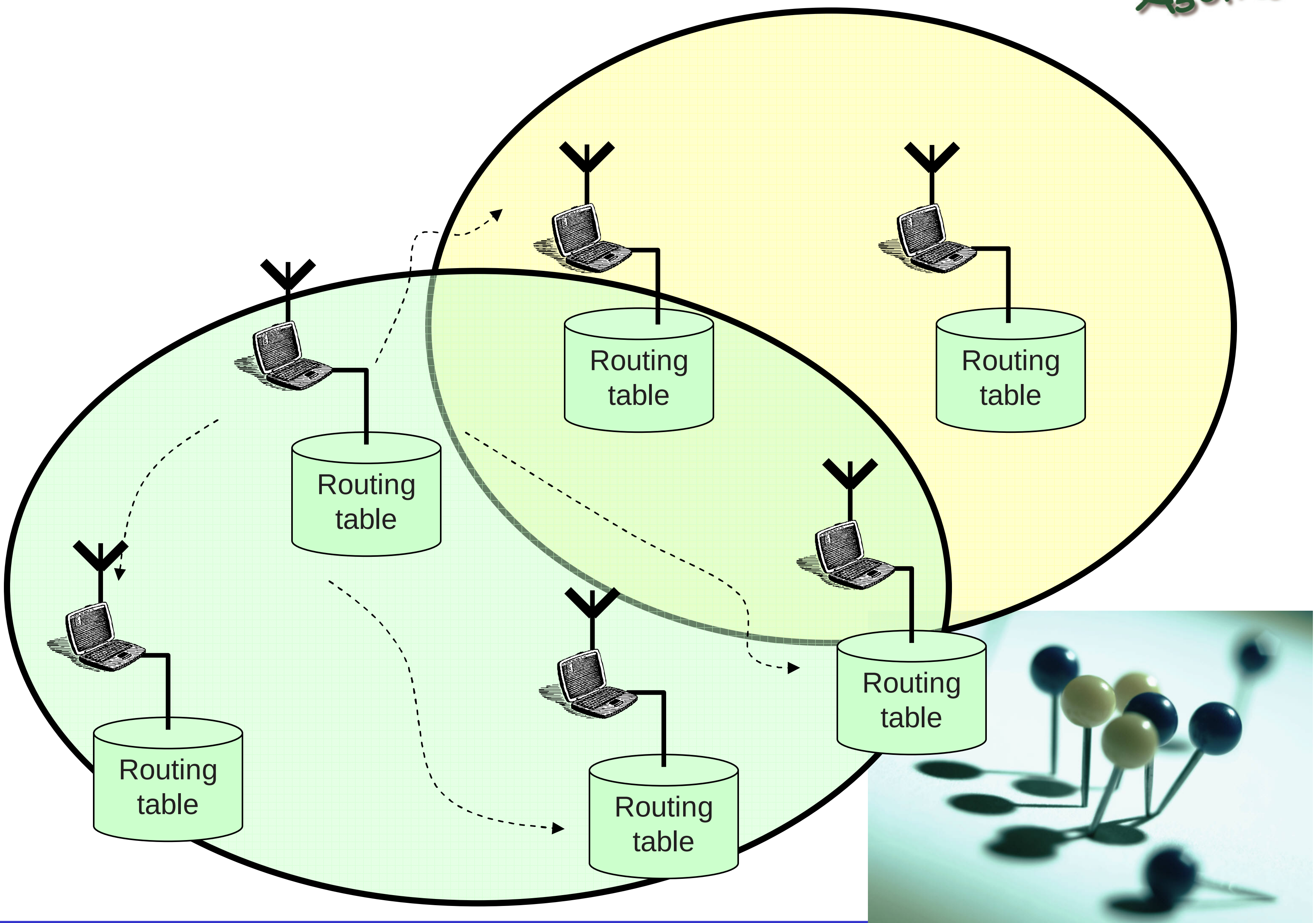
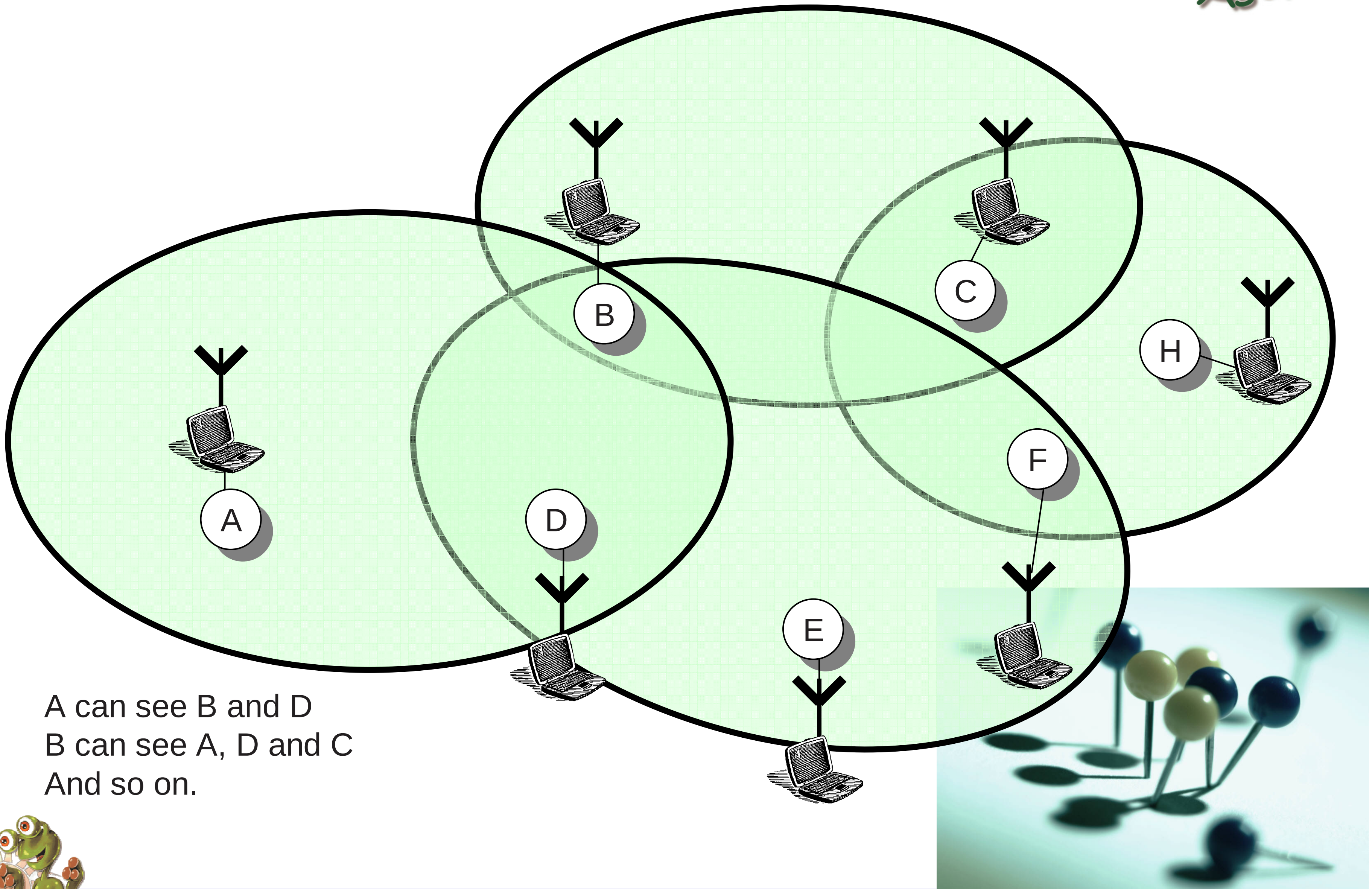
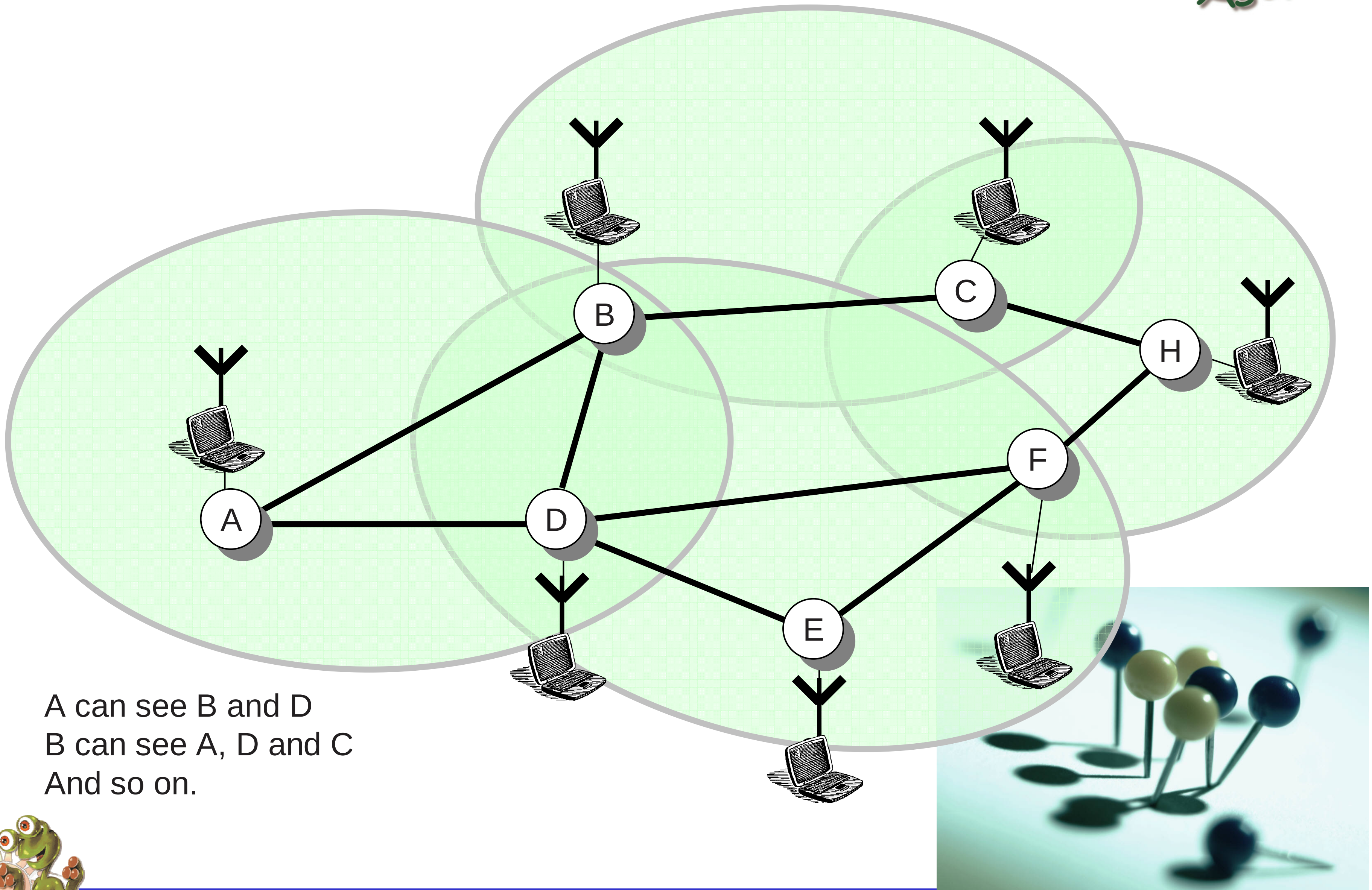


Table-based Routing



A can see B and D
B can see A, D and C
And so on.

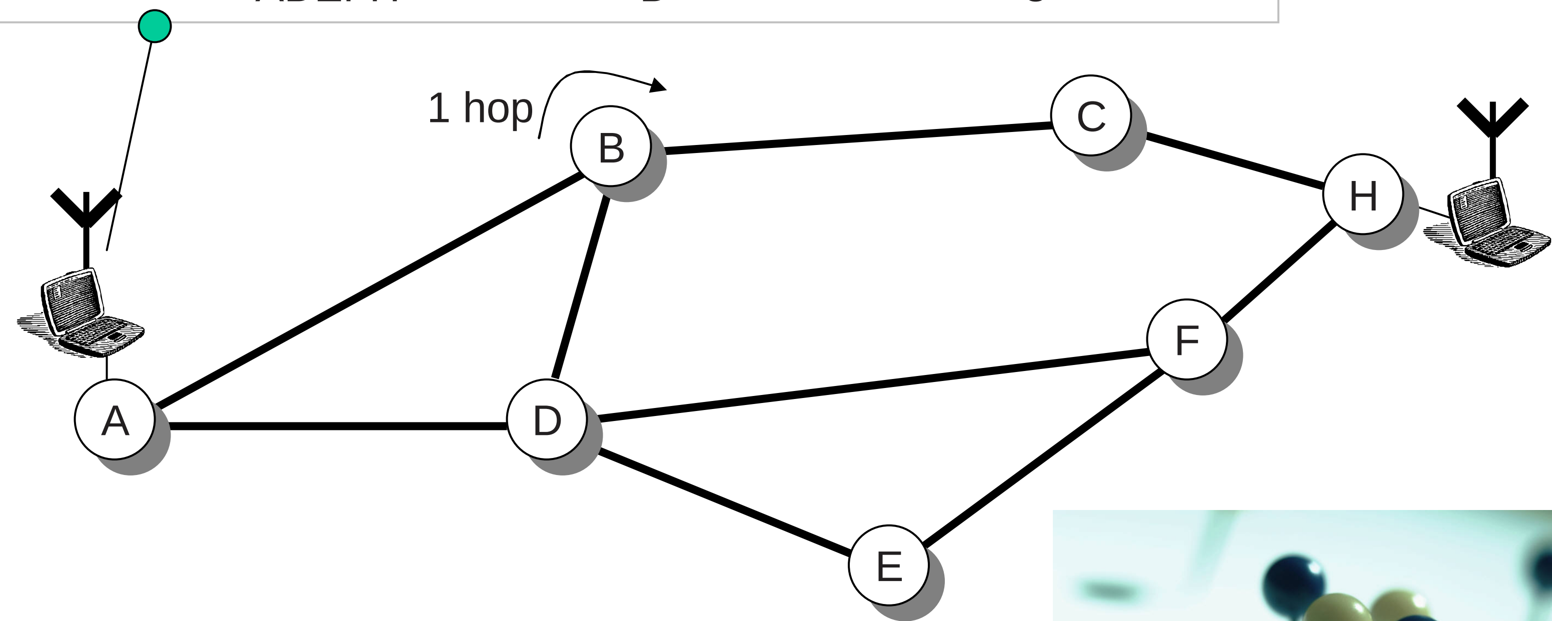
Example of wireless domains

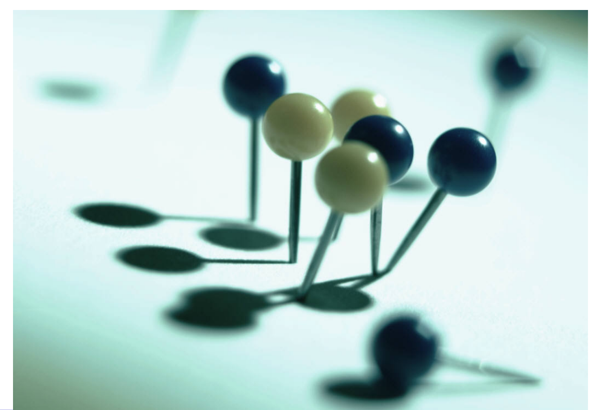


A can see B and D
B can see A, D and C
And so on.

Example of wireless domains

Destination	Route	Next Hop	No. of hops
H	ABCH	B	2
	ABDFH	B	3
	ABDEFH	B	4
	ADFH	D	2
	ADEFH	D	3

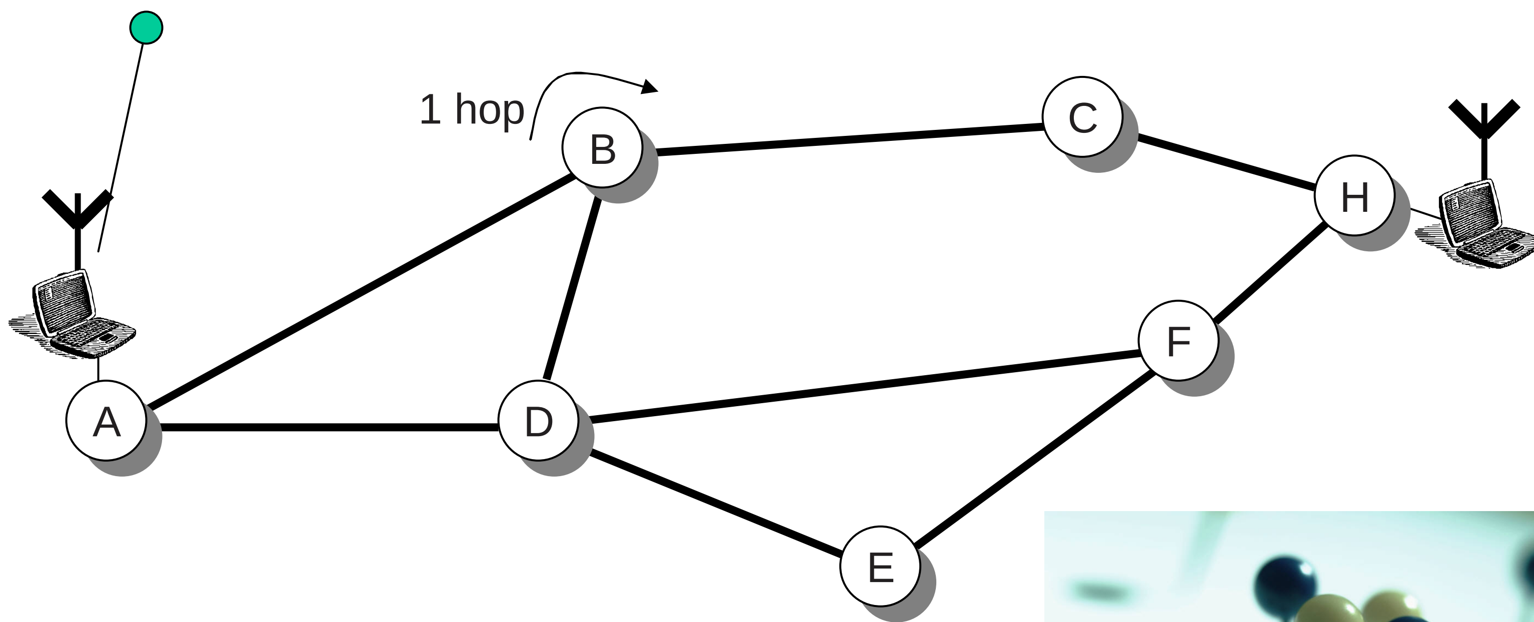


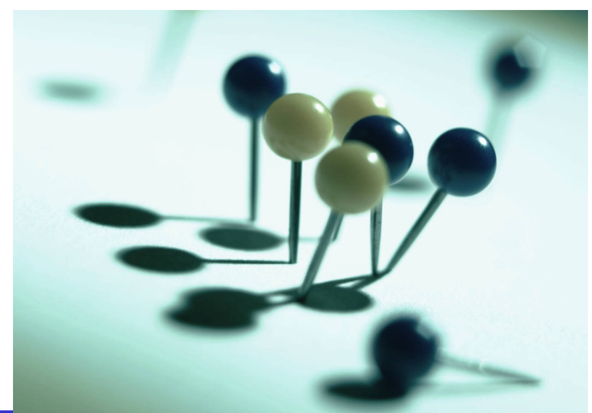


Destination-Sequenced Distance-Vector Routing

Destination	Route	Next Hop	No. of hops
H	ABCH	B	2
	ADFH	D	2

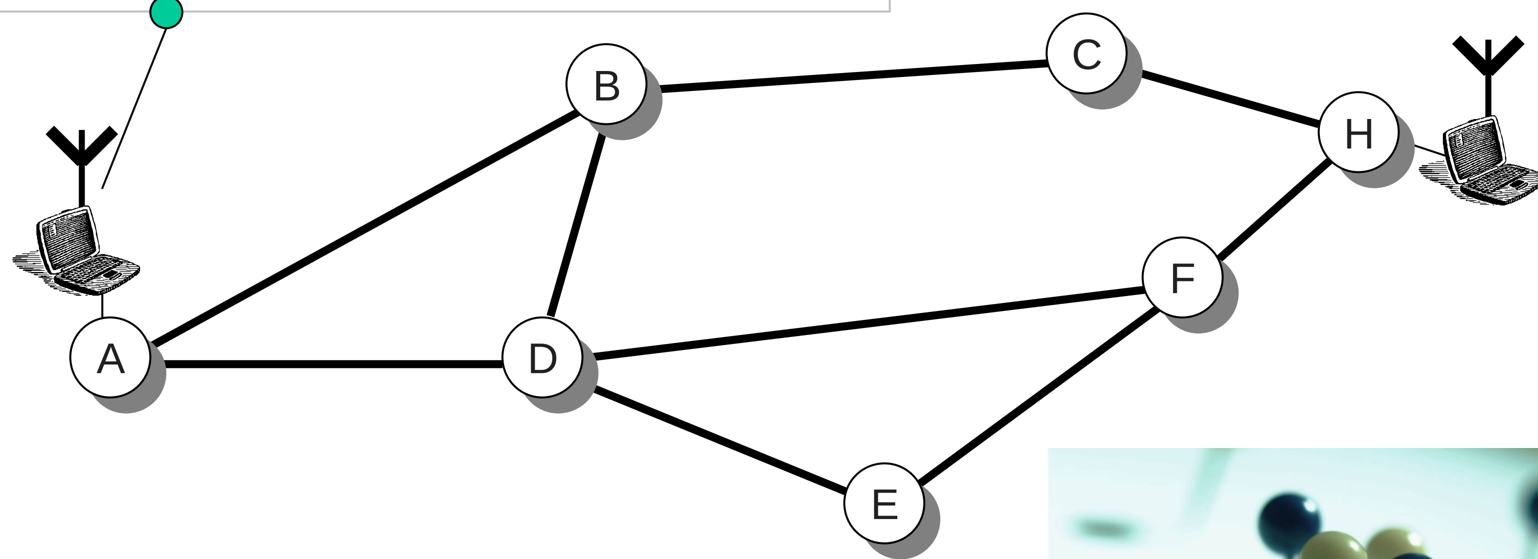
Best route is either through B or D

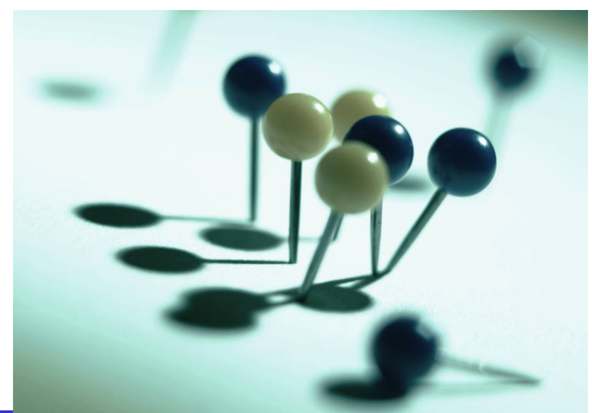




Destination-Sequenced Distance-Vector Routing

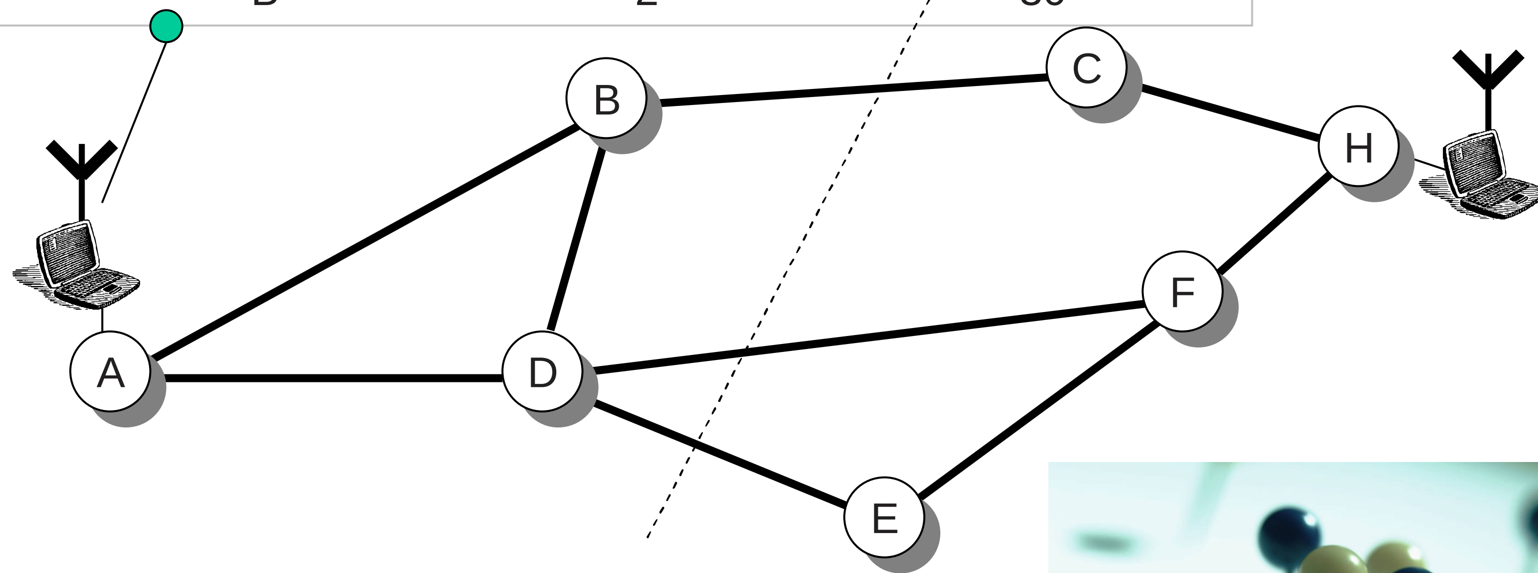
Destination	Next Hop	No. of hops
B	B	0
C	B	1
D	D	0
E	D	1
F	D	1
H	B	2



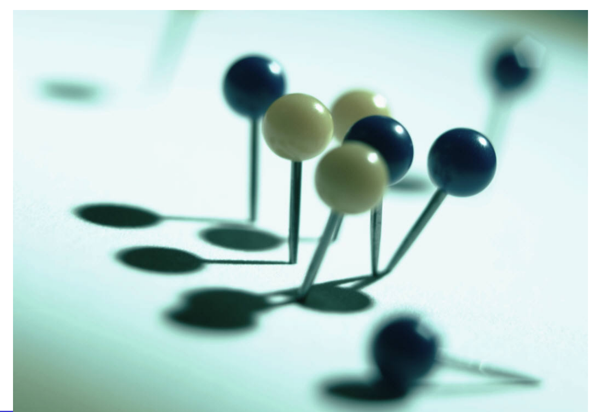


Destination-Sequenced Distance-Vector Routing

Destination	Next Hop	No. of hops	Sequence
B	B	0	5
C	B	1	6
D	D	0	10
E	D	1	1
F	D	1	4
H	B	2	30

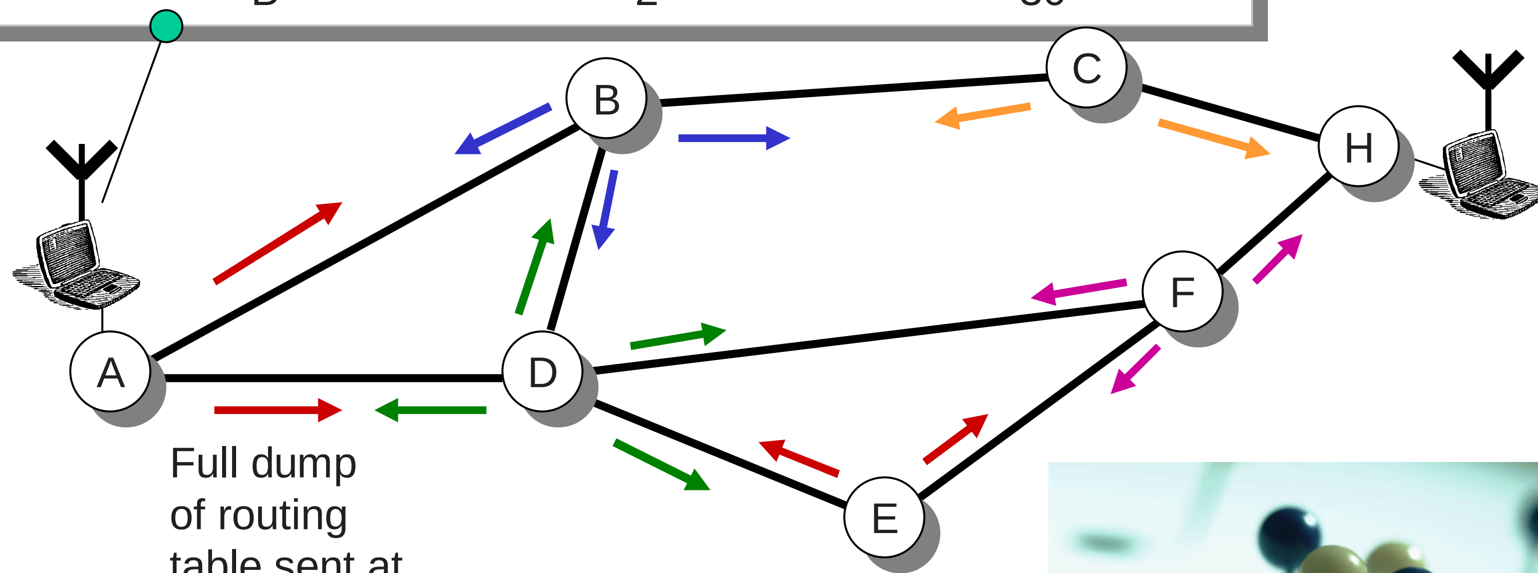


The sequence number
Is defined by the destination
node, and is used to determine
how up-to-date a route is.



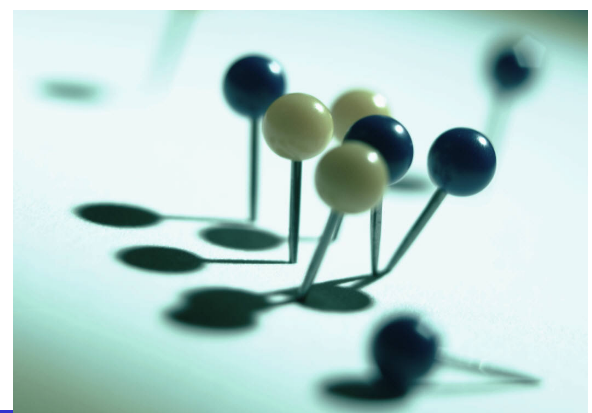
DSDV (with Sequence Number)

Destination	Next Hop	No. of hops	Sequence
B	B	0	5
C	B	1	6
D	D	0	10
E	D	1	1
F	D	1	4
H	B	2	30



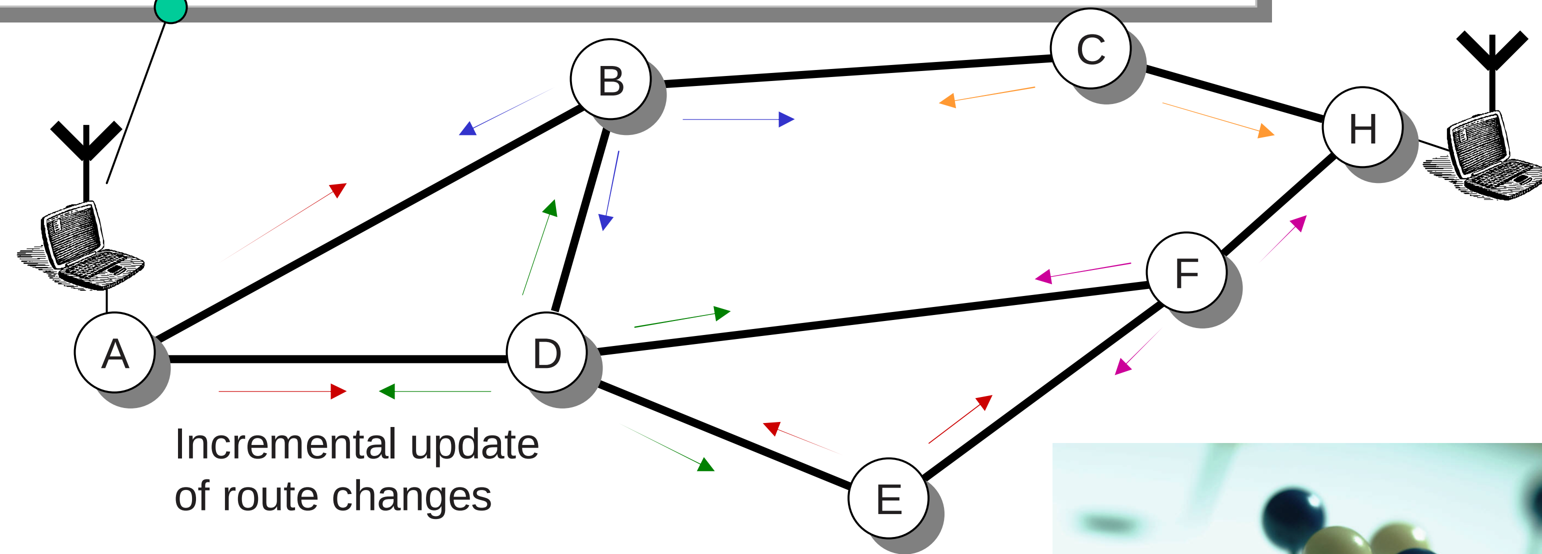
Full dump of routing table sent at startup or where there is large-scale mobility

NPDU - Network Packet Data Unit

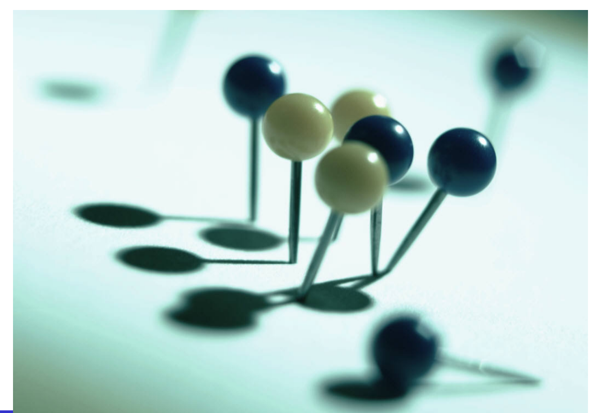


DSDV (Full dump of routing table)

Destination	Next Hop	No. of hops	Sequence
B	B	0	5
C	B	1	6
D	D	0	10
E	D	1	1
F	D	1	4
H	B	2	30

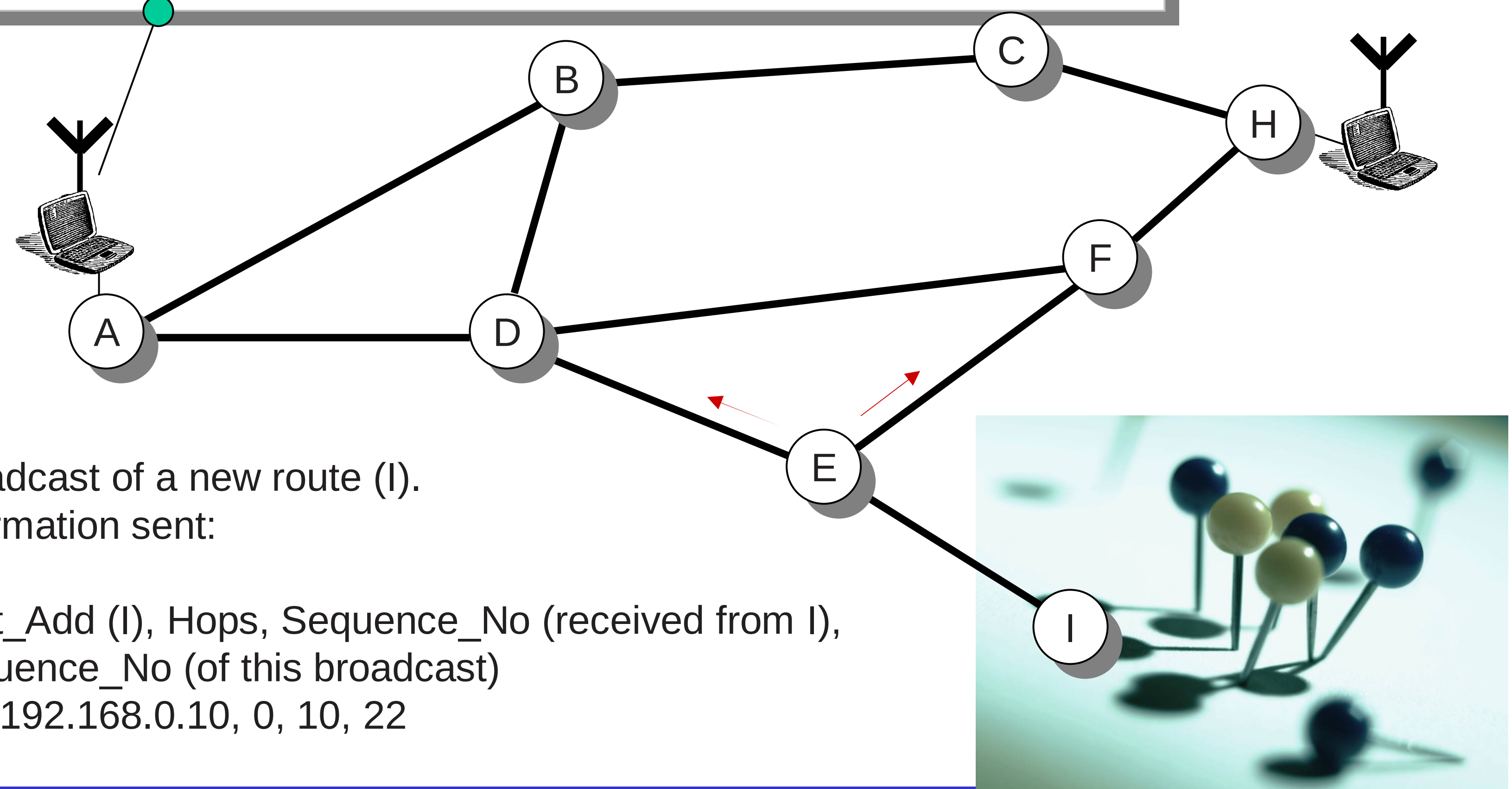


NPDU - Network Packet Data Unit



DSDV (Incremental update)

Destination	Next Hop	No. of hops	Sequence
B	B	0	5
C	B	1	6
D	D	0	10
E	D	1	1
F	D	1	4
H	B	2	30

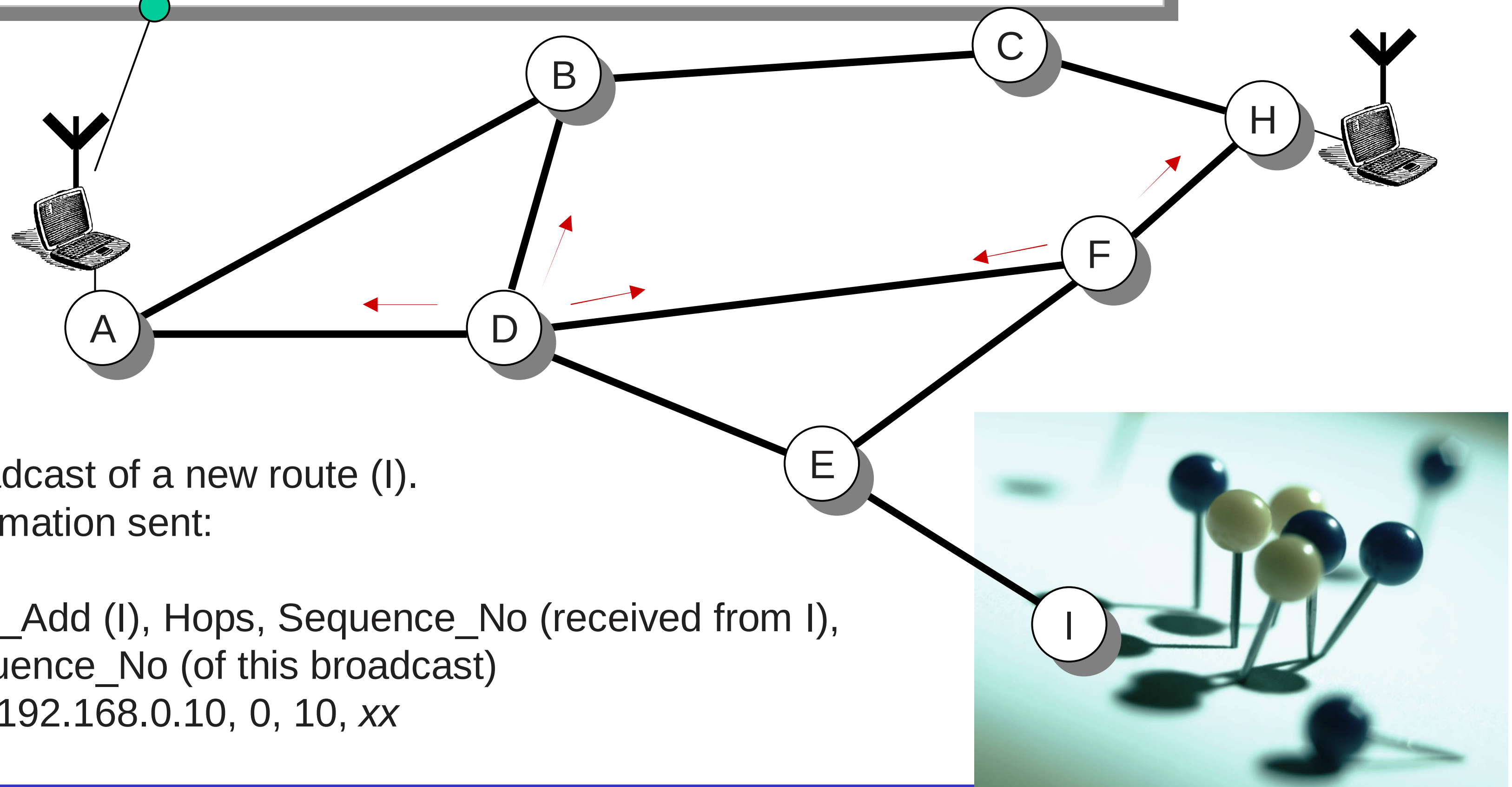


Broadcast of a new route (I).
Information sent:

Dest_Add (I), Hops, Sequence_No (received from I), Sequence_No (of this broadcast)
e.g. 192.168.0.10, 0, 10, 22

DSDV (New route discovery)

Destination	Next Hop	No. of hops	Sequence
B	B	0	5
C	B	1	6
D	D	0	10
E	D	1	1
F	D	1	4
H	B	2	30

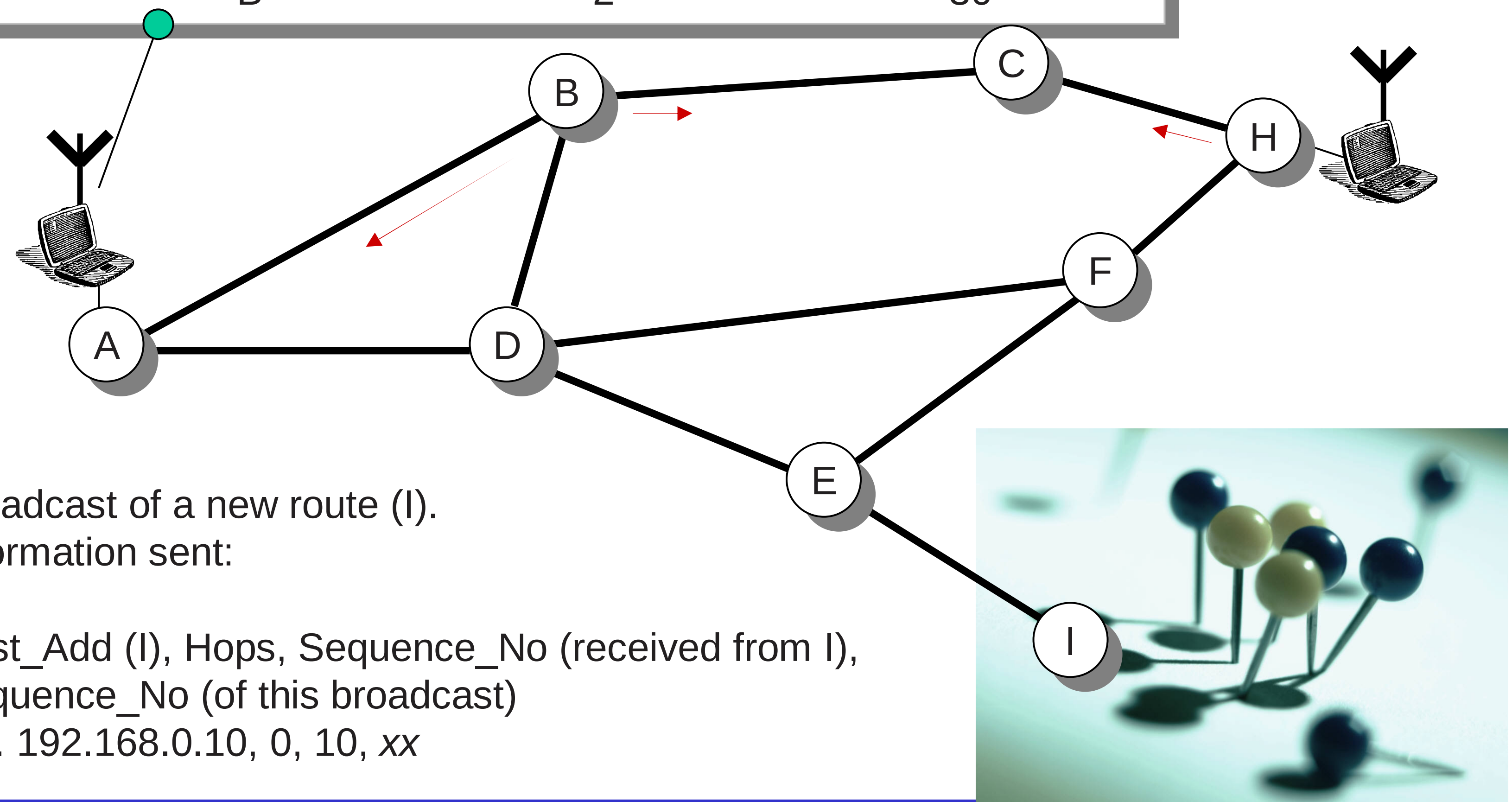


Broadcast of a new route (I).
Information sent:

Dest_Add (I), Hops, Sequence_No (received from I),
Sequence_No (of this broadcast)
e.g. 192.168.0.10, 0, 10, *xx*

DSDV (New route discovery)

Destination	Next Hop	No. of hops	Sequence
B	B	0	5
C	B	1	6
D	D	0	10
E	D	1	1
F	D	1	4
H	B	2	30

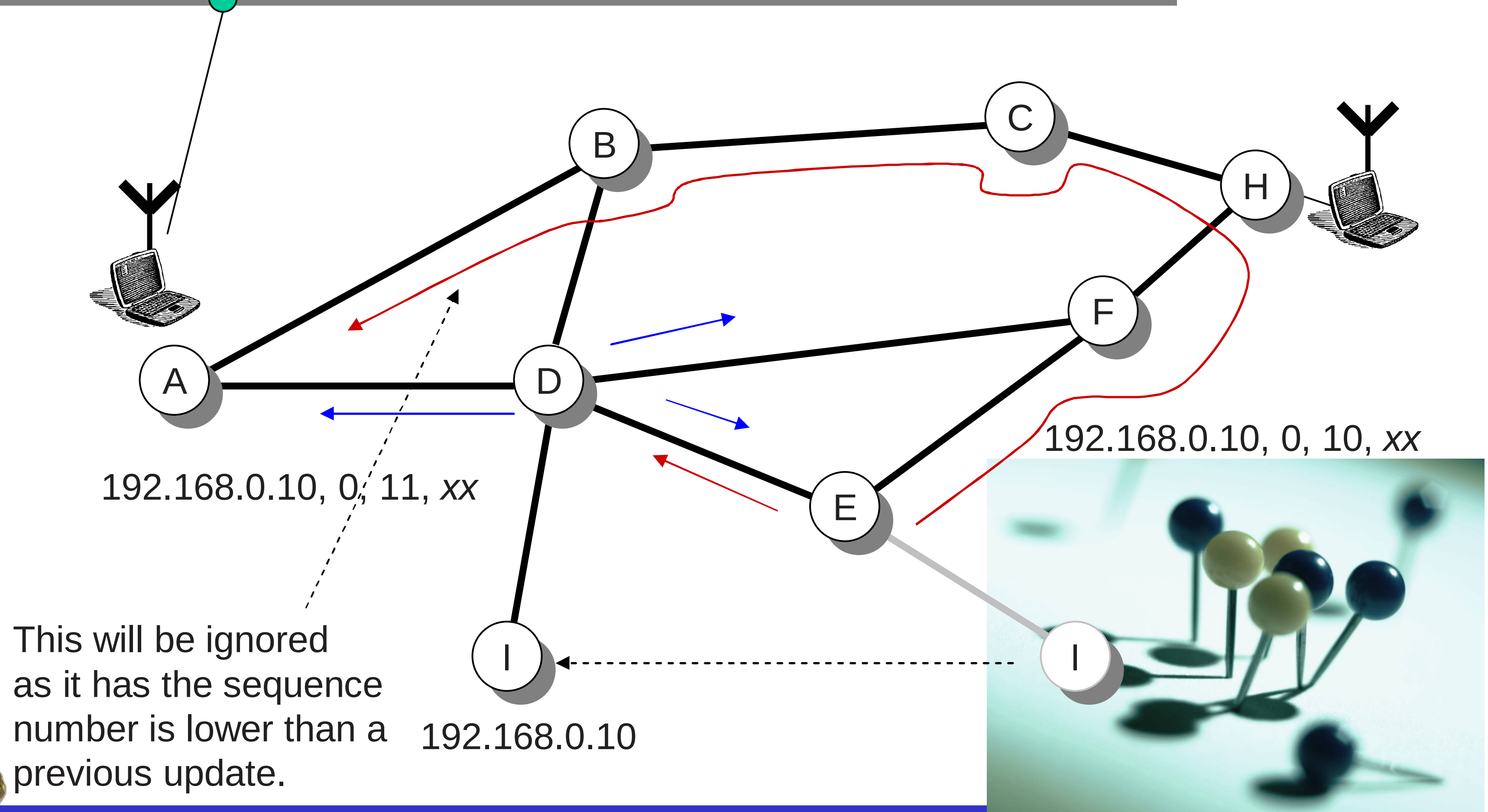


Broadcast of a new route (I).
Information sent:

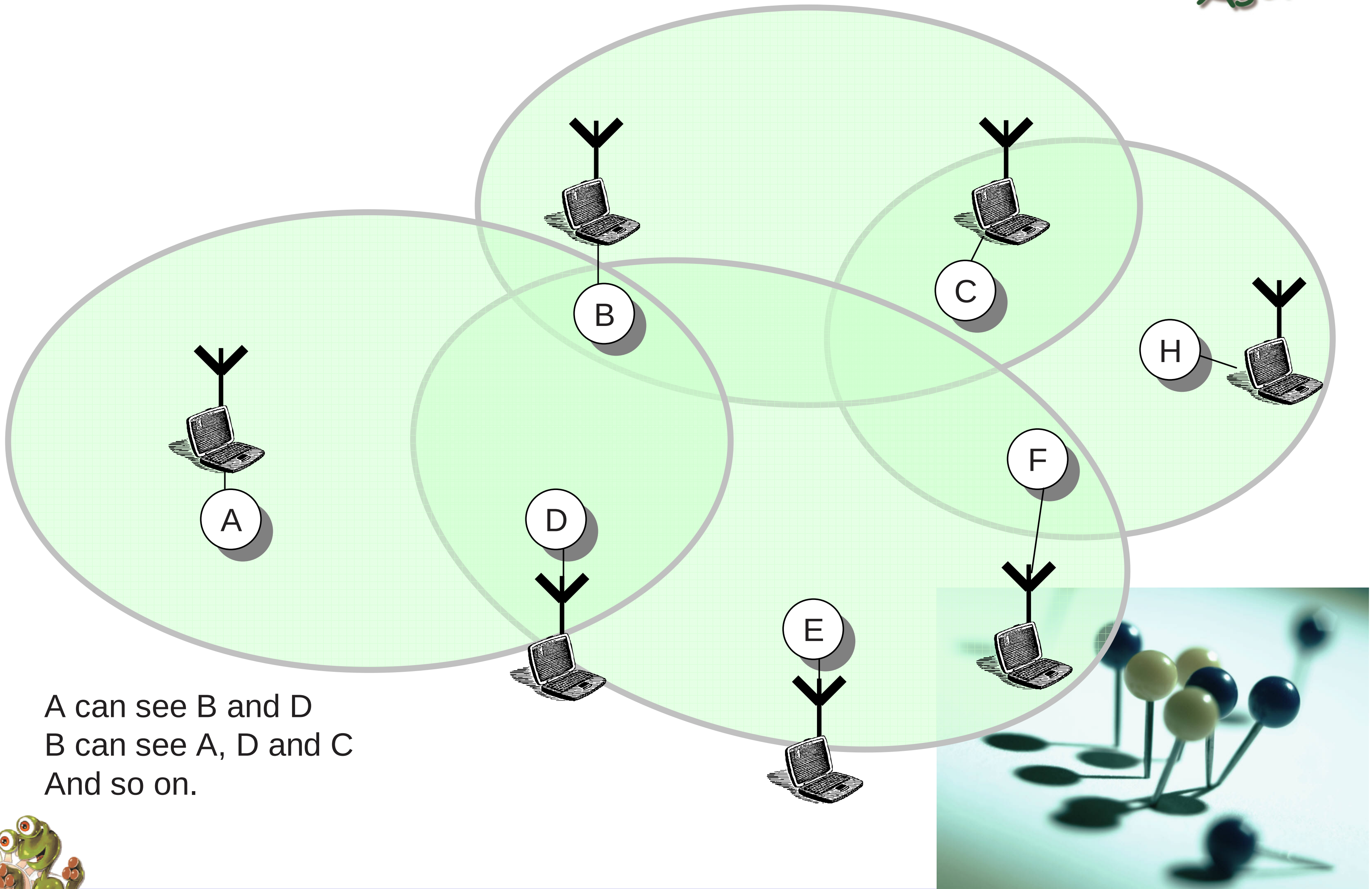
Dest_Add (I), Hops, Sequence_No (received from I), Sequence_No (of this broadcast)
e.g. 192.168.0.10, 0, 10, *xx*

DSDV (New route discovery)

Destination	Next Hop	No. of hops	Sequence
B	B	0	5
::			
H	B	2	30
I	D	1	11



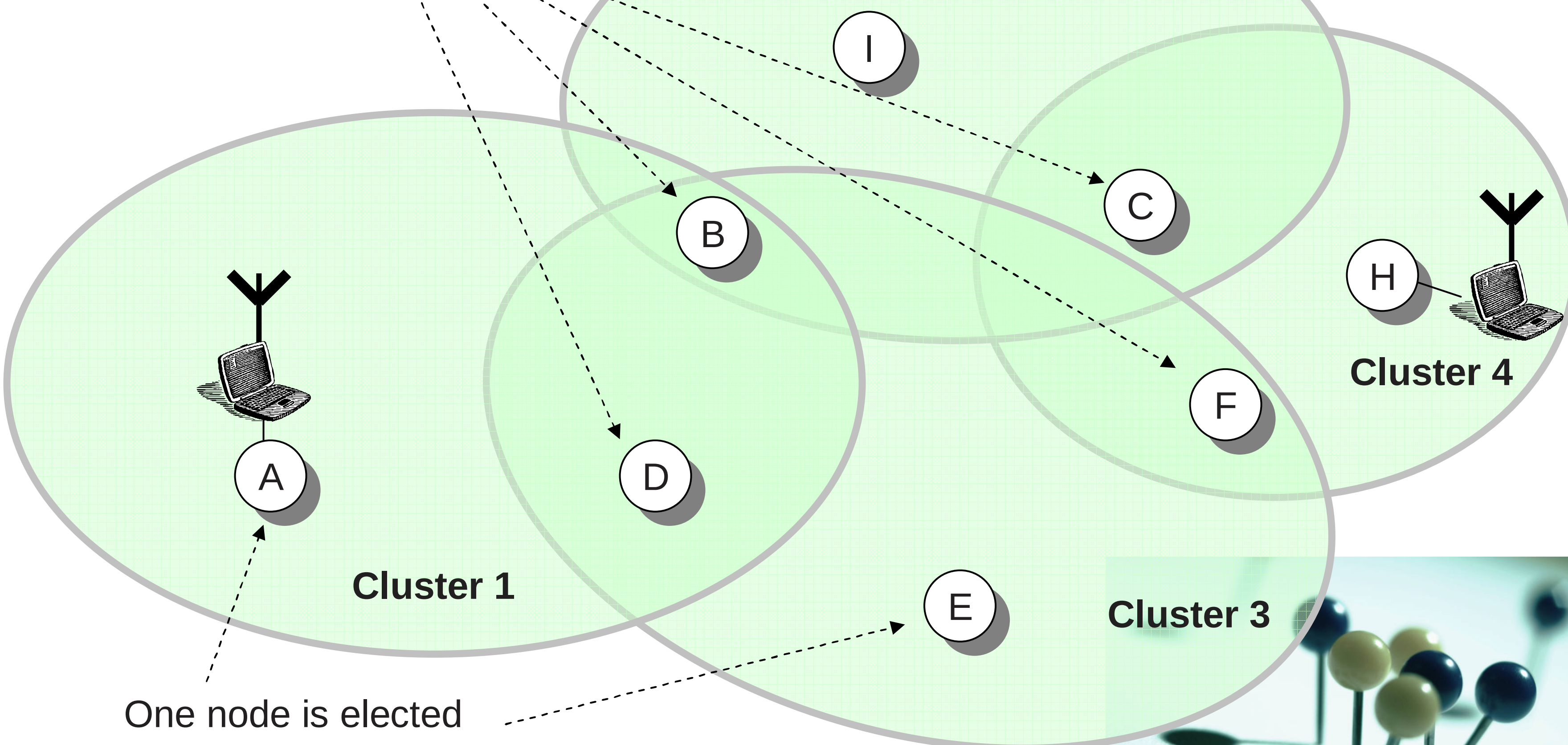
DSDV (New route discovery)



A can see B and D
B can see A, D and C
And so on.

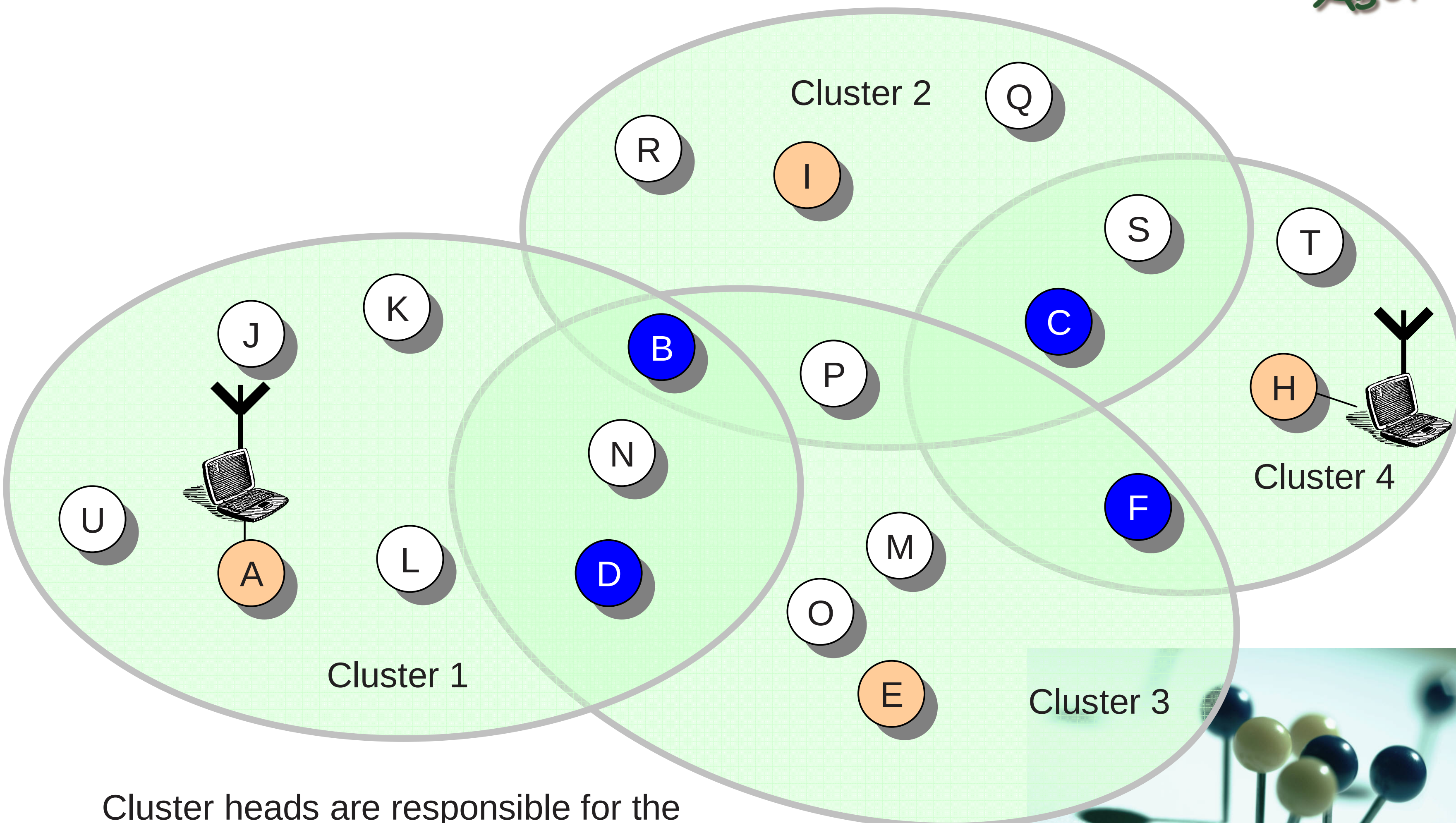
Clusterhead Gateway Switch Routing (CGSR)

Gateway node is used to route data between clusters



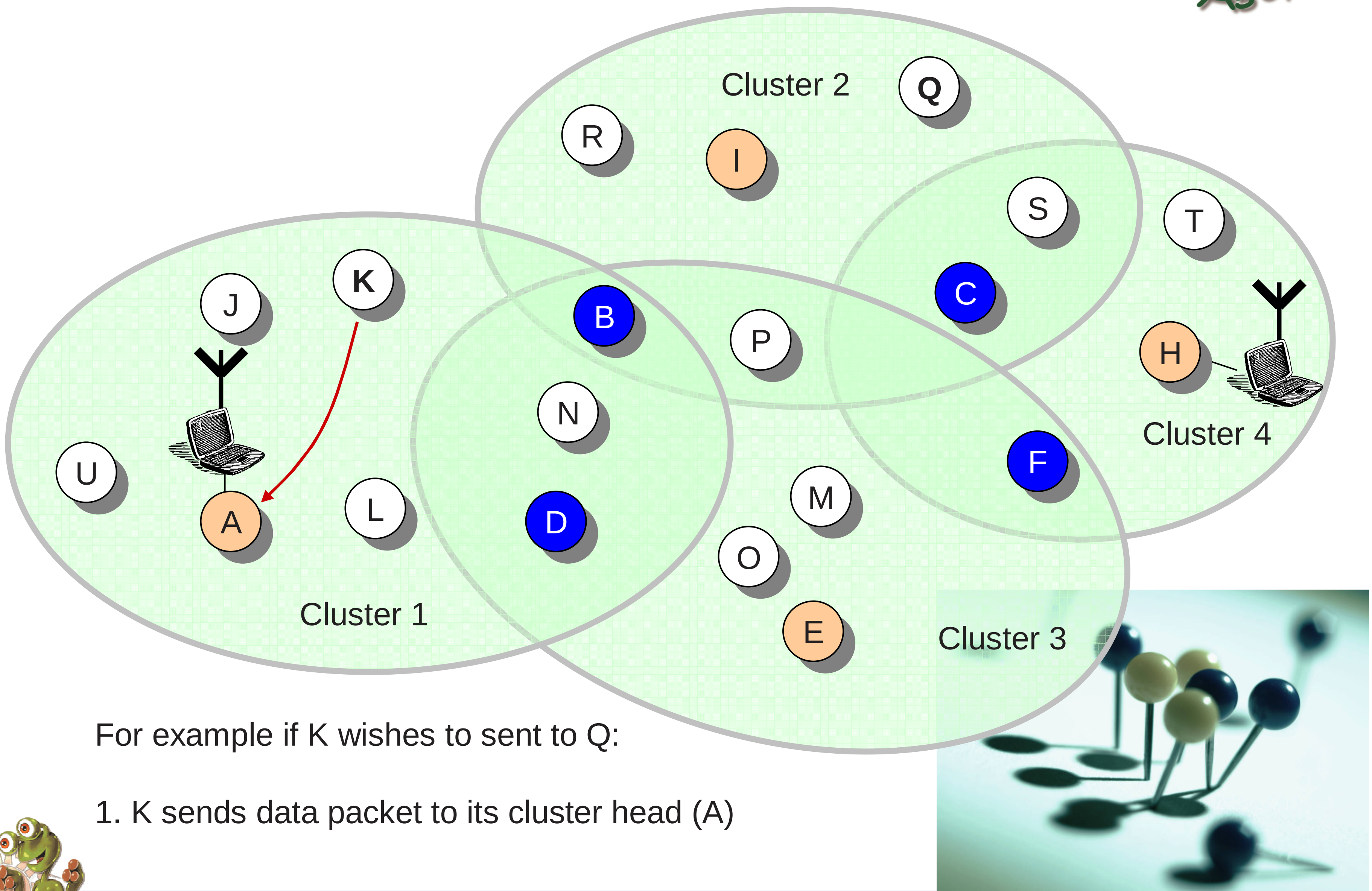
One node is elected a cluster head (normally based on geographical knowledge)

Clusterhead Gateway Switch Routing



Cluster heads are responsible for the clusters and locating clusters outwith its cluster.
A node which can be contacted by two cluster heads is elected as the gateway.

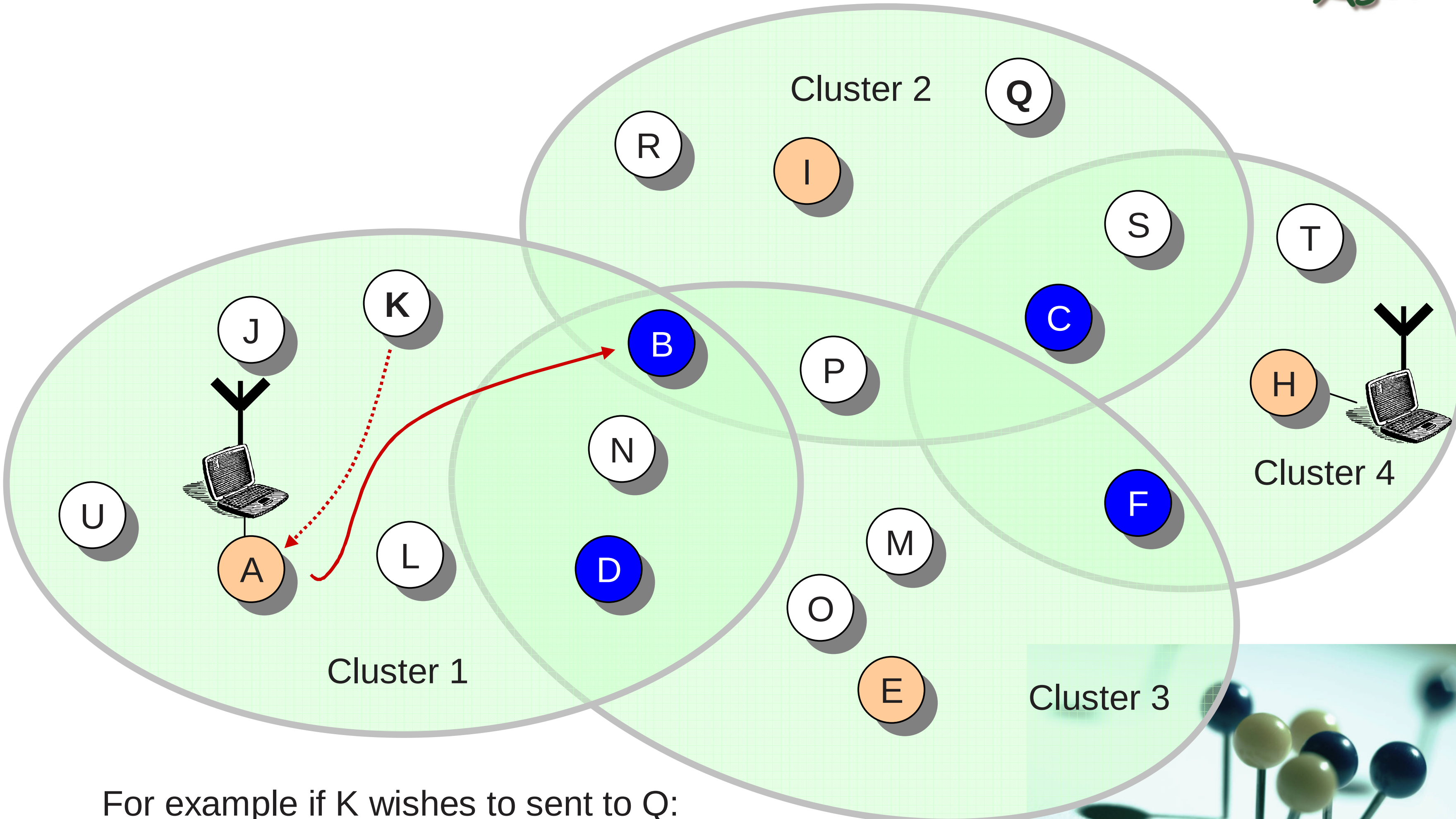
Clusterhead Gateway Switch Routing



For example if K wishes to sent to Q:

1. K sends data packet to its cluster head (A)

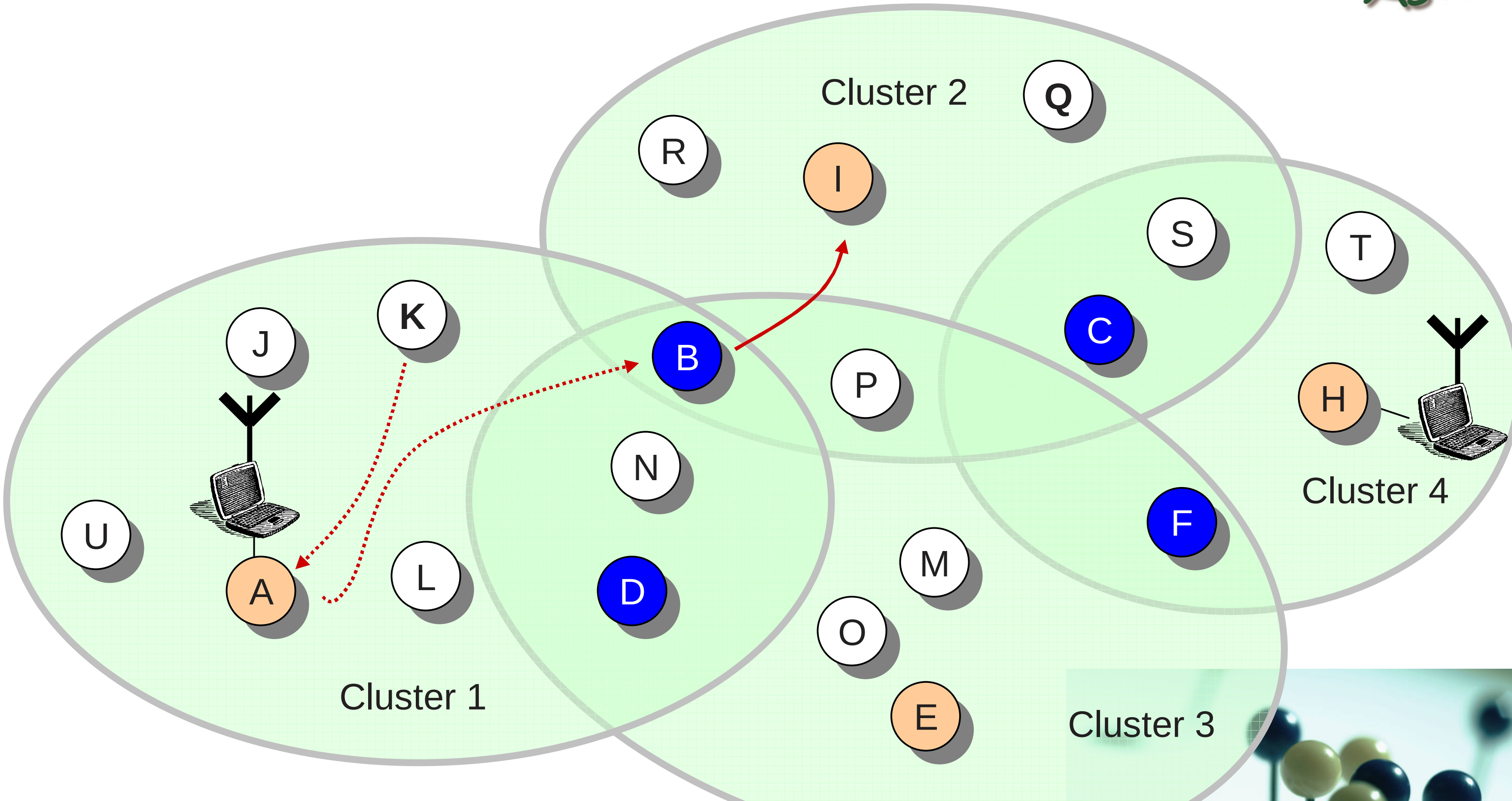
Clusterhead Gateway Switch Routing



For example if K wishes to sent to Q:

2. Cluster head (A) sends the data packet to the required gateway (B)

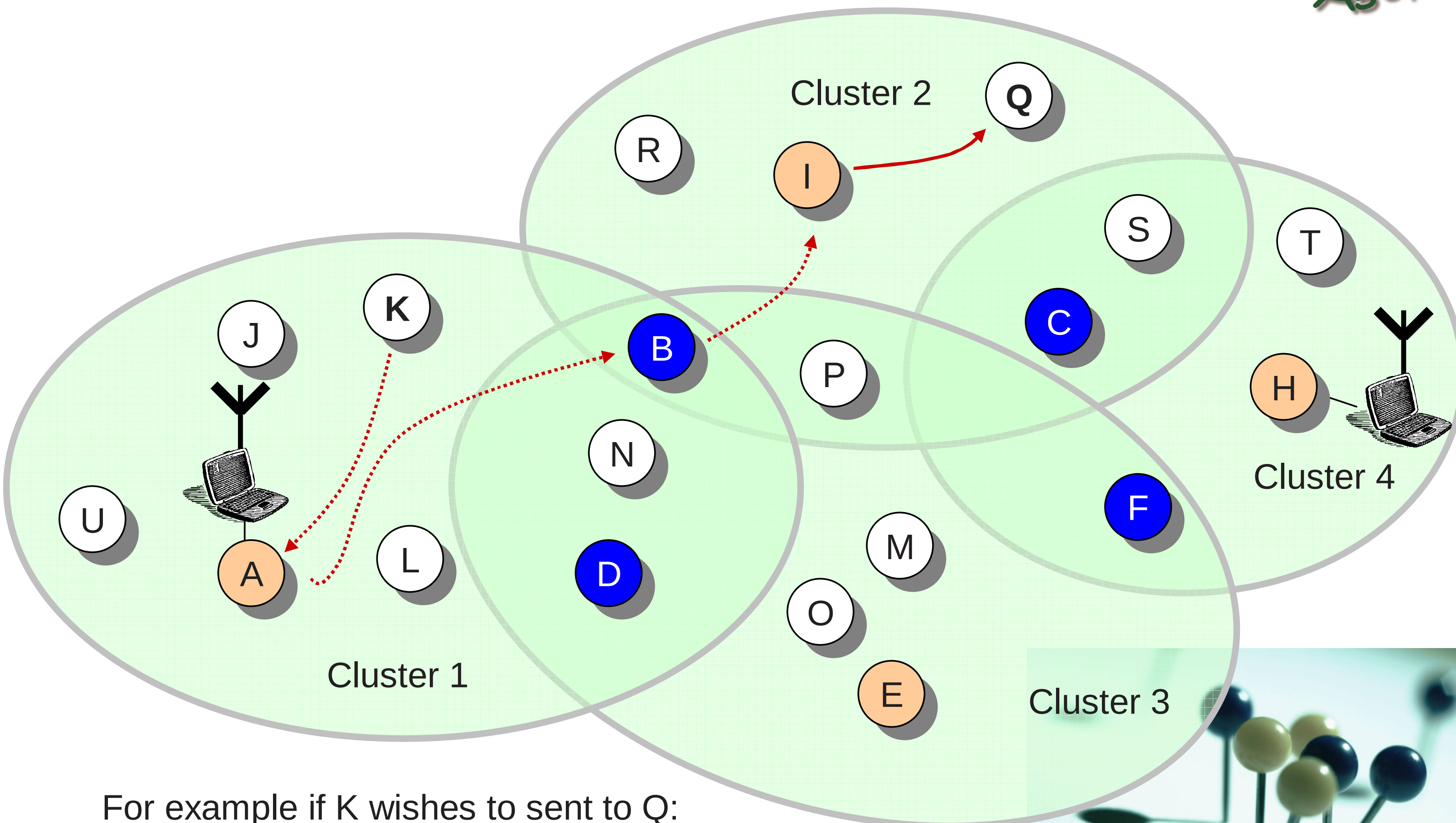
Clusterhead Gateway Switch Routing



For example if K wishes to sent to Q:

3. The gateway node (B) forwards the data packet to the cluster head (I)

Clusterhead Gateway Switch Routing



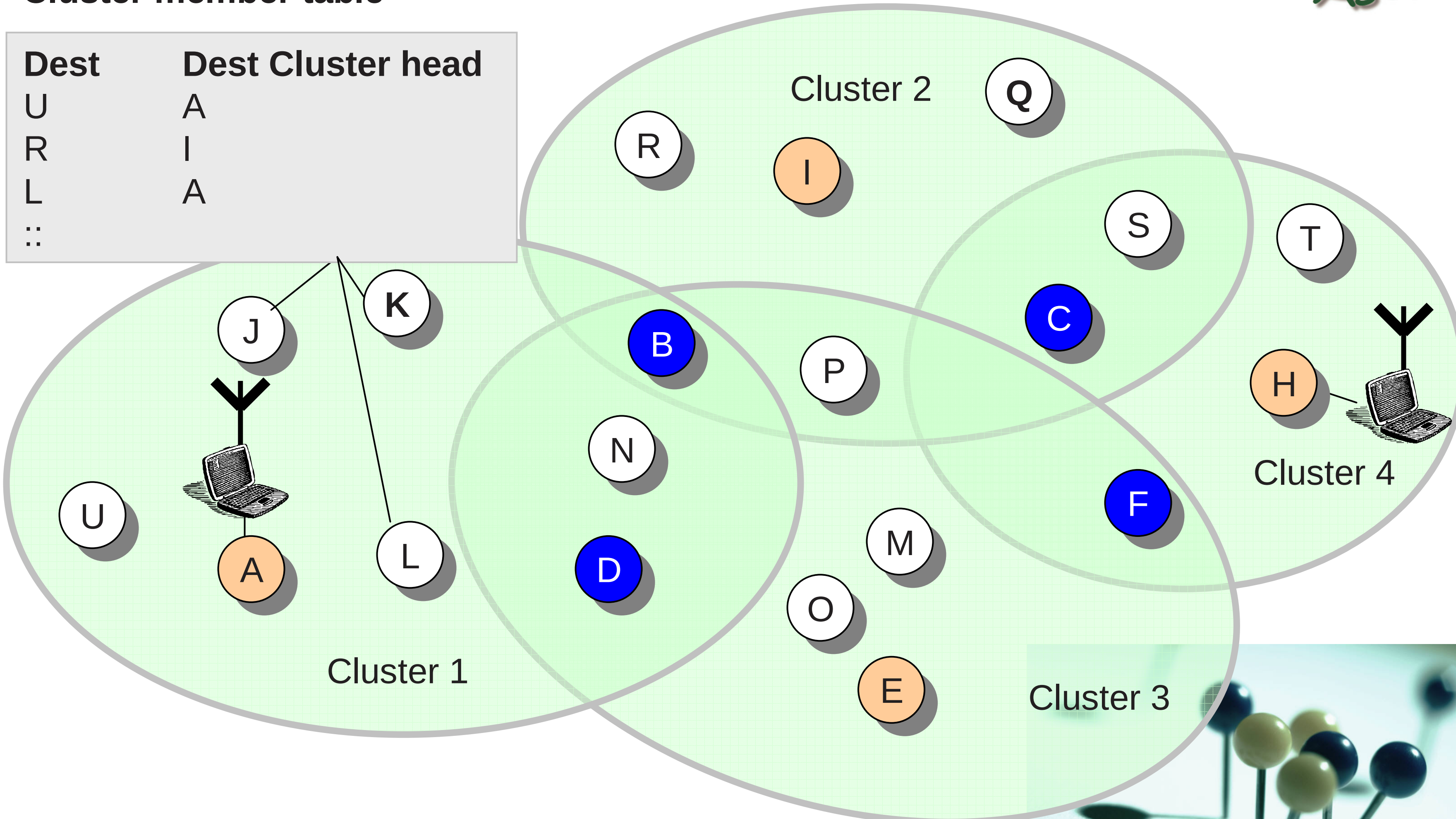
For example if K wishes to sent to Q:

4. Cluster head forwards the data packet to the required node (Q)

Clusterhead Gateway Switch Routing

Cluster member table

Dest	Dest Cluster head
U	A
R	I
L	A
::	



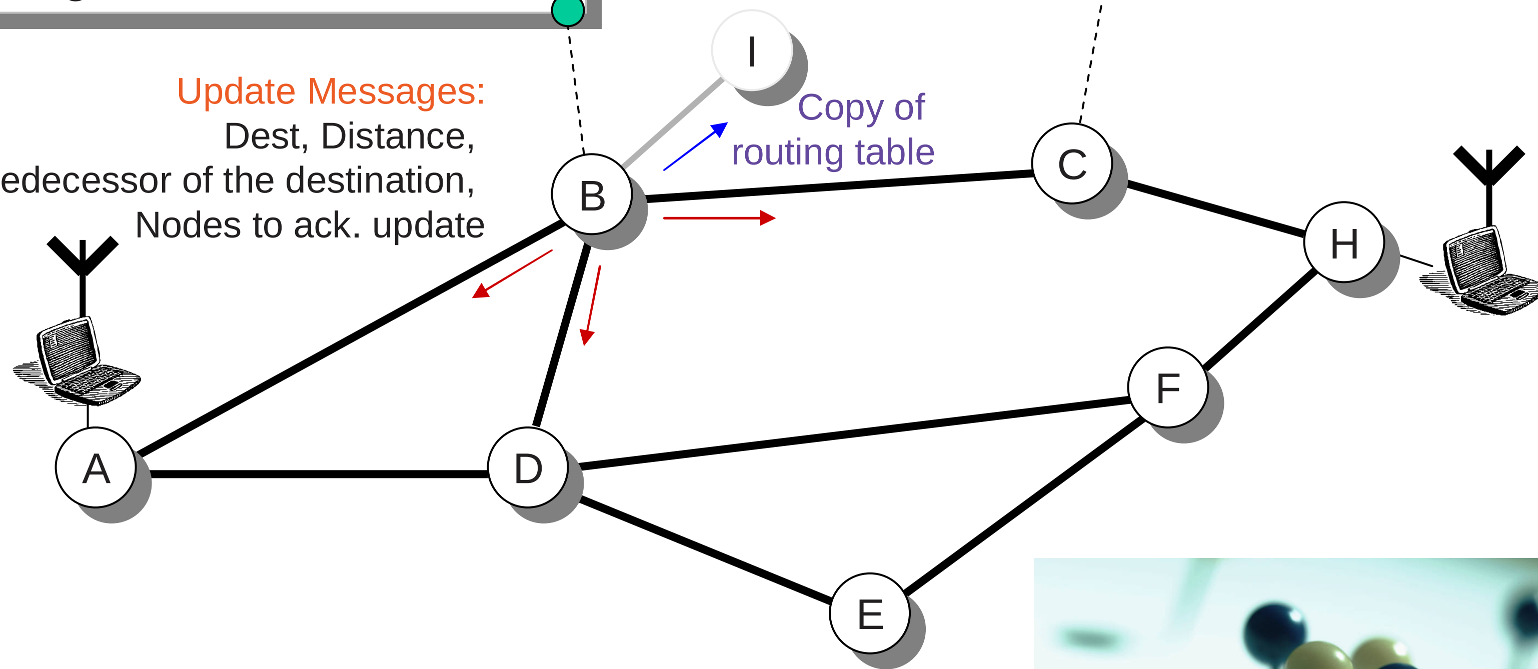
Each node stores a Cluster Member Table which is broadcast by each node at various times

CGSR - Cluster Member Table

Distance Table.
Routing Table.
Link-cost Table.
Message retransmission Table.

Distance Table.
Routing Table.
Link-cost Table.
Message retransmission Table.

Update Messages:
Dest, Distance,
Predecessor of the destination,
Nodes to ack. update

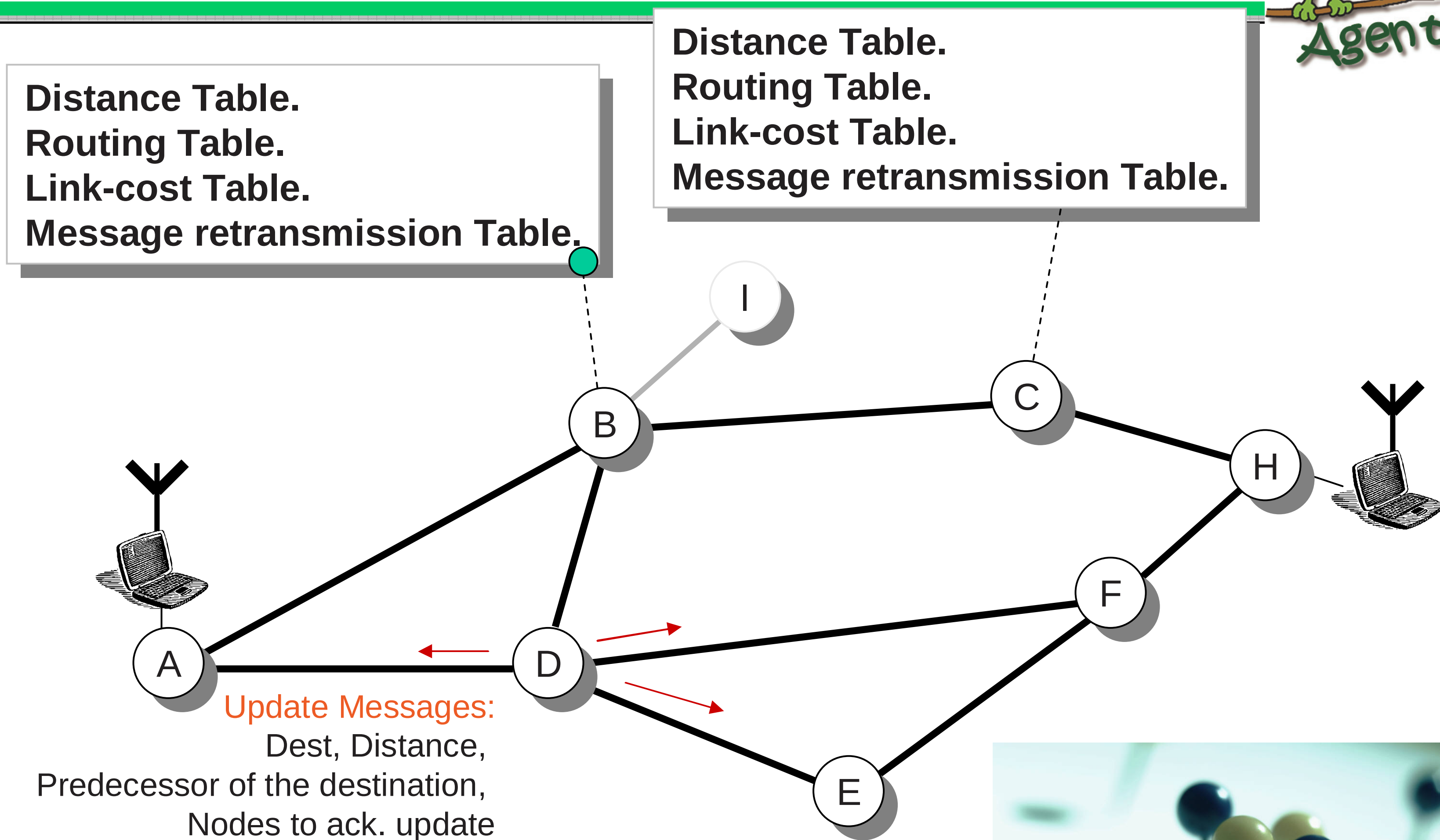


MRT stores the required updates for an update message, and contains:

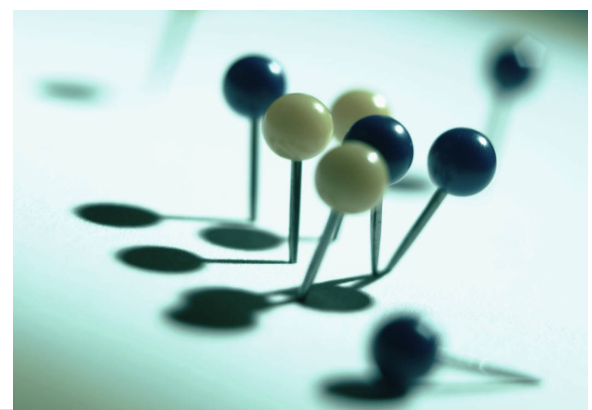
Seq_No_of_Update_Message Retransmission_Counter Acknowledgement_flag List_Of_Updates

Wireless Routing Protocol (WRP)

Wireless Routing Protocol (WRP)

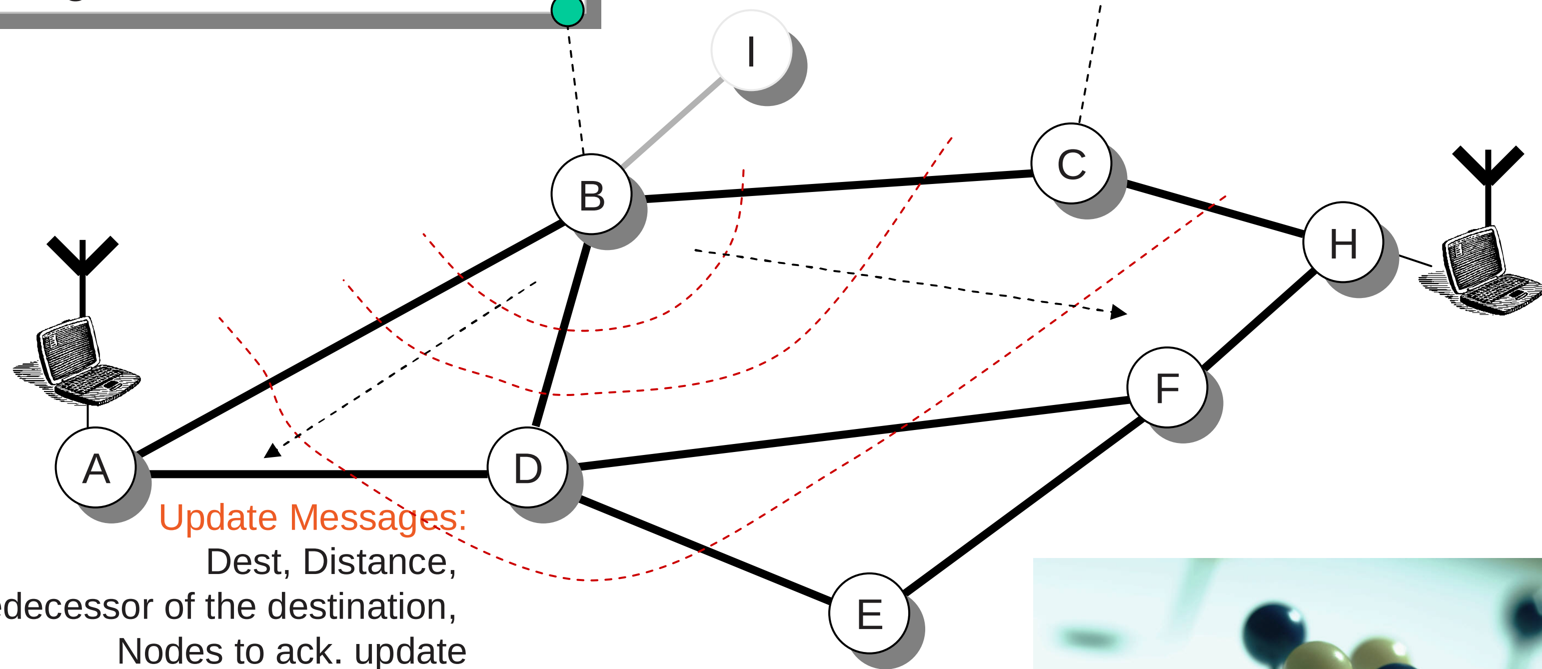


Once received, a node may forward the update message to its own neighbours

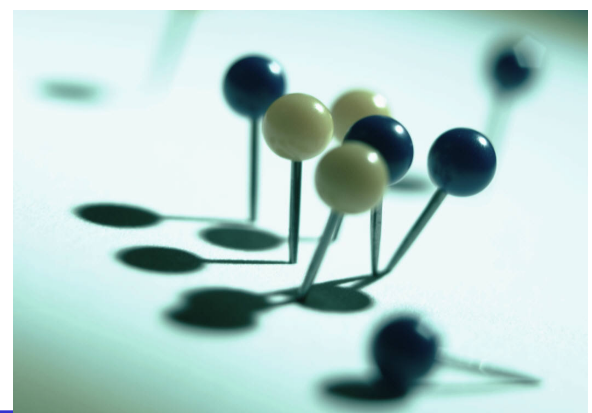


Distance Table.
Routing Table.
Link-cost Table.
Message retransmission Table.

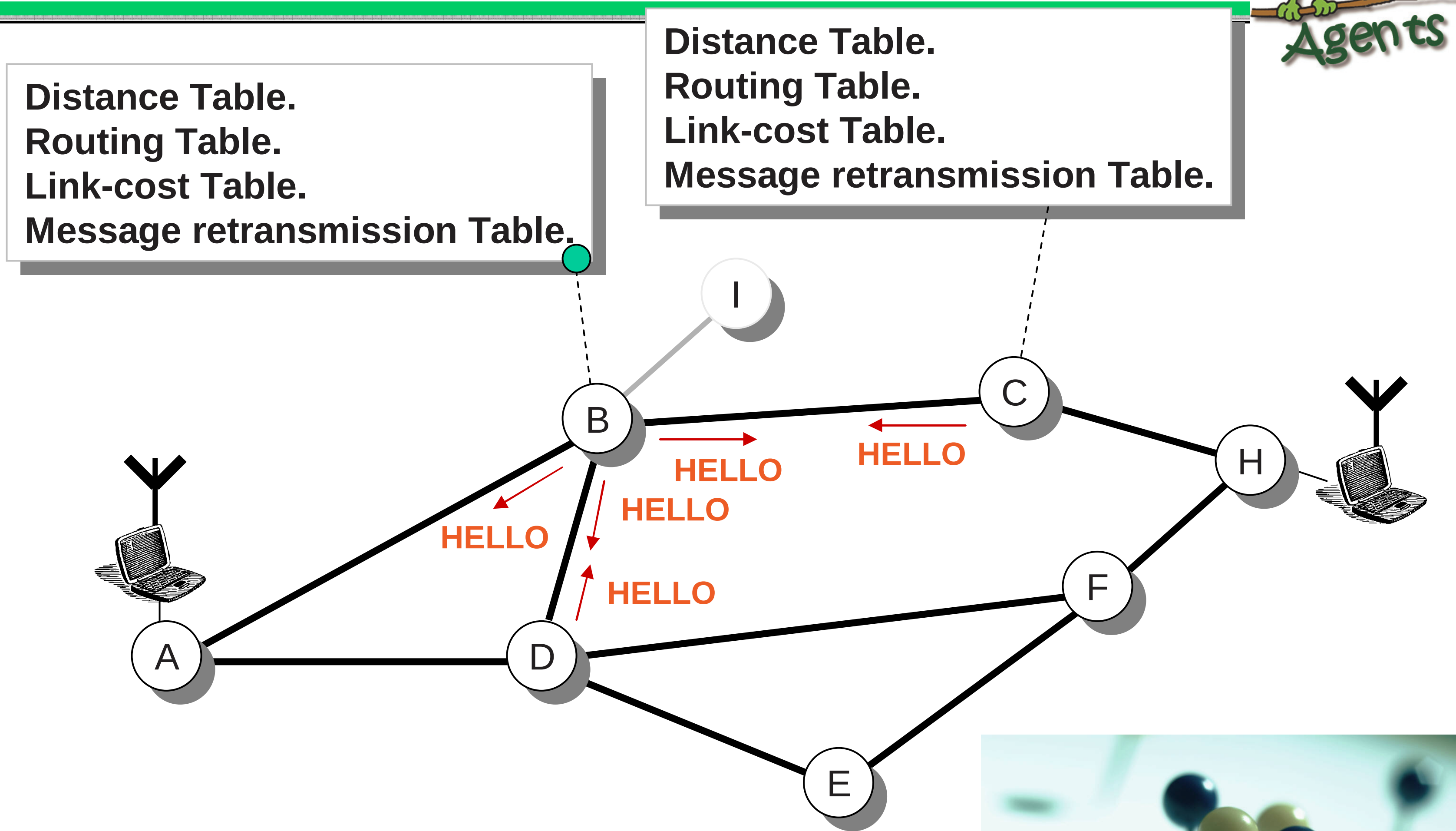
Distance Table.
Routing Table.
Link-cost Table.
Message retransmission Table.



Updates thus propagate throughout the network



Wireless Routing Protocol (WRP)



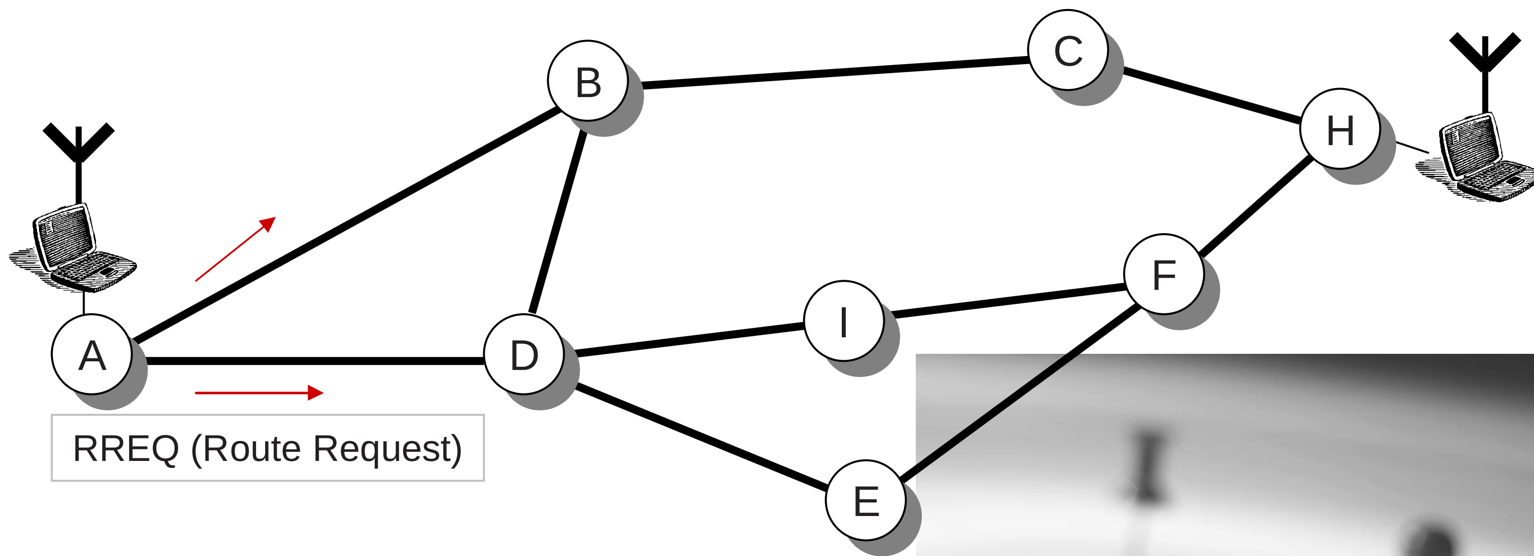
Nodes determine their neighbours by listening to HELLO messages, which are sent out at regular intervals.

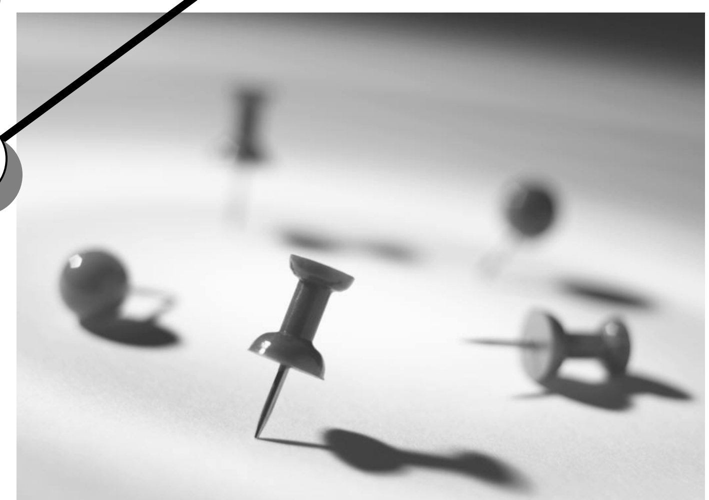
A lack of a HELLO message means that the node have gone.

WRP - HELLO messages

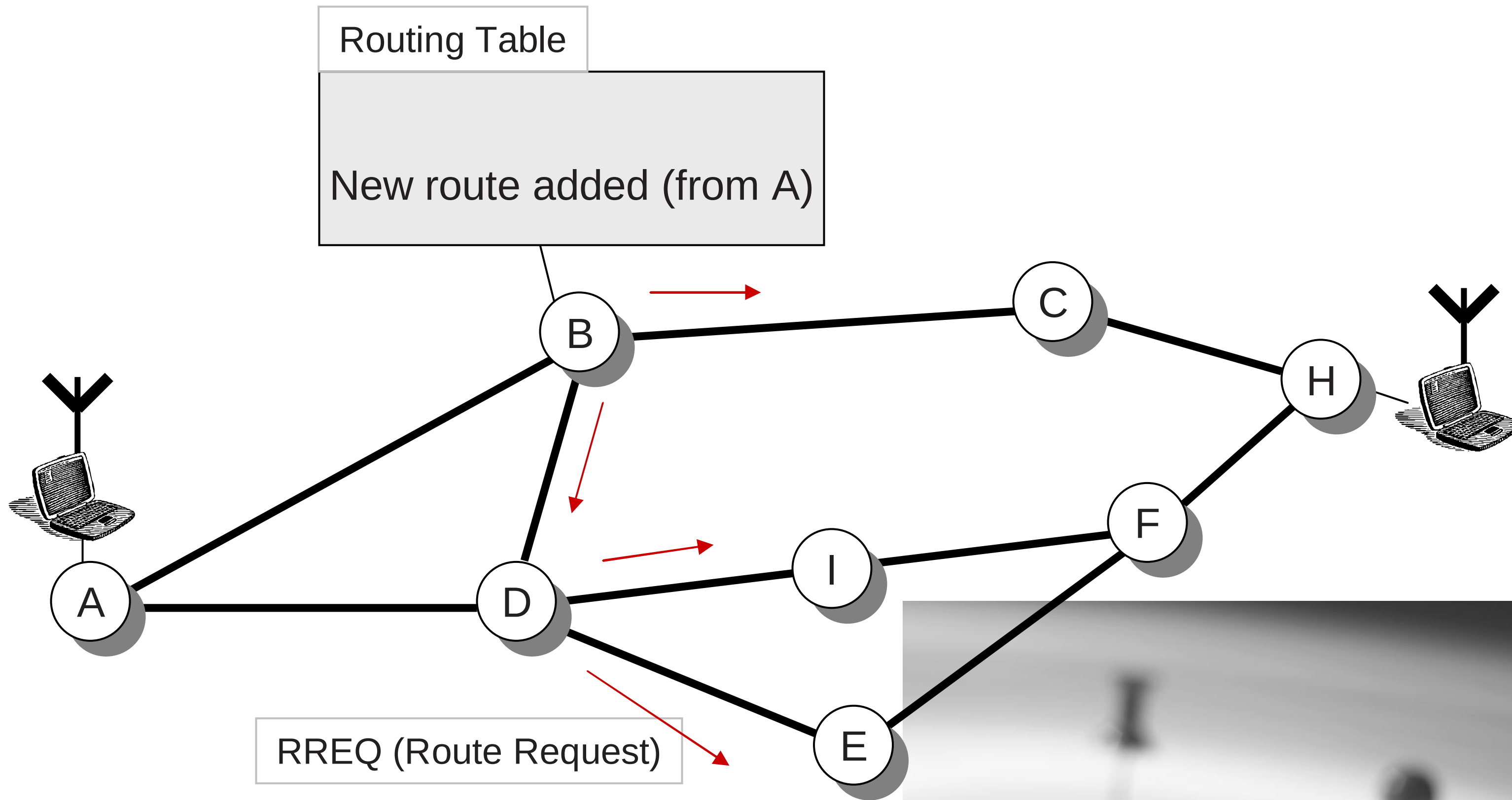
Source-Initiated Ad-hoc Routing

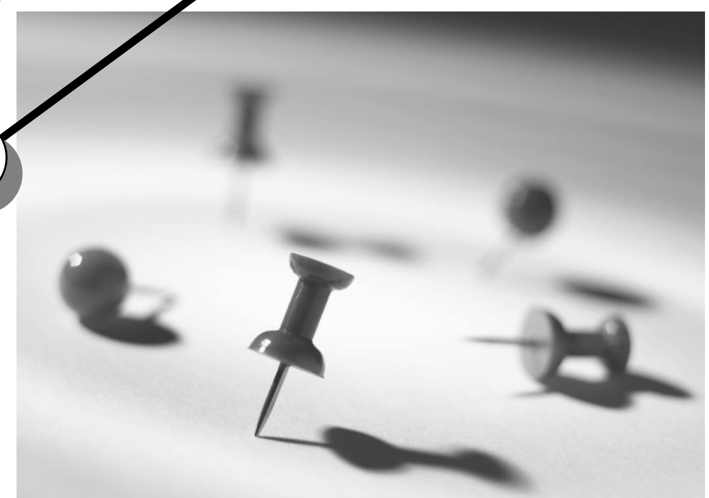
AODV
DSR
TORA
ABR



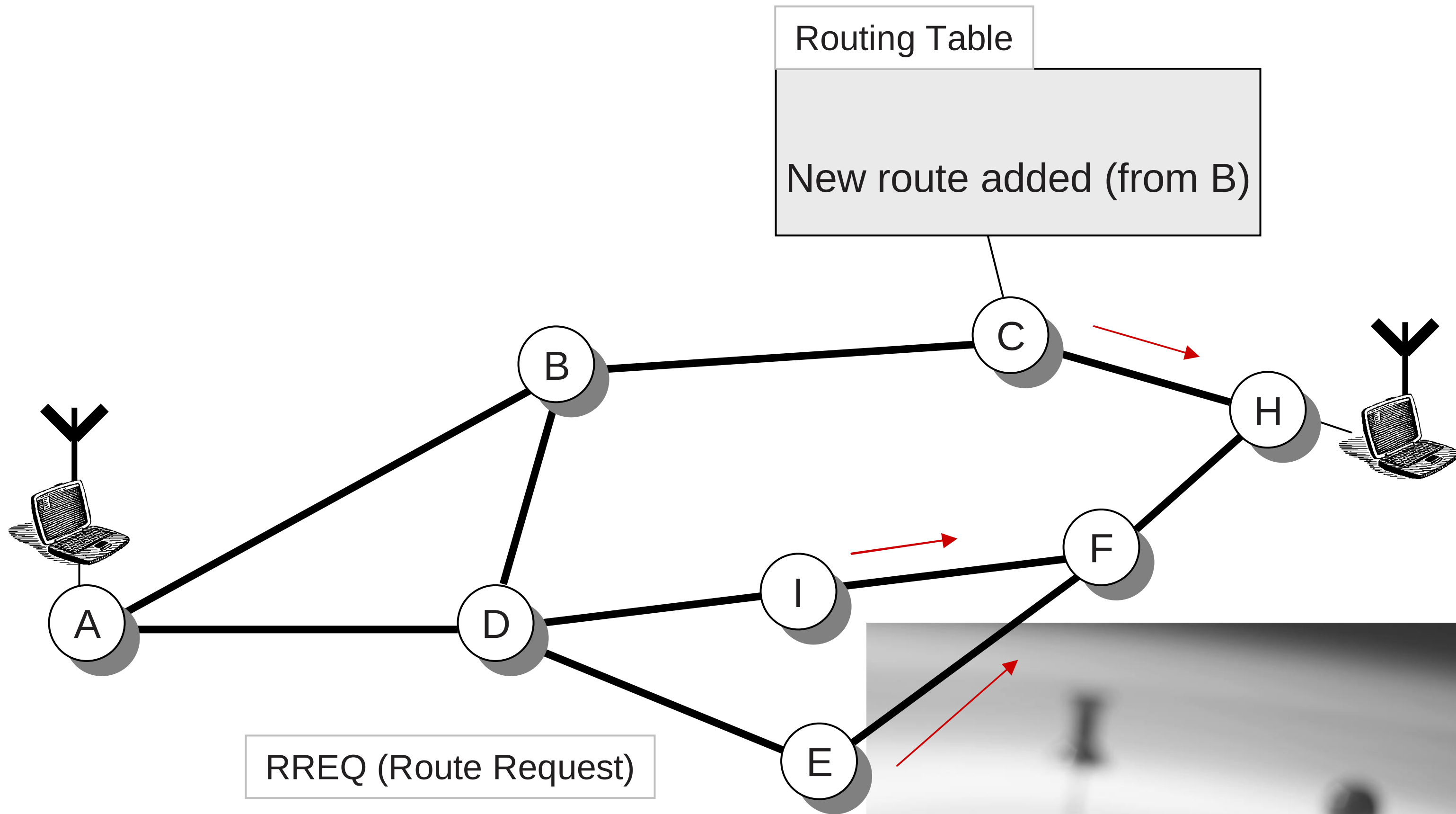


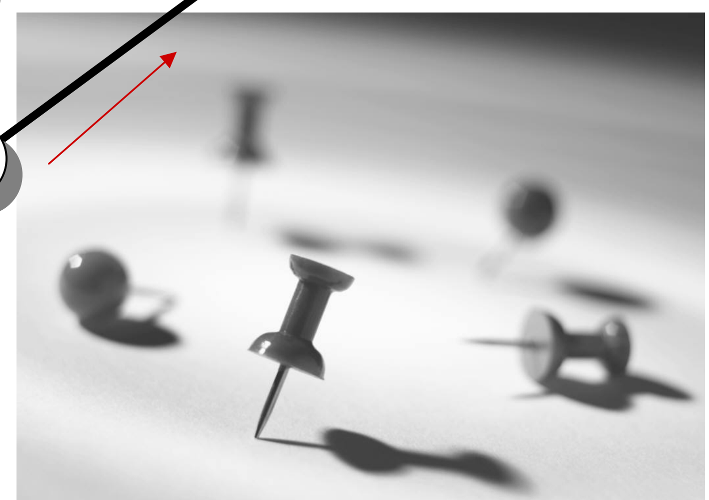
Ad-hoc On-demand Distance Vector (AODV)



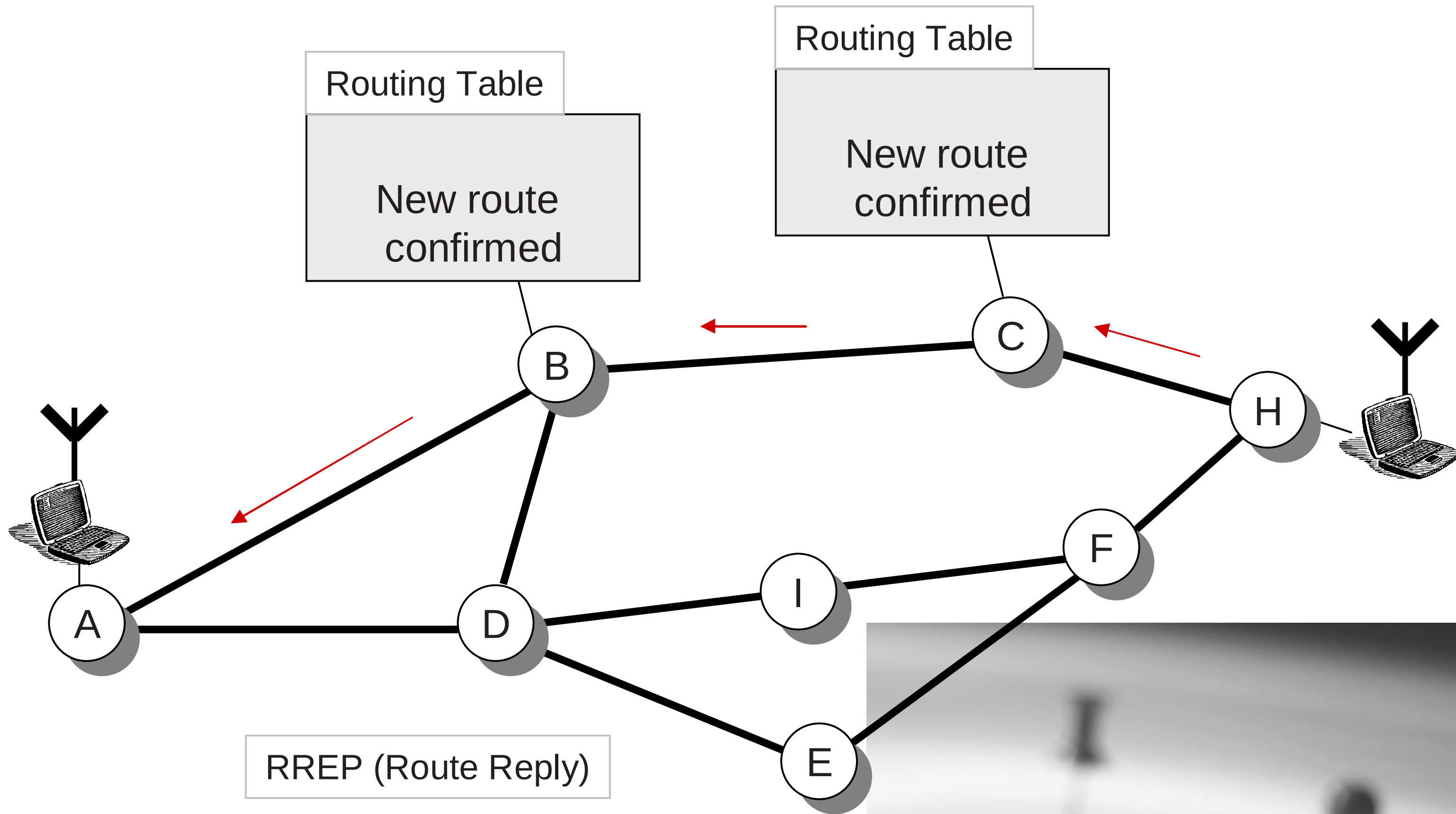


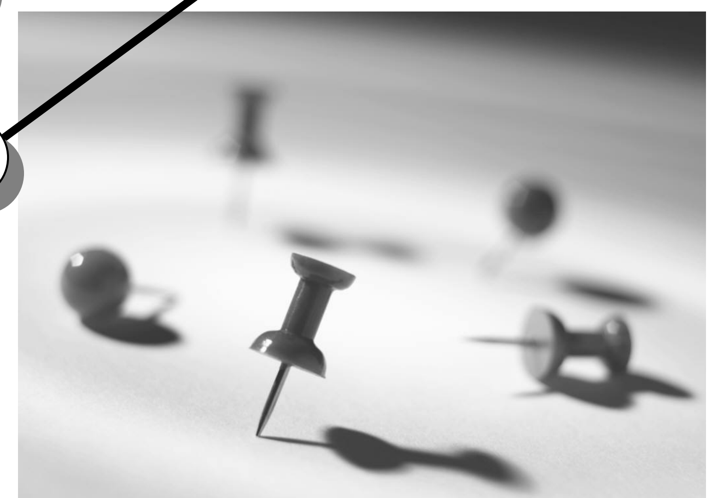
AODV - Route Request


Routing Table
New route added (from B)
A
B
C
D
E
F
H
I
RREQ (Route Request)

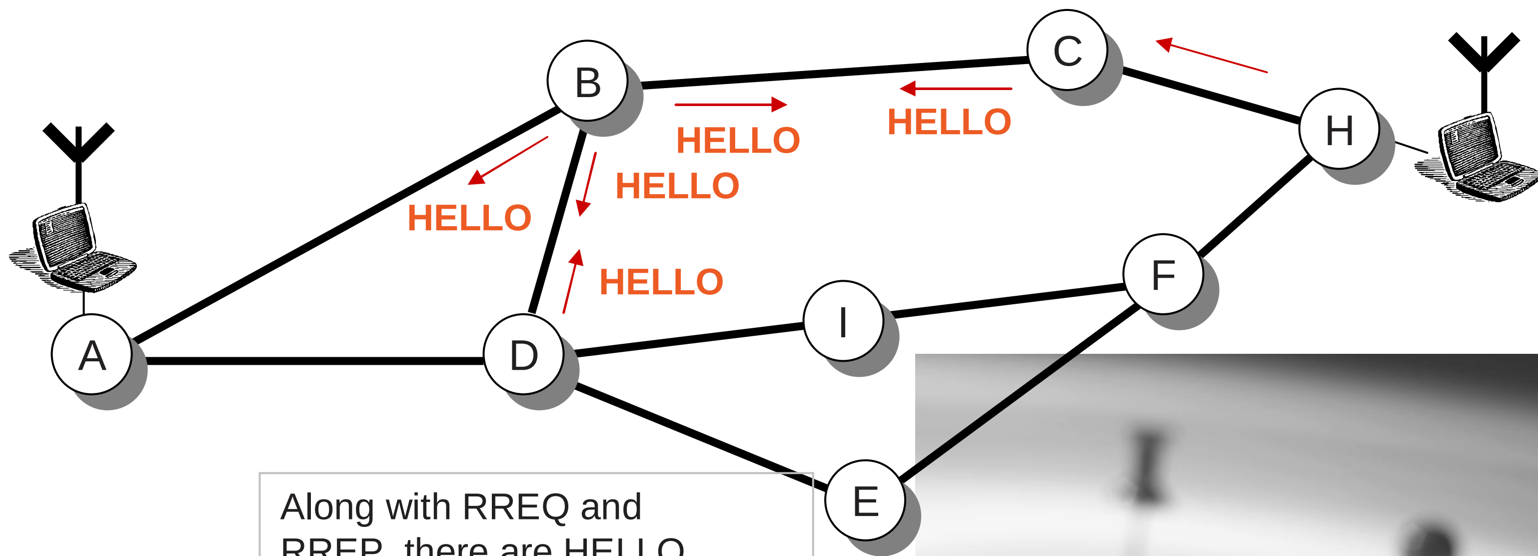


AODV

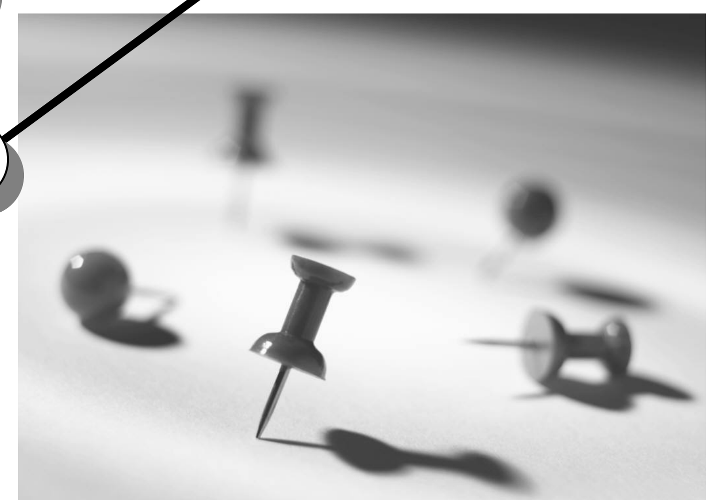




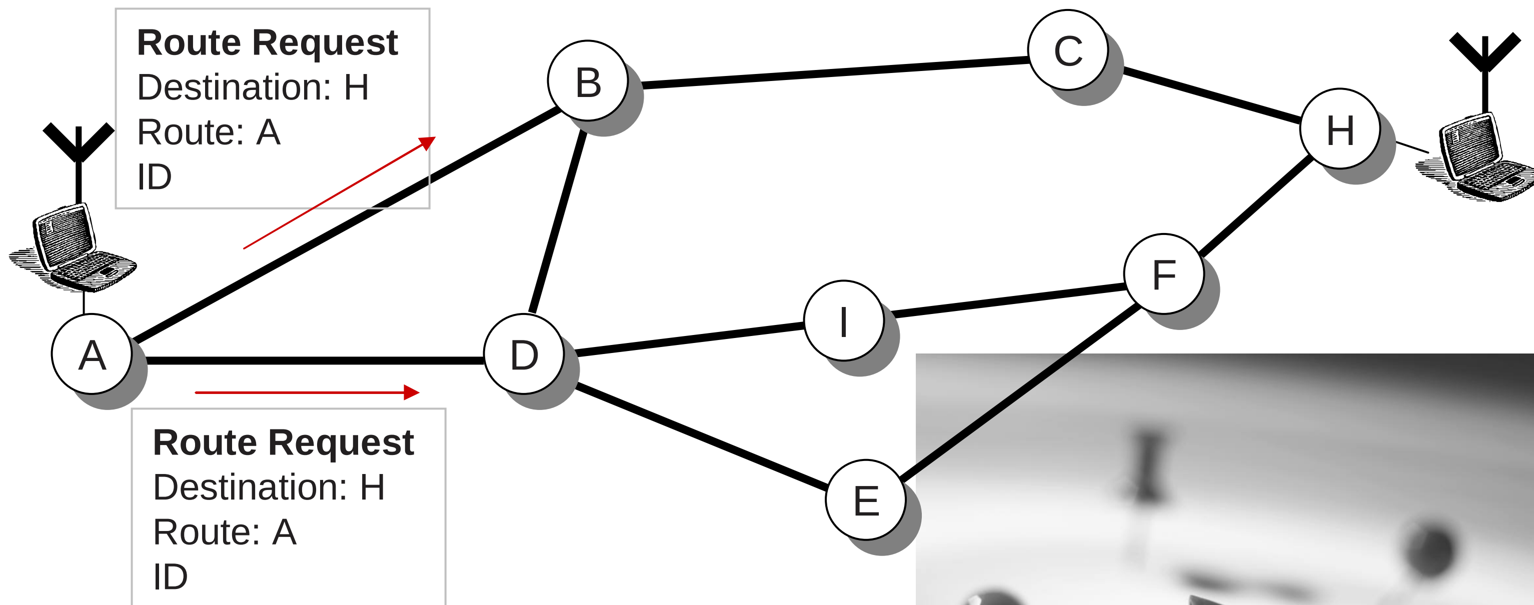
AODV - Route Reply

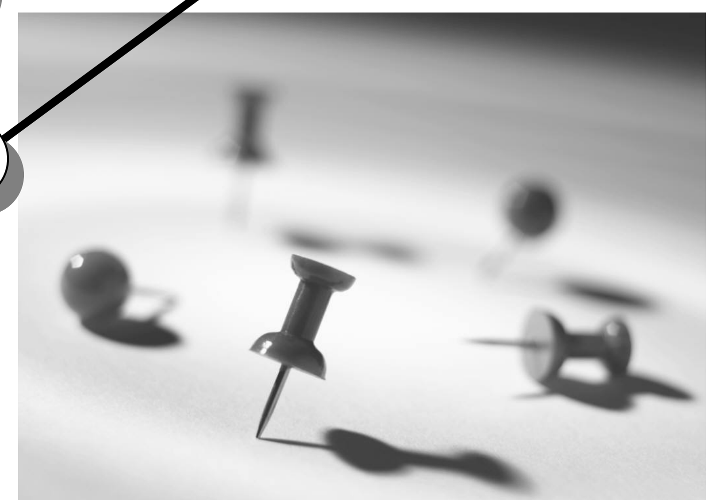


Along with RREQ and RREP, there are HELLO broadcasts which inform nodes as to their neighbours



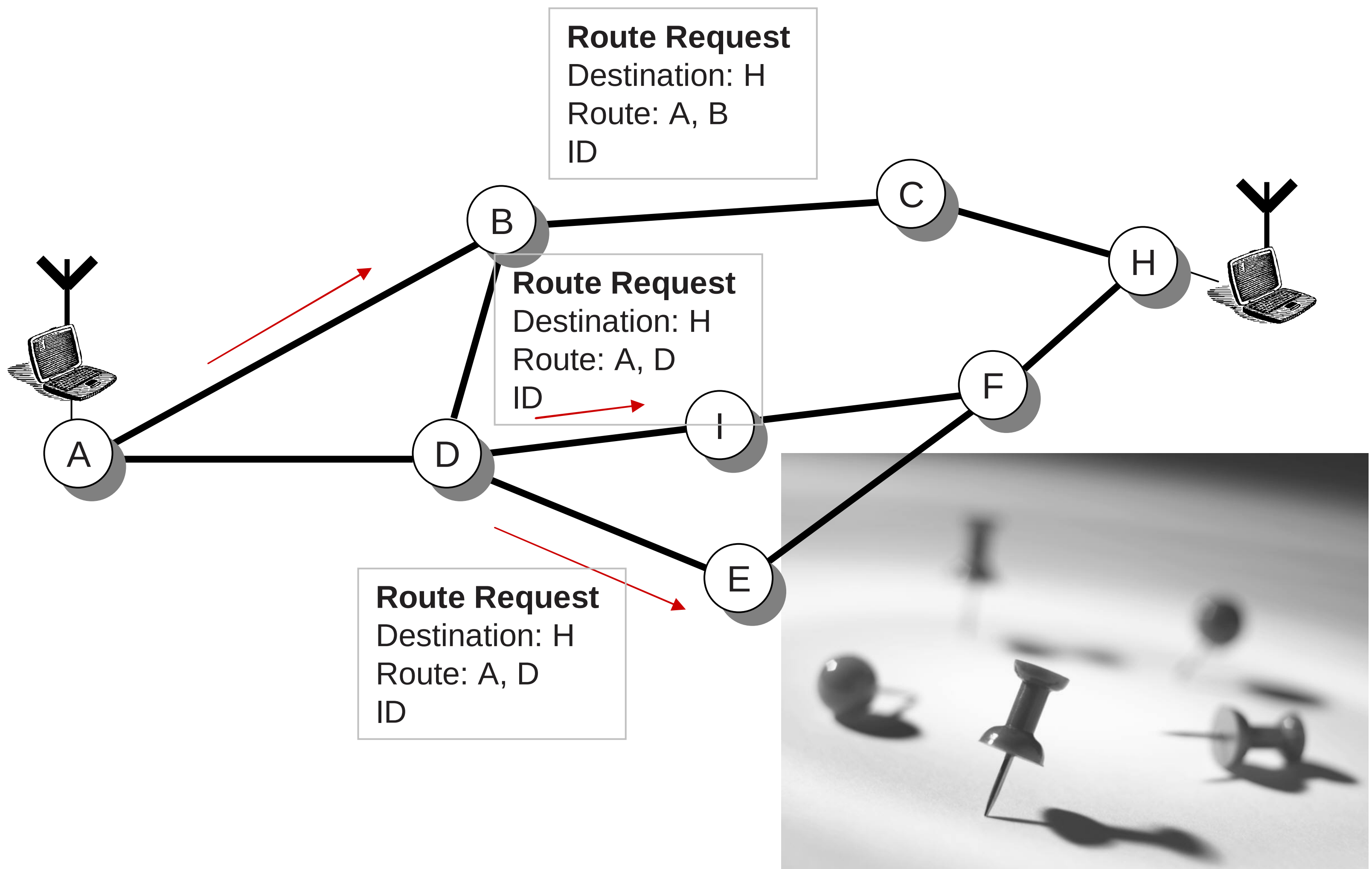
AODV - HELLO broadcasts

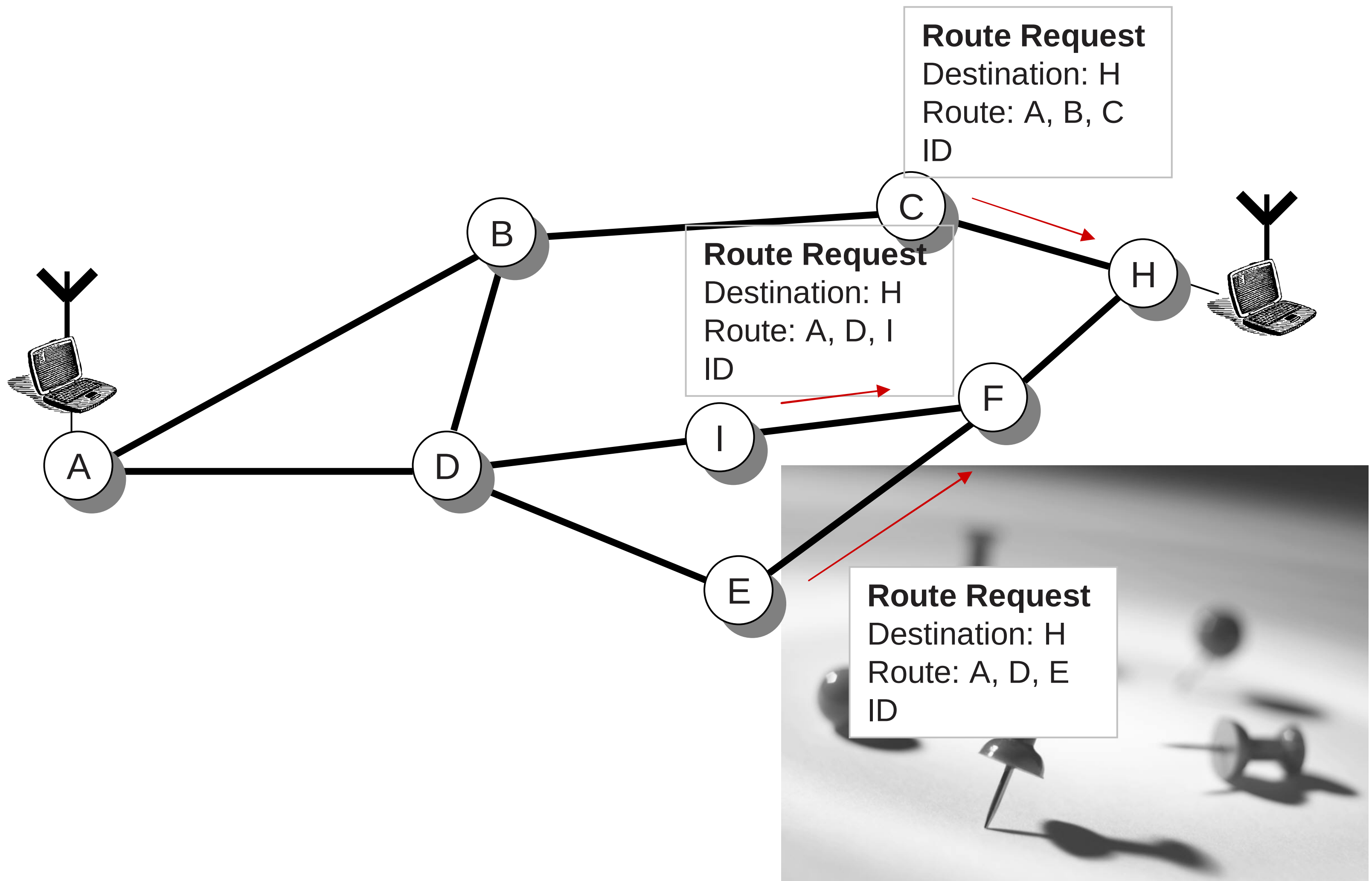




Dynamic Source Route - Route Discovery

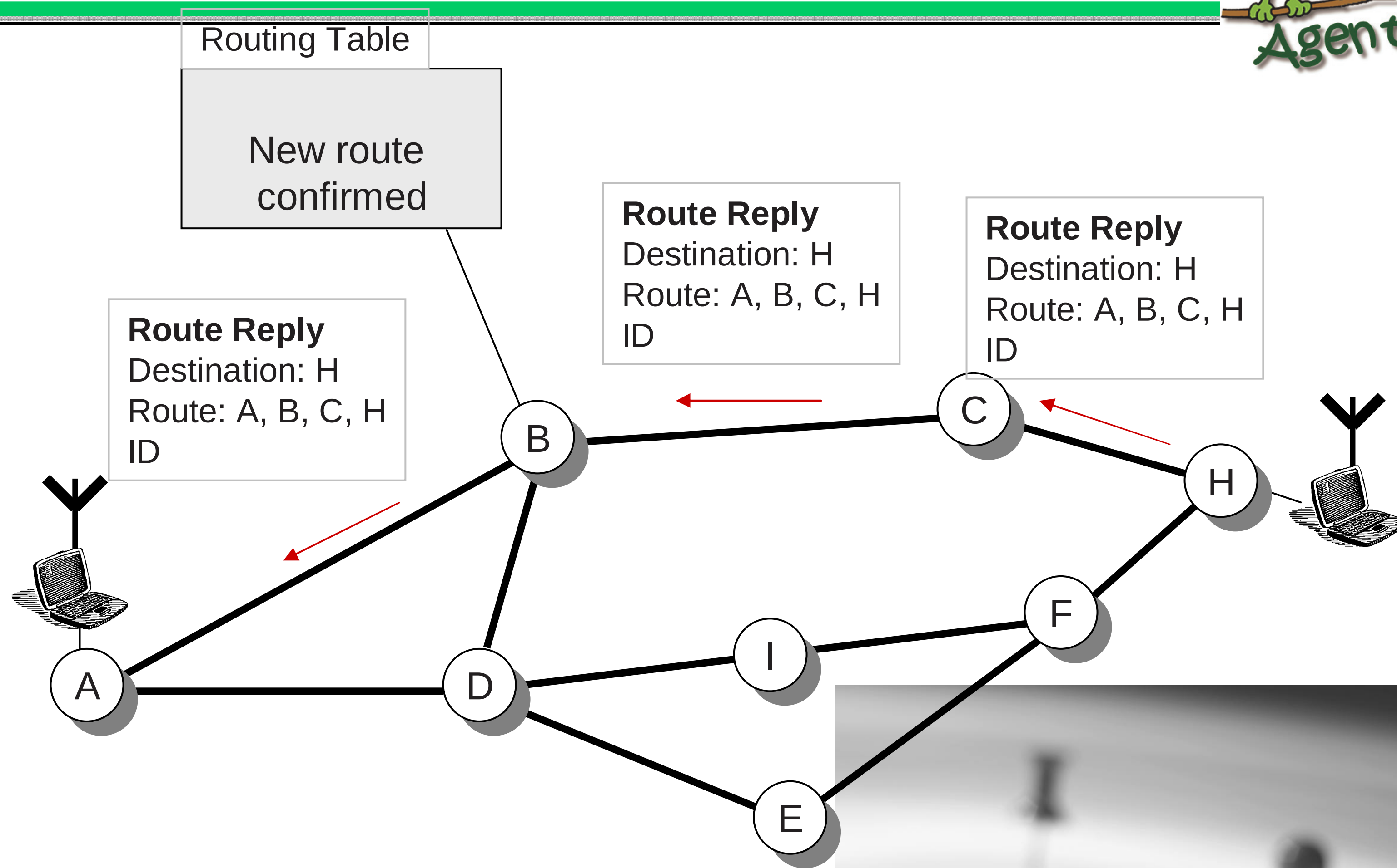
DSR - Route Discovery

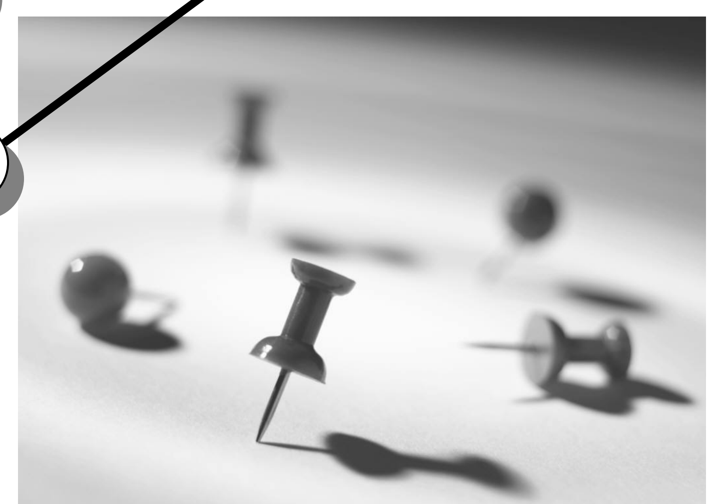




DSR - Route Discovery

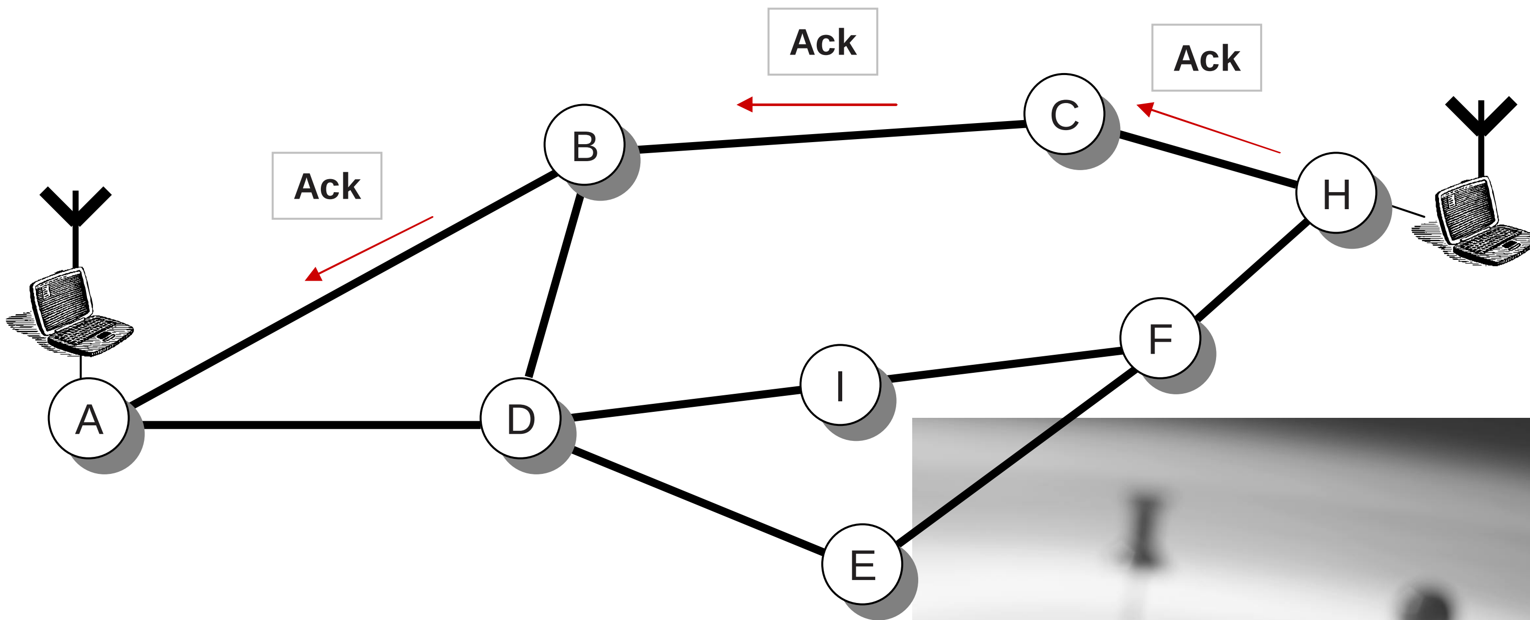
DSR - Route Reply

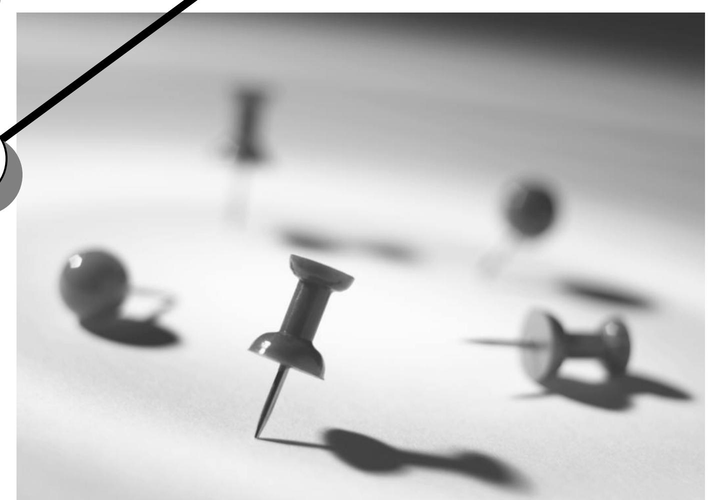




DSR - Keeping the route alive

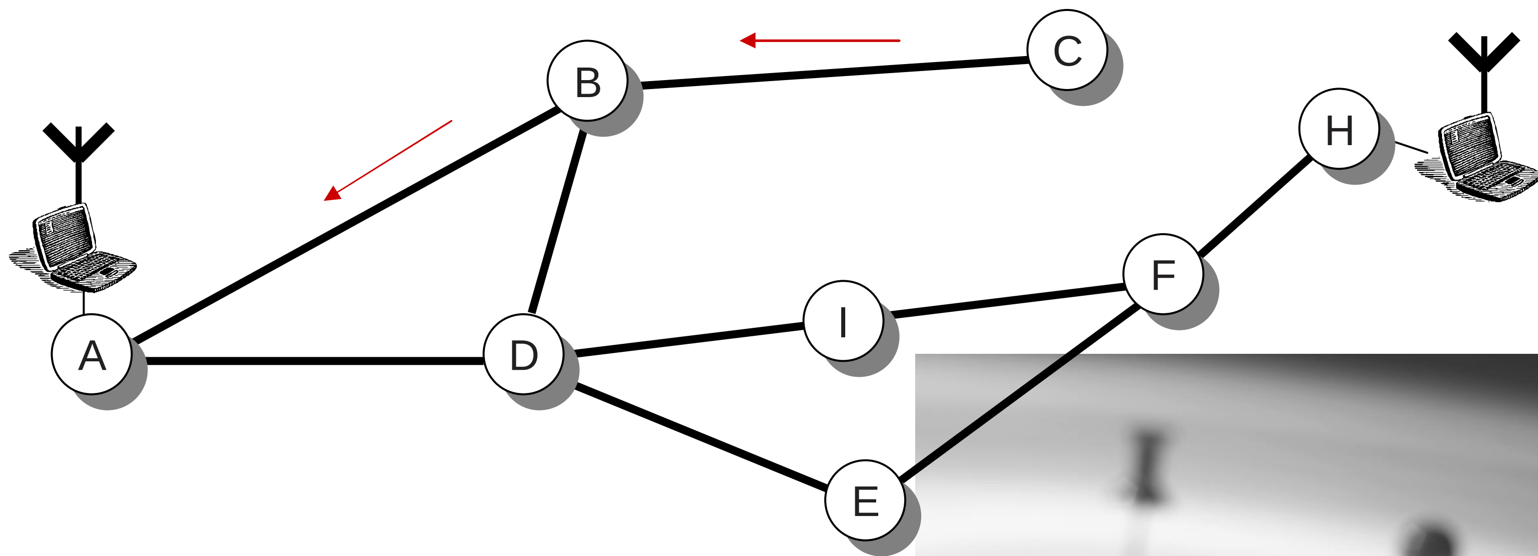
The route is kept alive with acknowledgements of data received

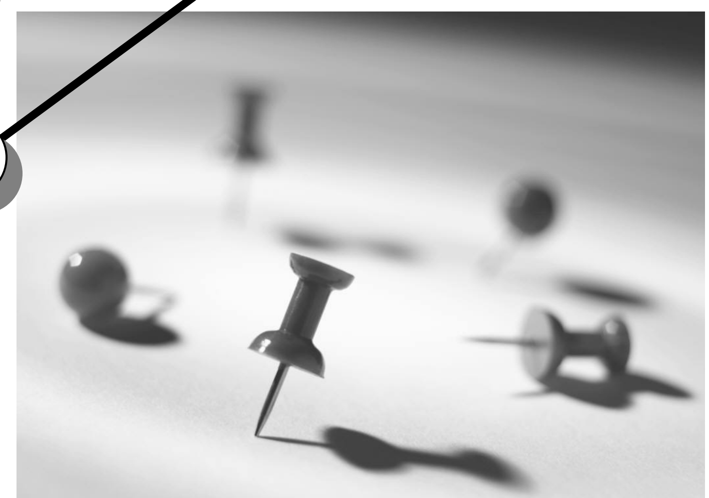




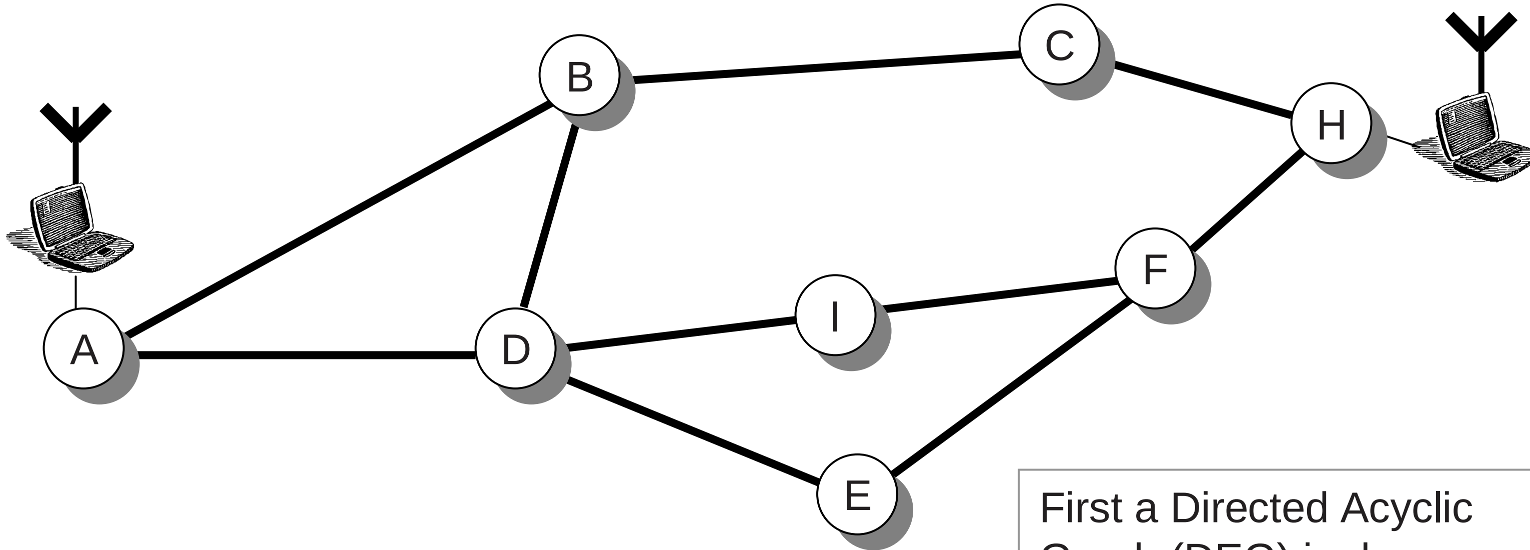
Nodes will rip-up the route on receiving the error packets

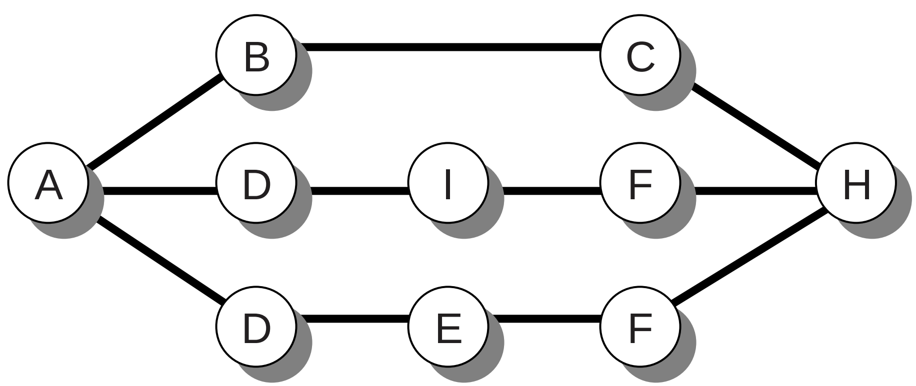
Error packet
Destination not available





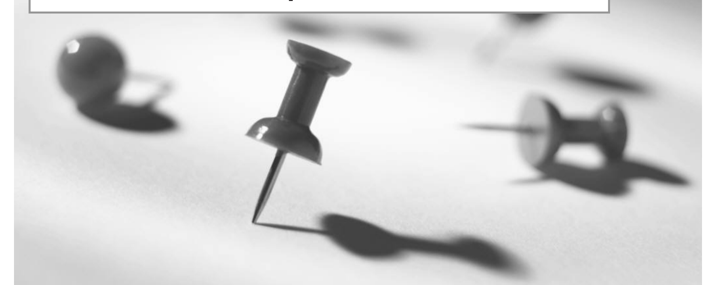
DSR - Route Reply





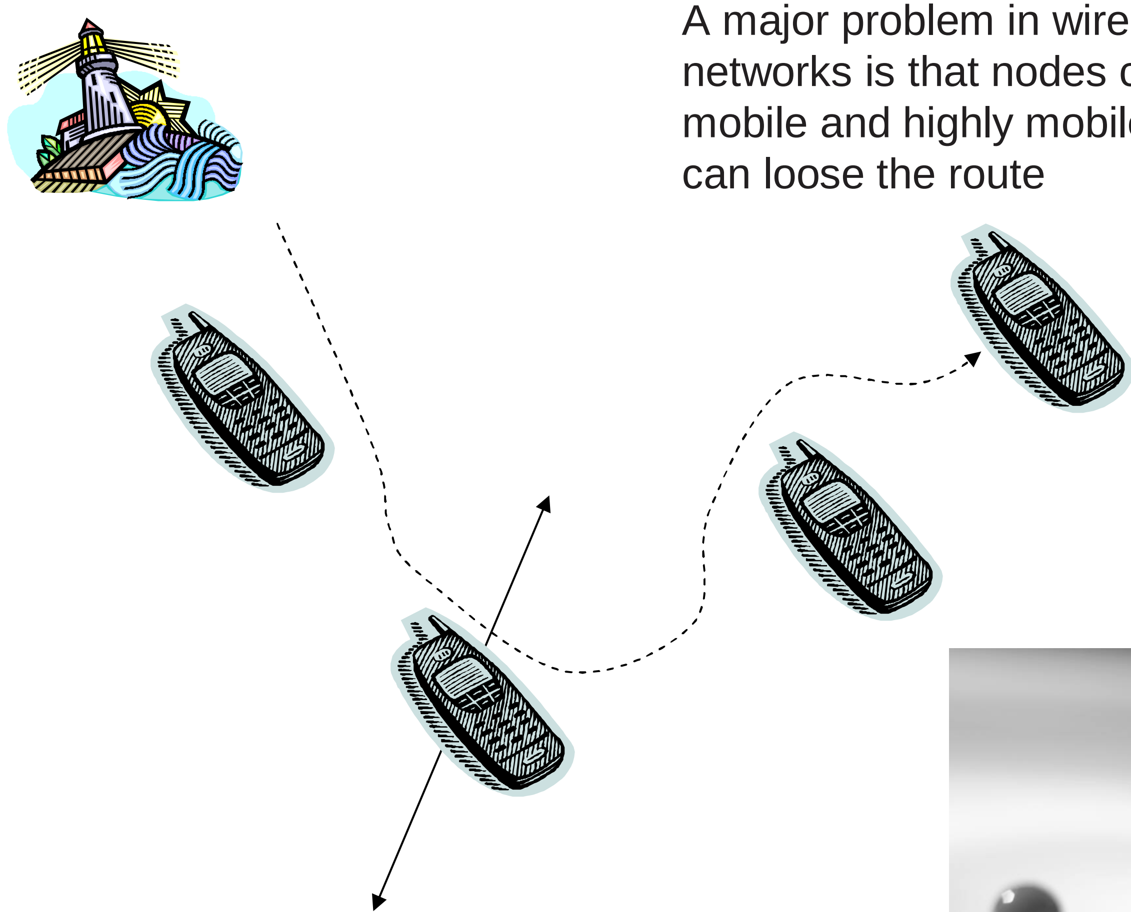
First a Directed Acyclic Graph (DEG) is drawn

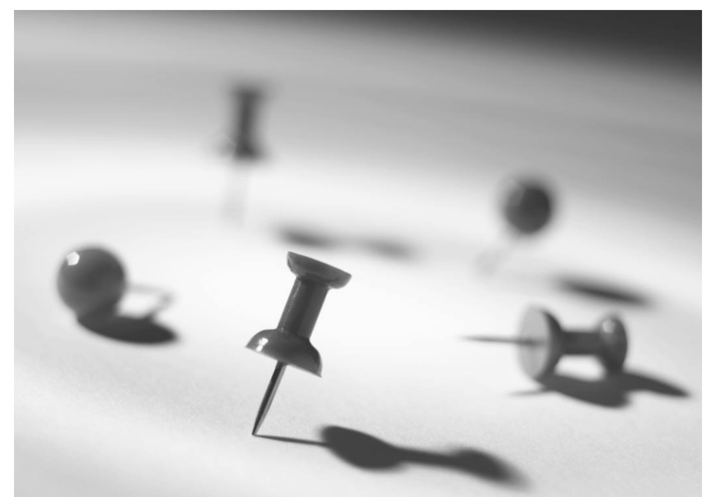
A DEG contains no cycles, thus it is not possible to start from one position and end at the same place



Temporally Ordered Routing Algorithm

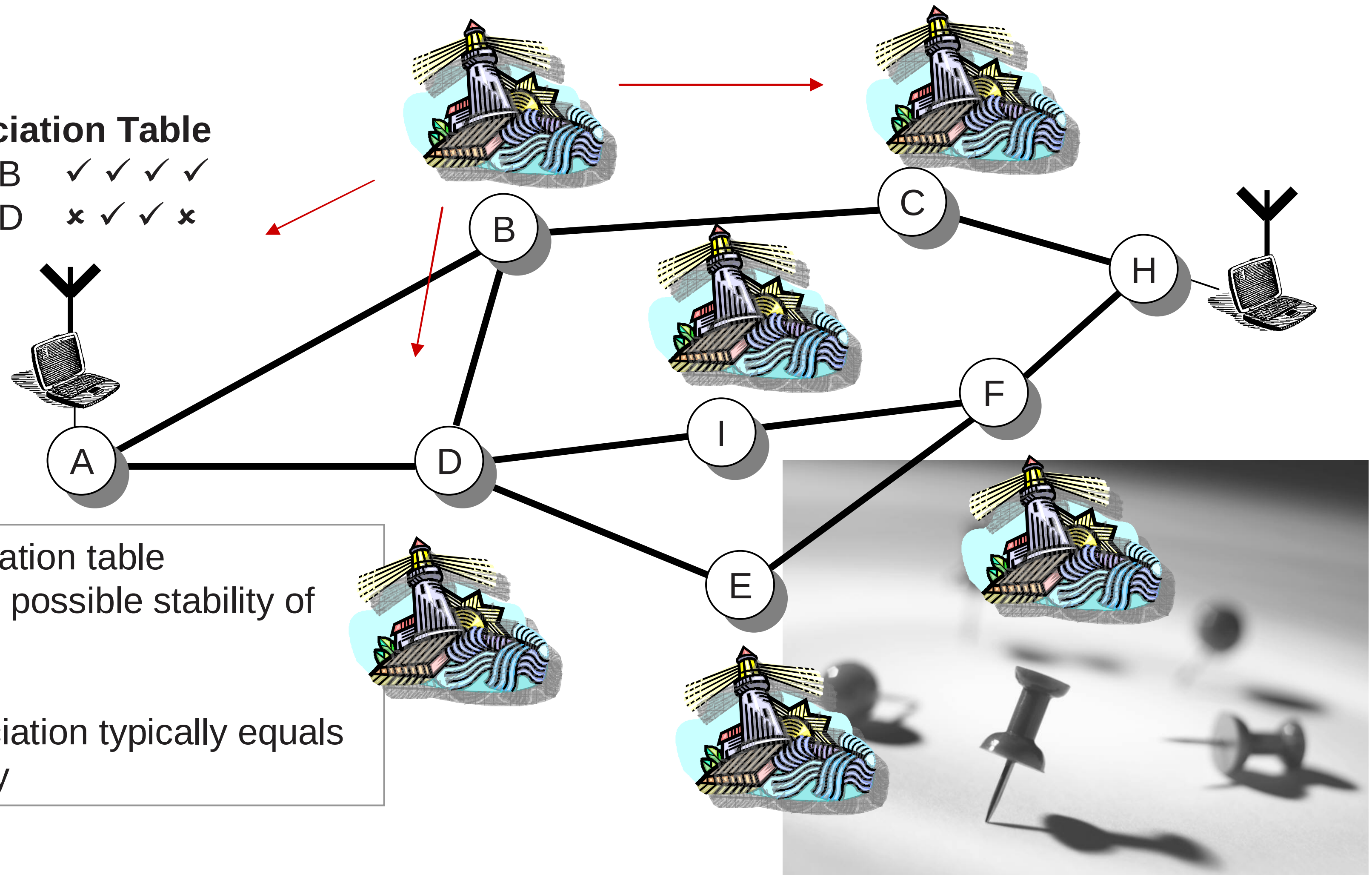
A major problem in wireless networks is that nodes can be mobile and highly mobile nodes can loose the route





Association Table

Node B	✓	✓	✓	✓
Node D	×	✓	✓	×

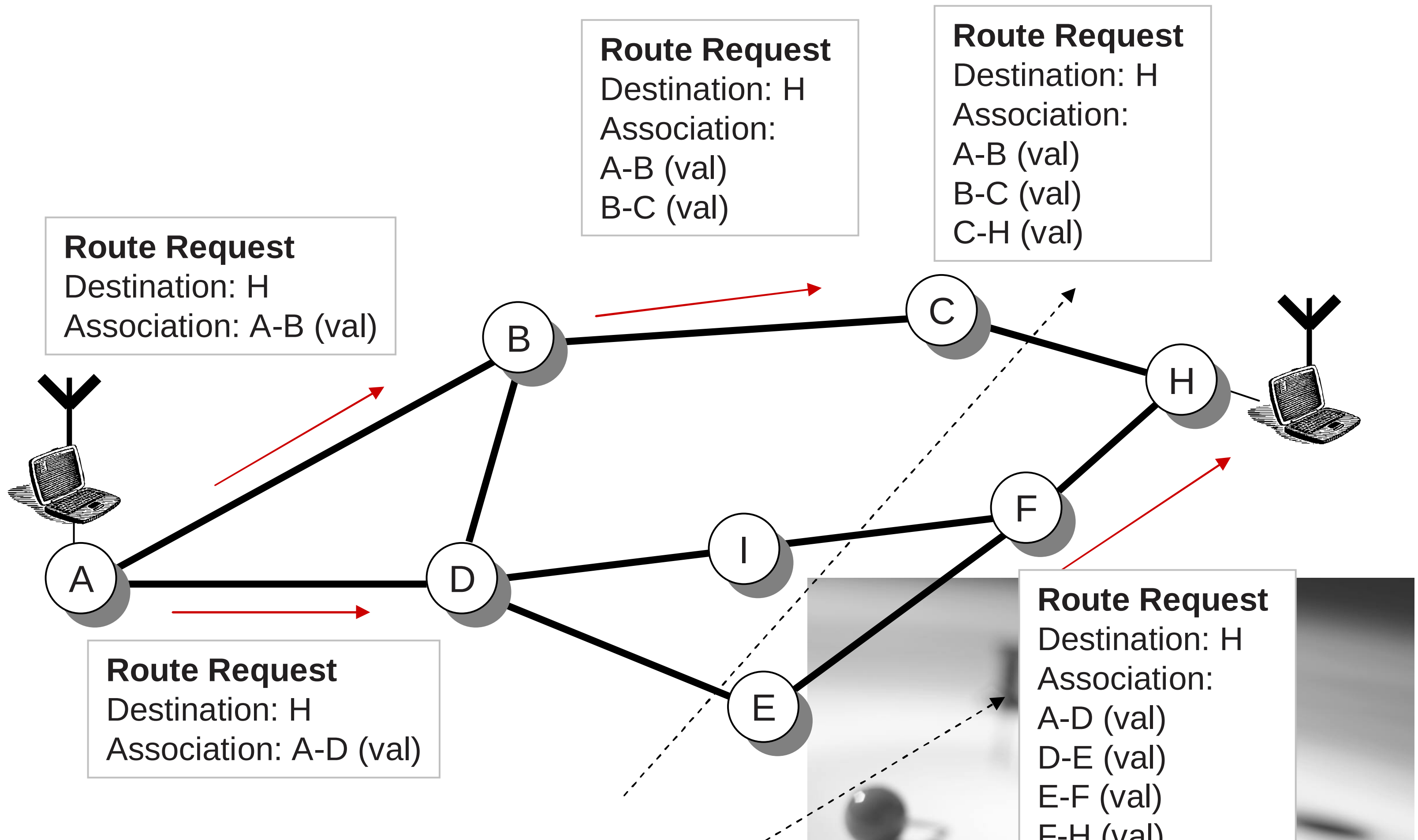


The association table defines the possible stability of the route.

High association typically equals low mobility

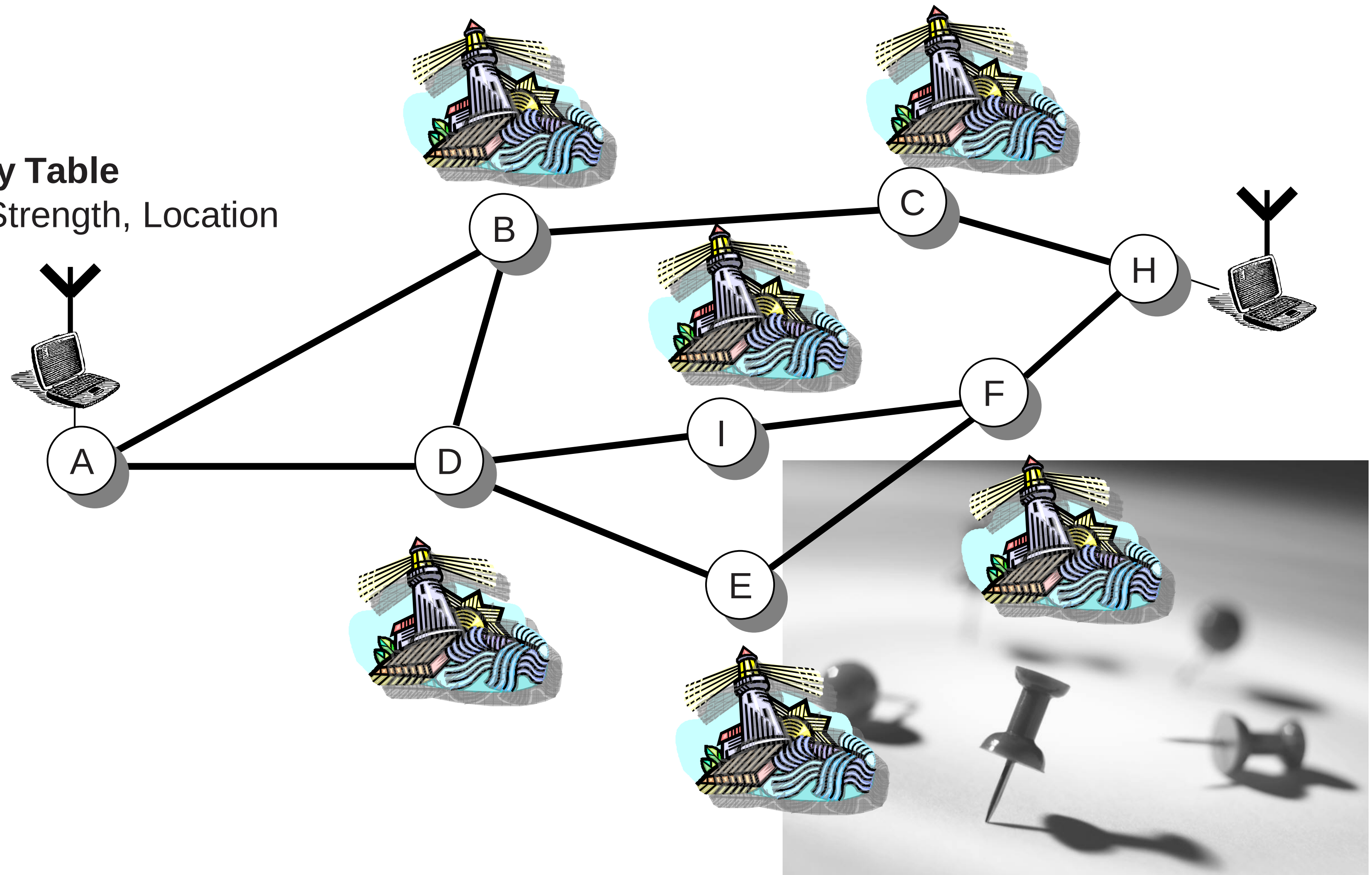
Associated-Based Routing (ABR)

ABR



The destination can choice which route is the most stable. If they have the same associativity then a hop count is used.

Stability Table
Signal Strength, Location



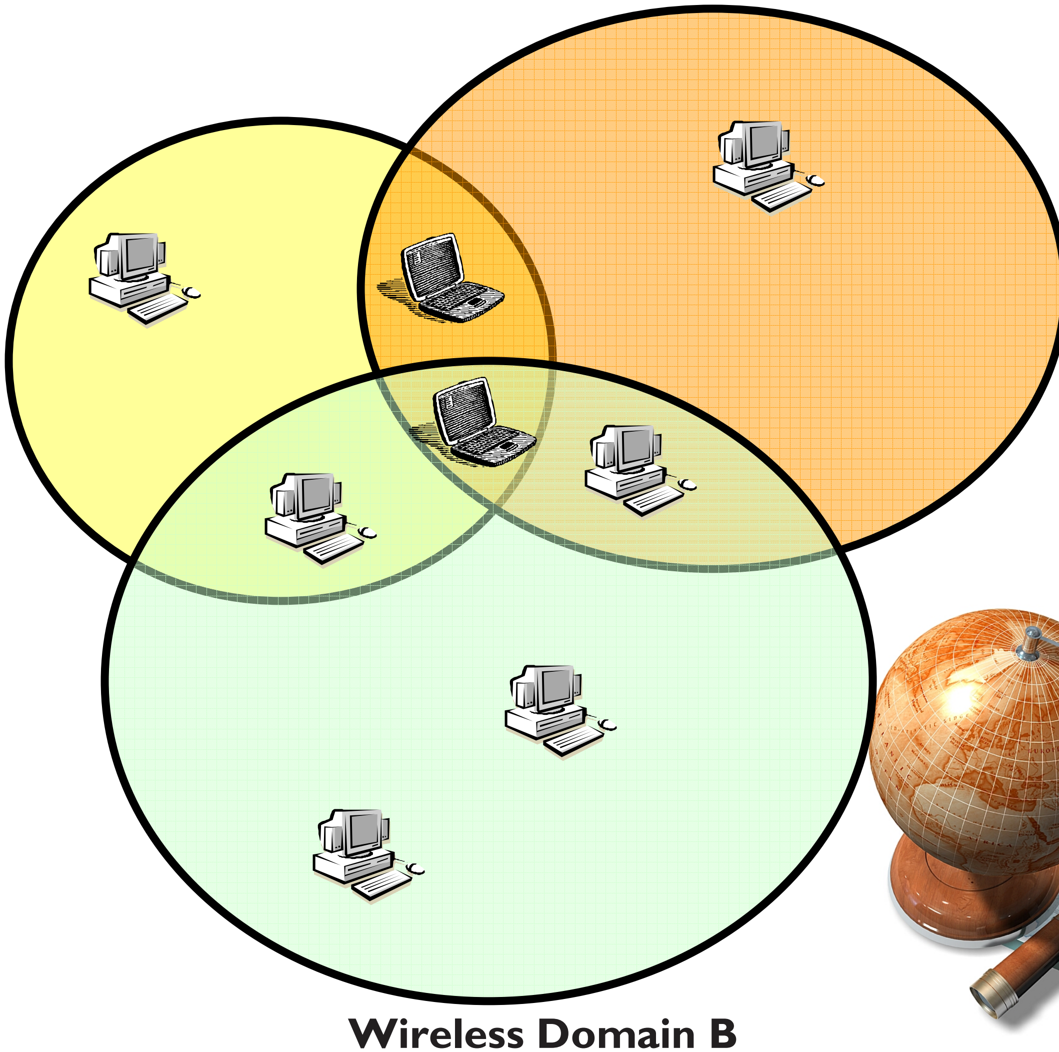
Signal Stability Routing

Agents for Ad-hoc Routing

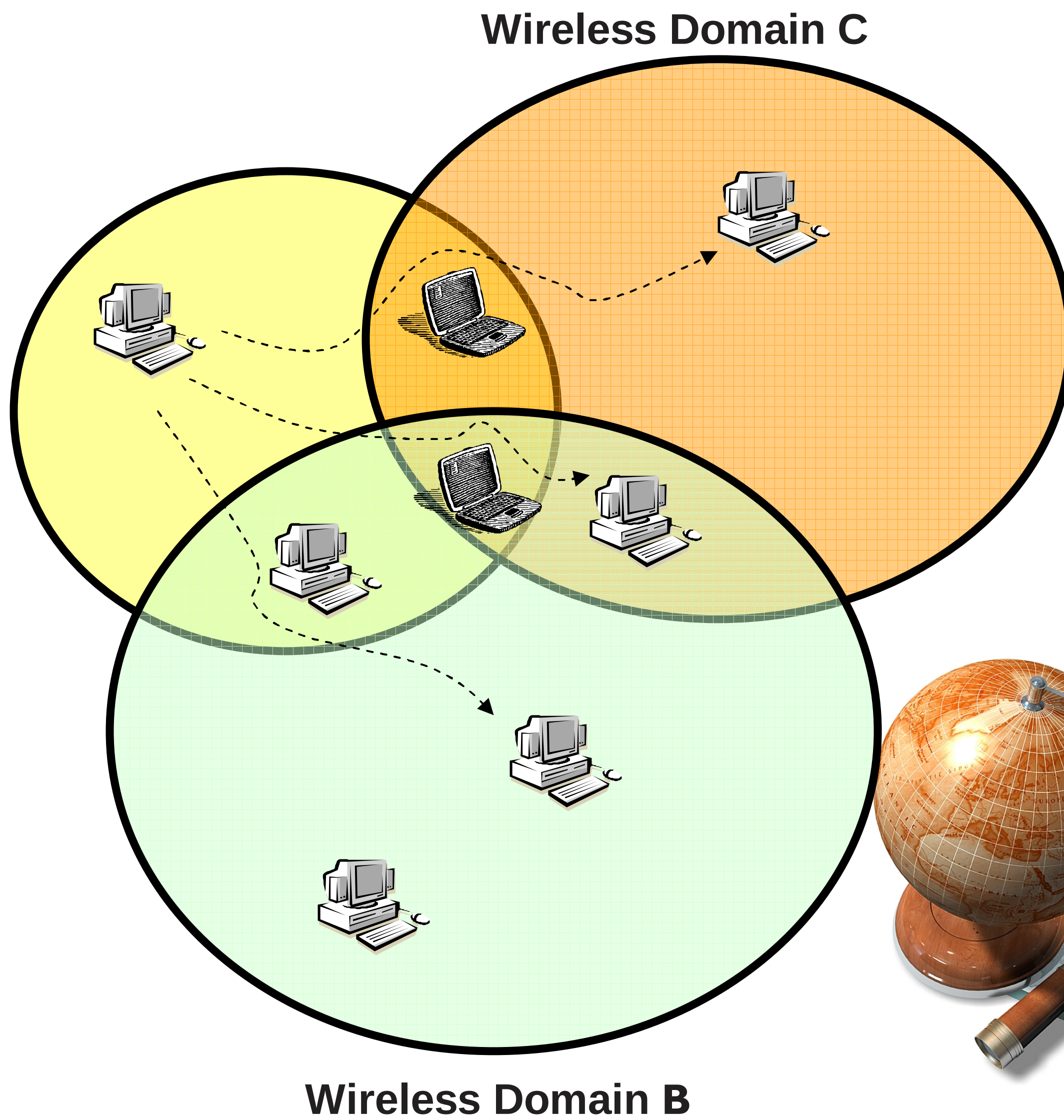
Bill Buchanan,
Nikos Migas,
Kevin McArtney

Wireless Domains

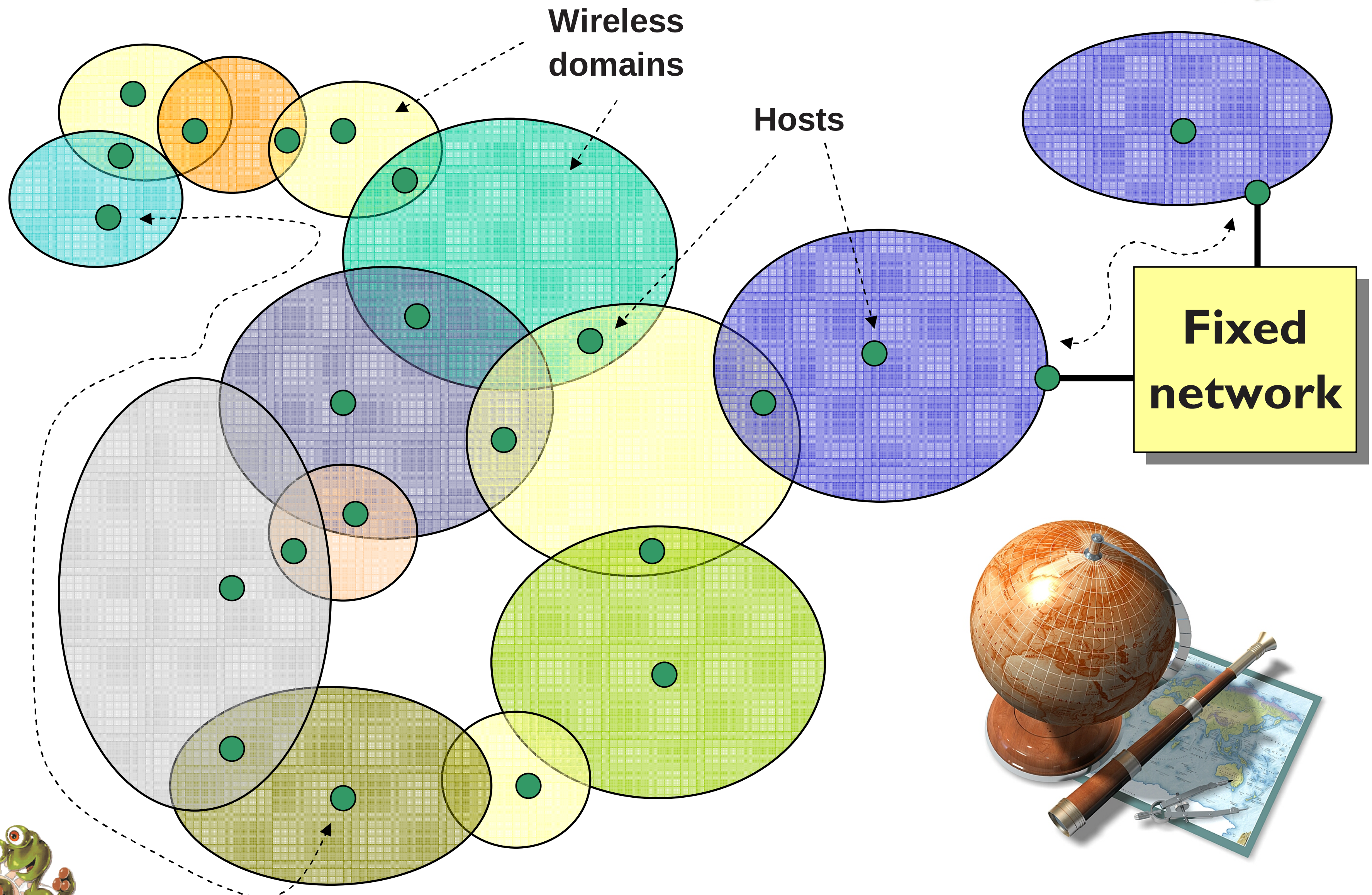
Wireless Domain A



Wireless Domain C



Routing Over Wireless Domains



Bridging Wireless Domains

MARIAN – Mobile Agents In Ad-hoc Networks

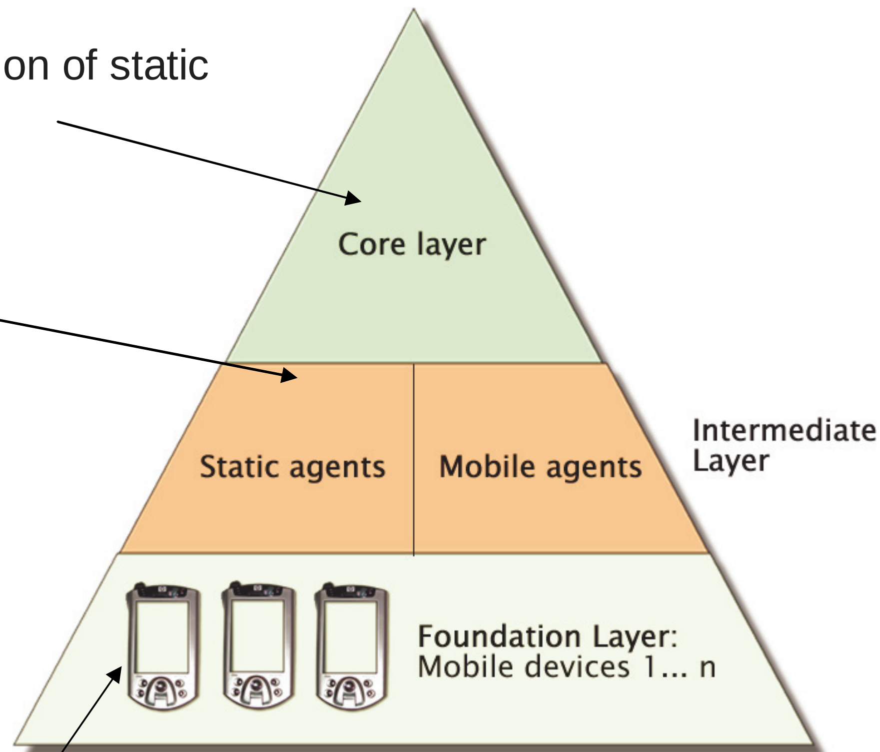
Core layer

–Hybrid approach. Combination of static and mobile agent approach

Intermediate layer

–Static agent approach. Centralized routing process

–Mobile agent approach. Decentralized routing process



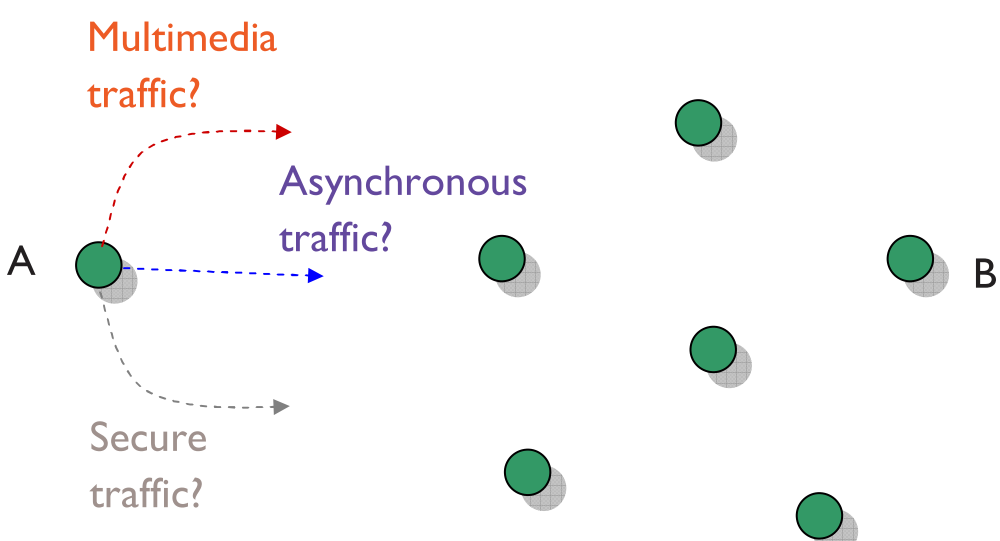
Foundation layer

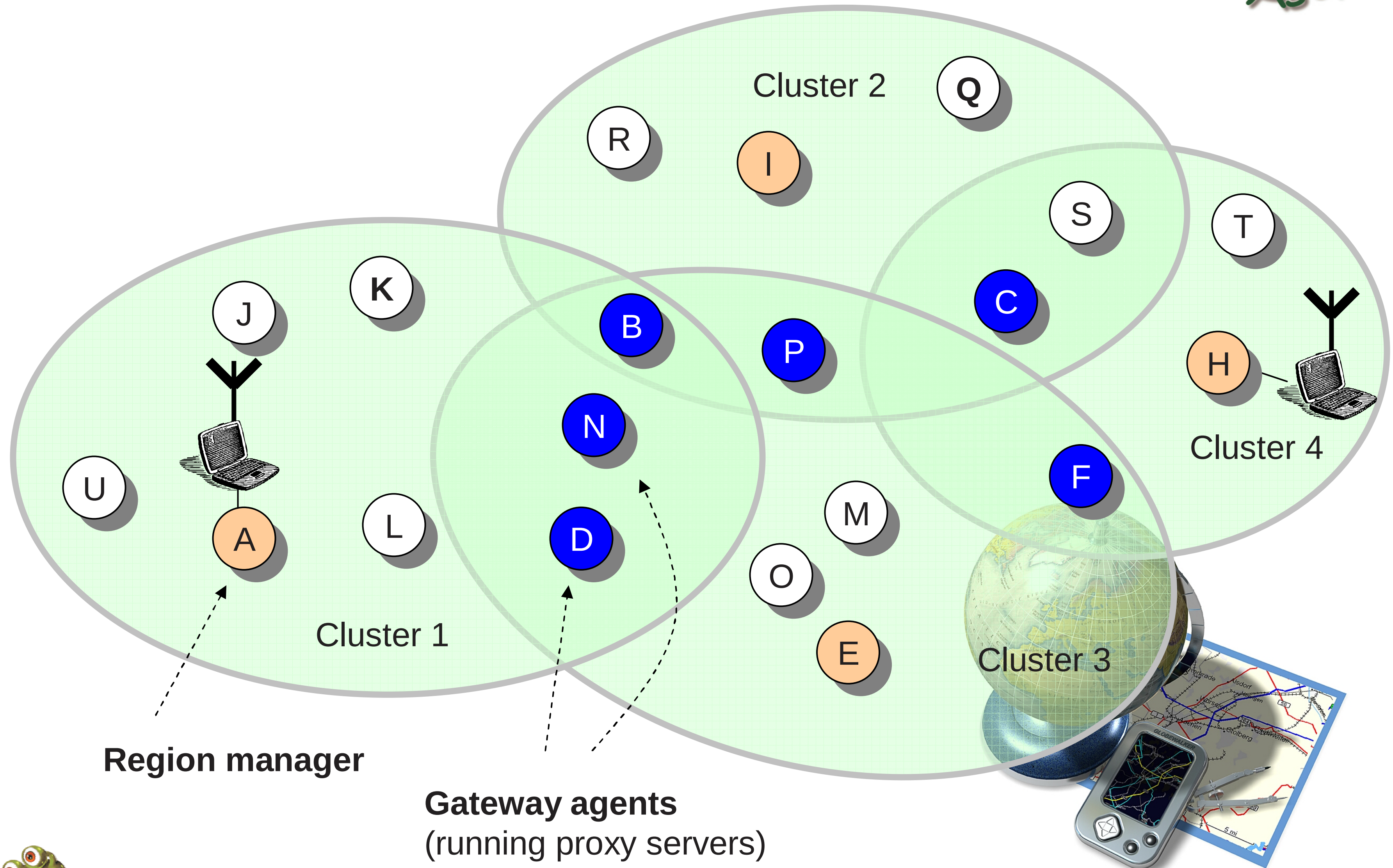
–The physical layer consisting of all mobile nodes

Overall Framework

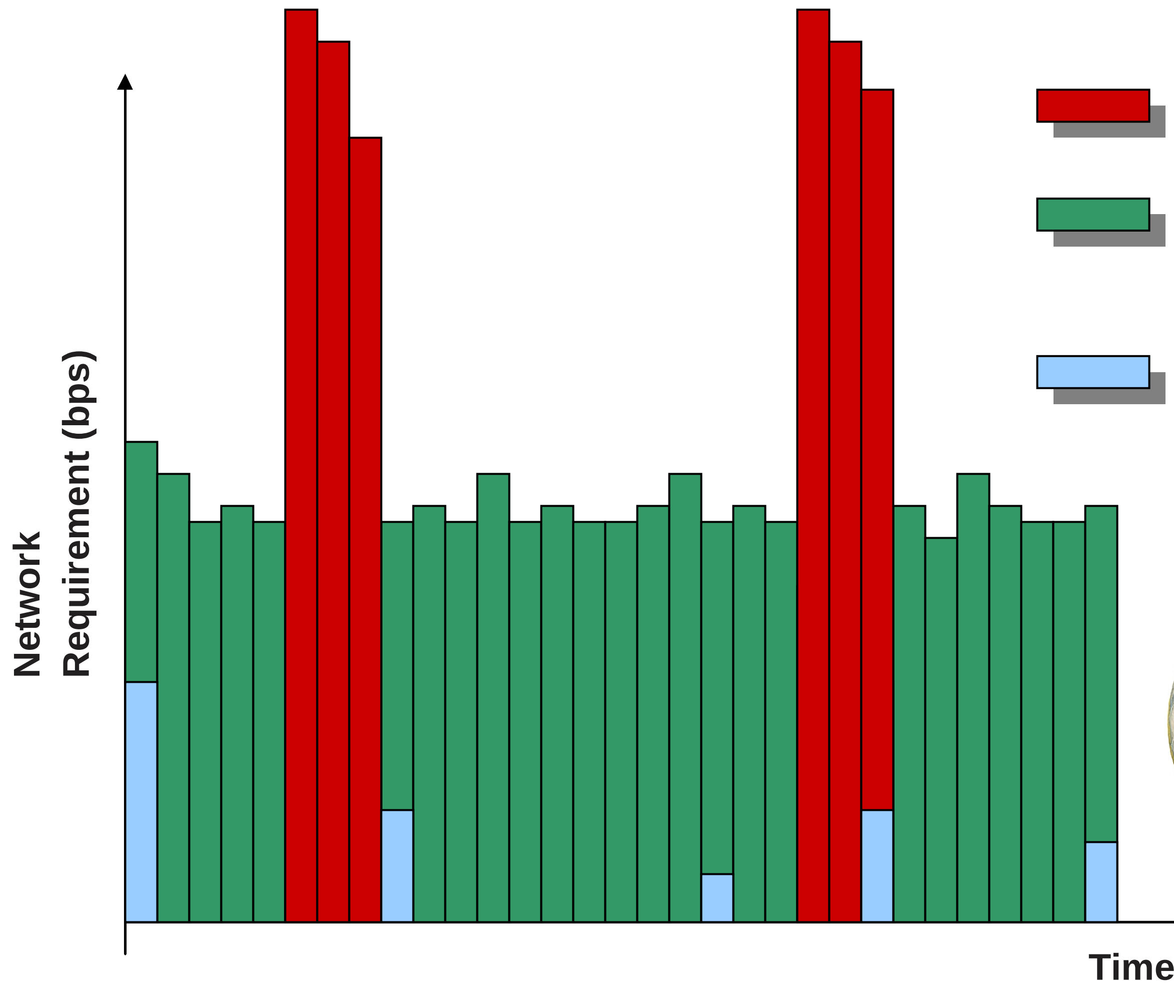
Routing in Ad-hoc networks

- Routing is the most important function in ad-hoc networks, however a challenging issue:
  - **Mobility**
    - e.g. A route that is considered as good now, may become unavailable or not optimal some time later
  - **Routing requires a lot of processing power**
    - Mobile nodes running on batteries may not be able to cope

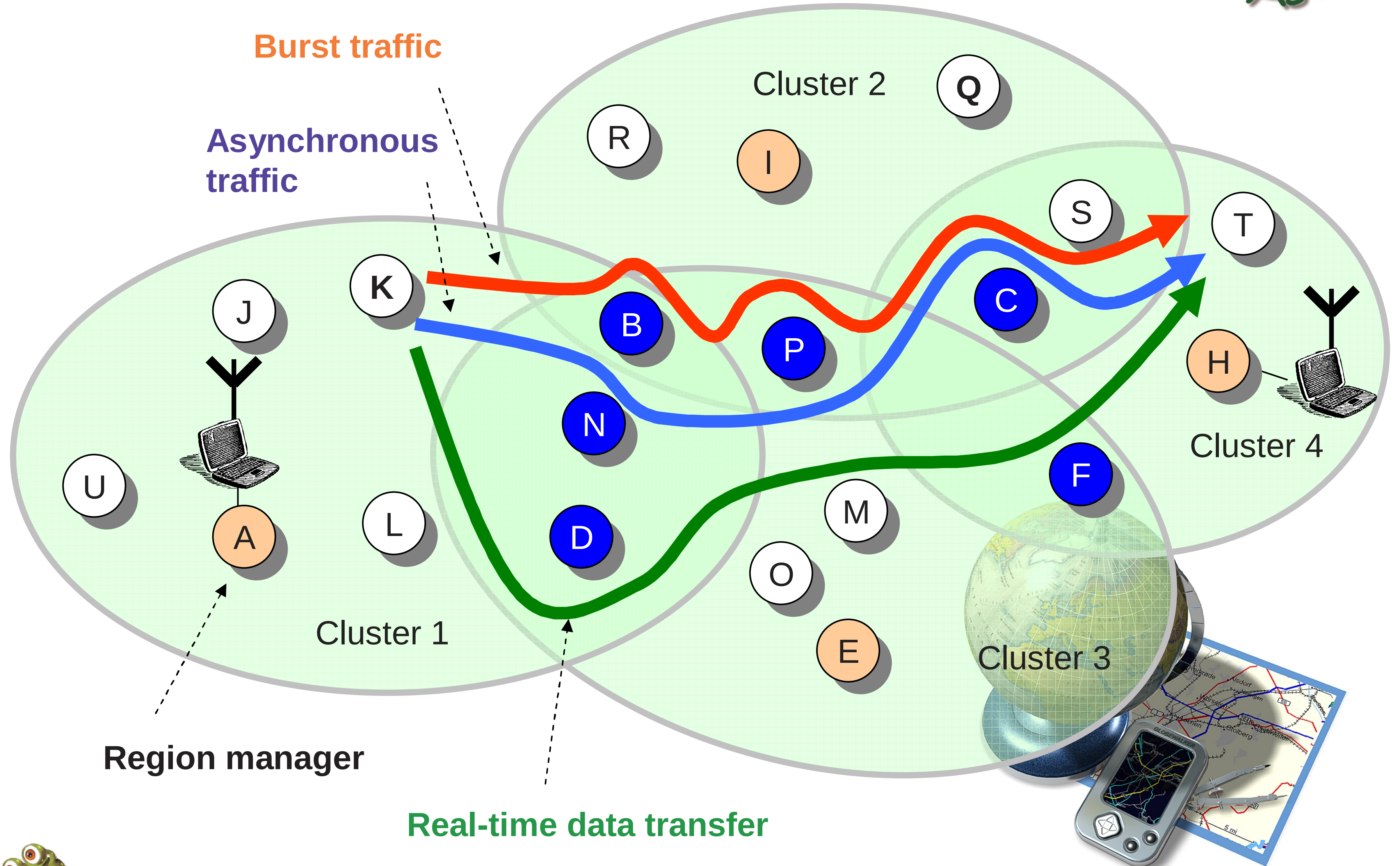



Cluster 2
Q
R
I
S
T
K
J
B
P
C
H
N
U
A
L
D
M
F
Cluster 4
O
E
Cluster 1
Cluster 3
Region manager
Gateway agents
(running proxy servers)

MARIAN



Routing for different traffics for QoS


Burst traffic
Asynchronous traffic
Cluster 2
Q
R
I
S
T
K
J
B
P
C
H
N
Cluster 4
U
A
L
D
M
F
O
E
Cluster 1
Cluster 3
Region manager
Real-time data transfer

Fitness of Route

Bill Buchanan

Core layer

–Hybrid approach. Combination of static and mobile agent approach

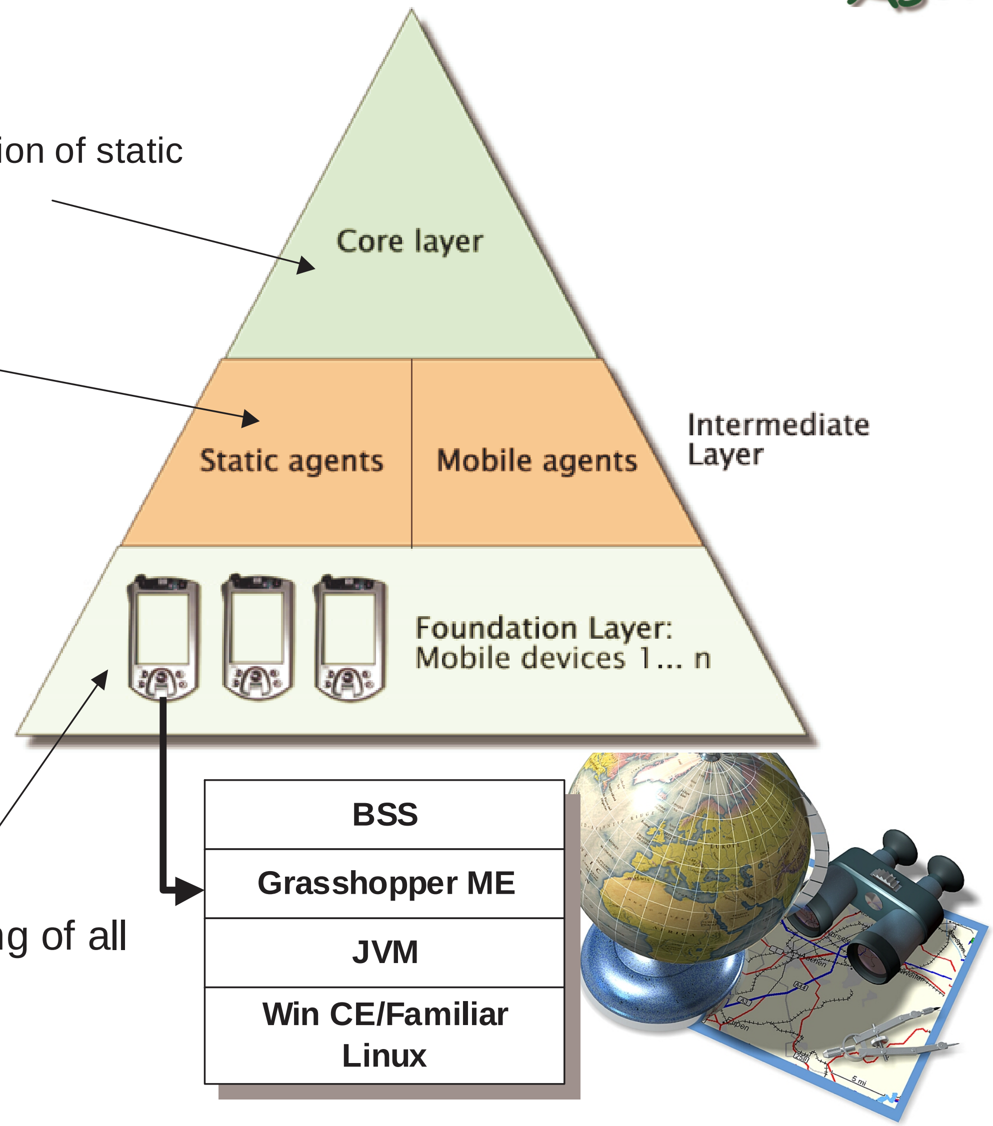
Intermediate layer

–Static agent approach. Centralized routing process

–Mobile agent approach. Decentralized routing process

Foundation layer

–The physical layer consisting of all mobile nodes



- Routing mechanism between wireless domains (Defining metrics for different traffic types).
- Performance tests on hosts, especially related to routing.
- Compatibility of devices (Window, UNIX, Windows CE, and so on).
- Generation of routing tables.
- Definition of QoS for connections.
- How agents are used (Static or Mobile?).

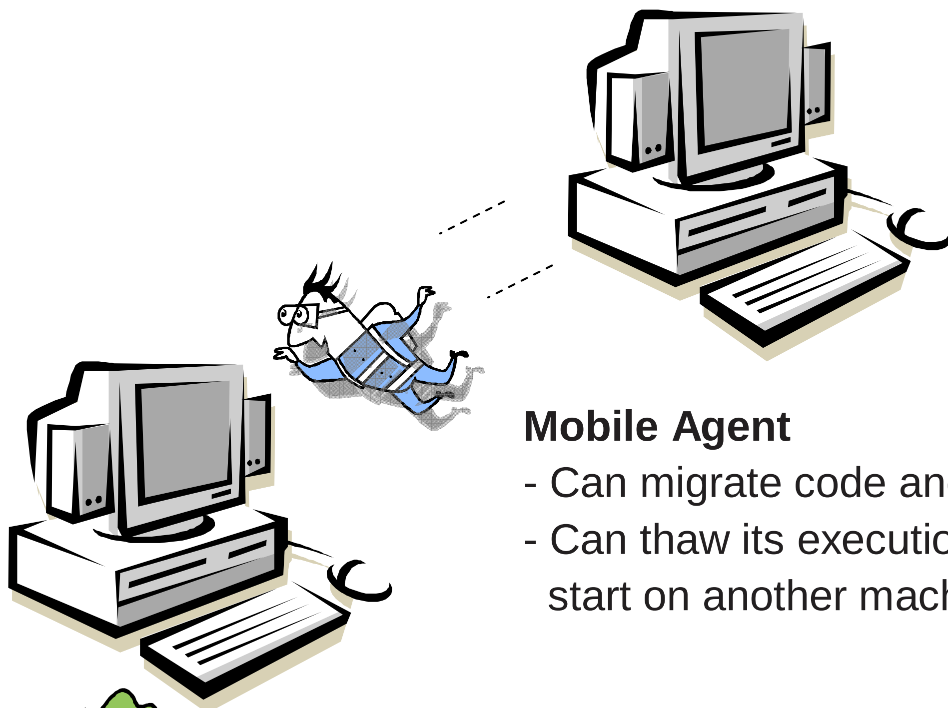
Tests:

- Network performance.
- Connectivity.
- Memory buffering.
- Processor performance.
- Protocol performance.
- And so on.

Issues?

Agents?

- Goal-oriented programs



Mobile Agent

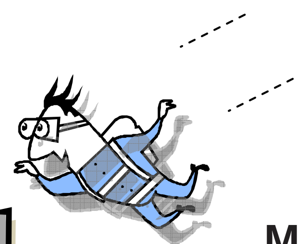
- Can migrate code and data between hosts
- Can thaw its execution, serialise itself, and the start on another machine

Static Agent

- Stays local to the machine
- Can have high integration with host

Mobile and Static Agents

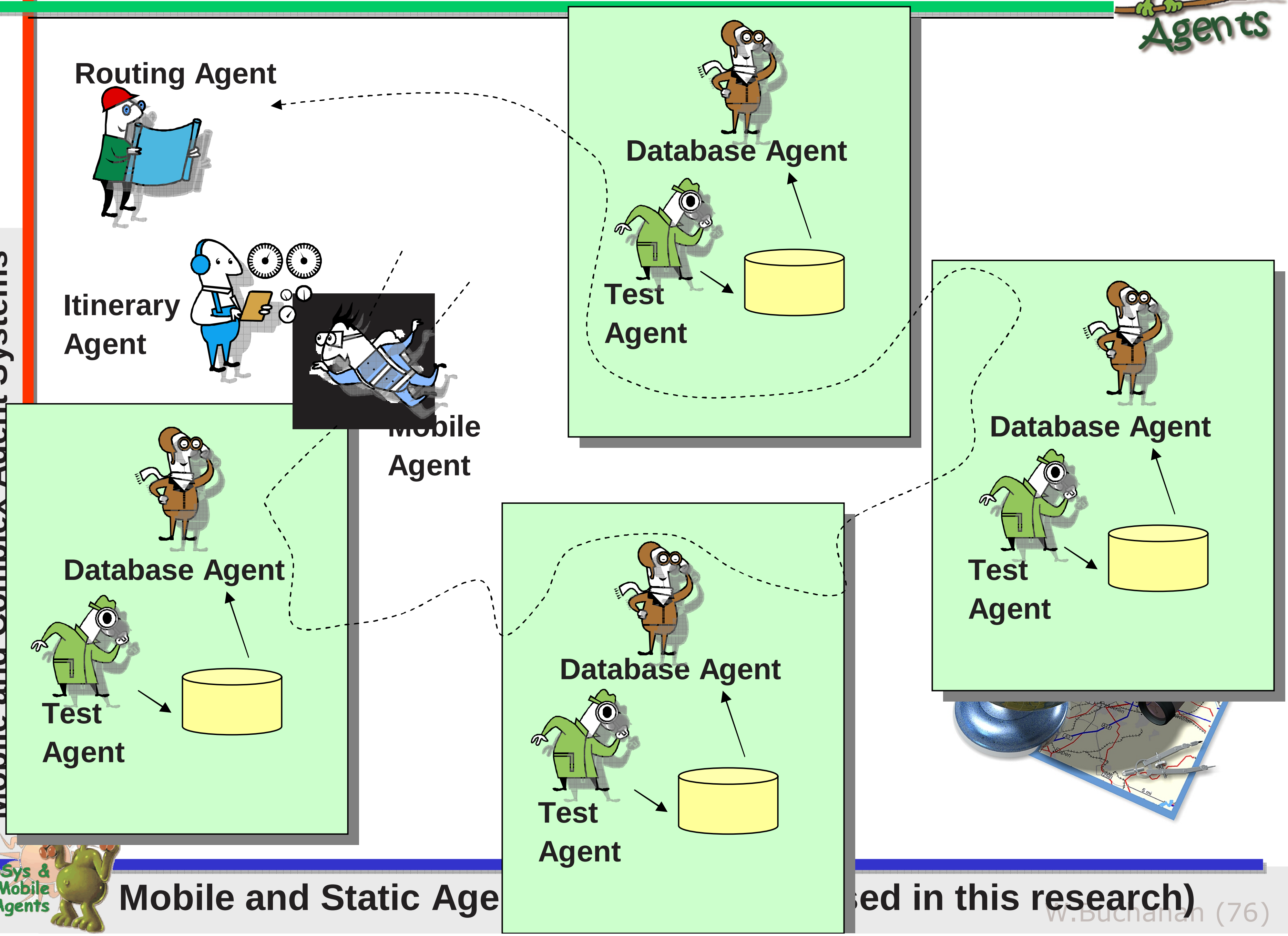
Agents?
- Goal-oriented programs



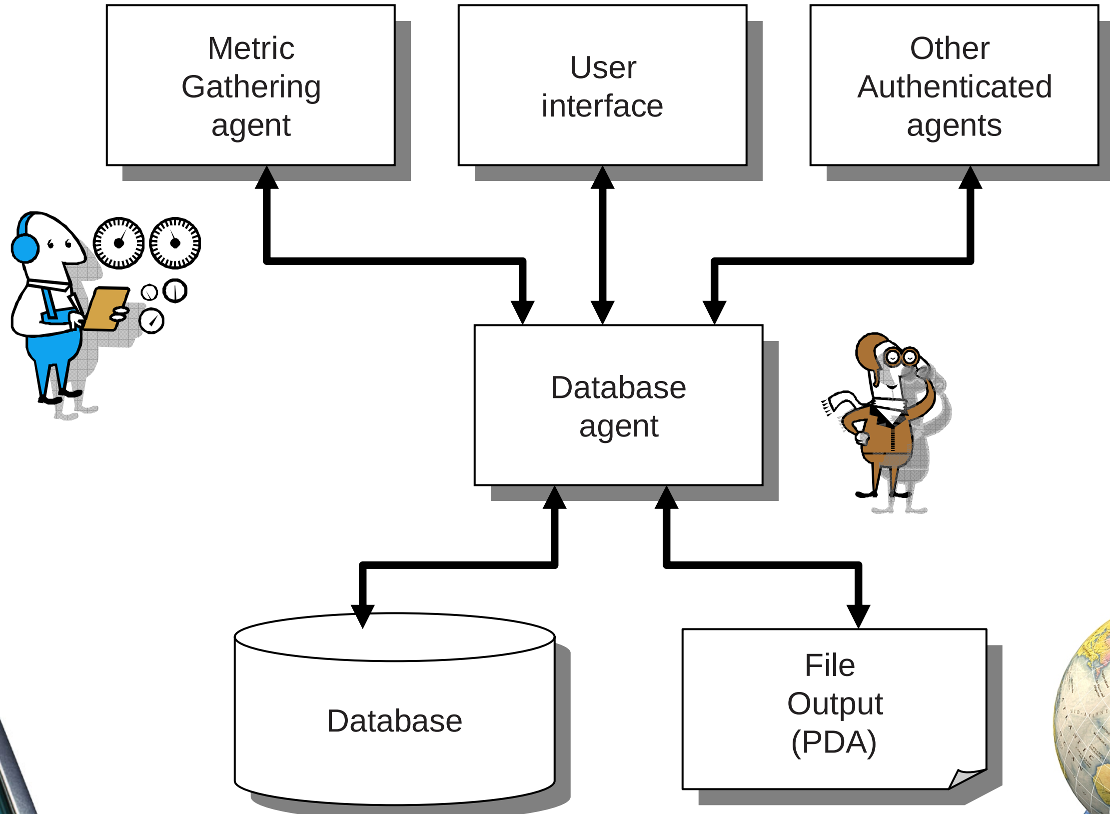
Mobile Agent
- Gathers information on hosts

Static Agent
- Runs routing tests, such as
  Processor tests, network tests, and so on.

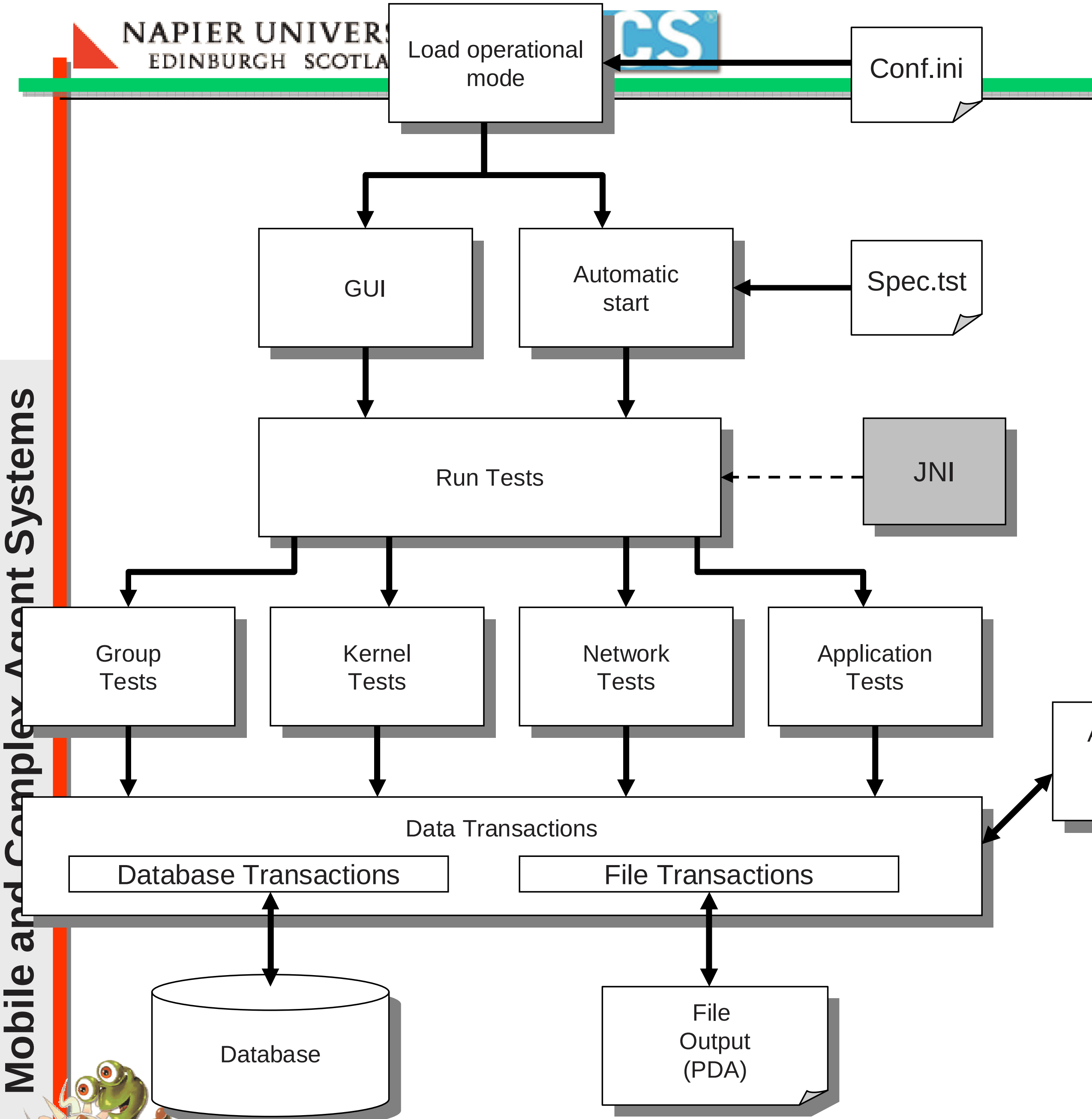
Mobile and Static Agents (How they are used in this research)



Mobile and Static Age… …ed in this research)


Metric
Gathering
agent
User
interface
Other
Authenticated
agents
Database
agent
Database
File
Output
(PDA)

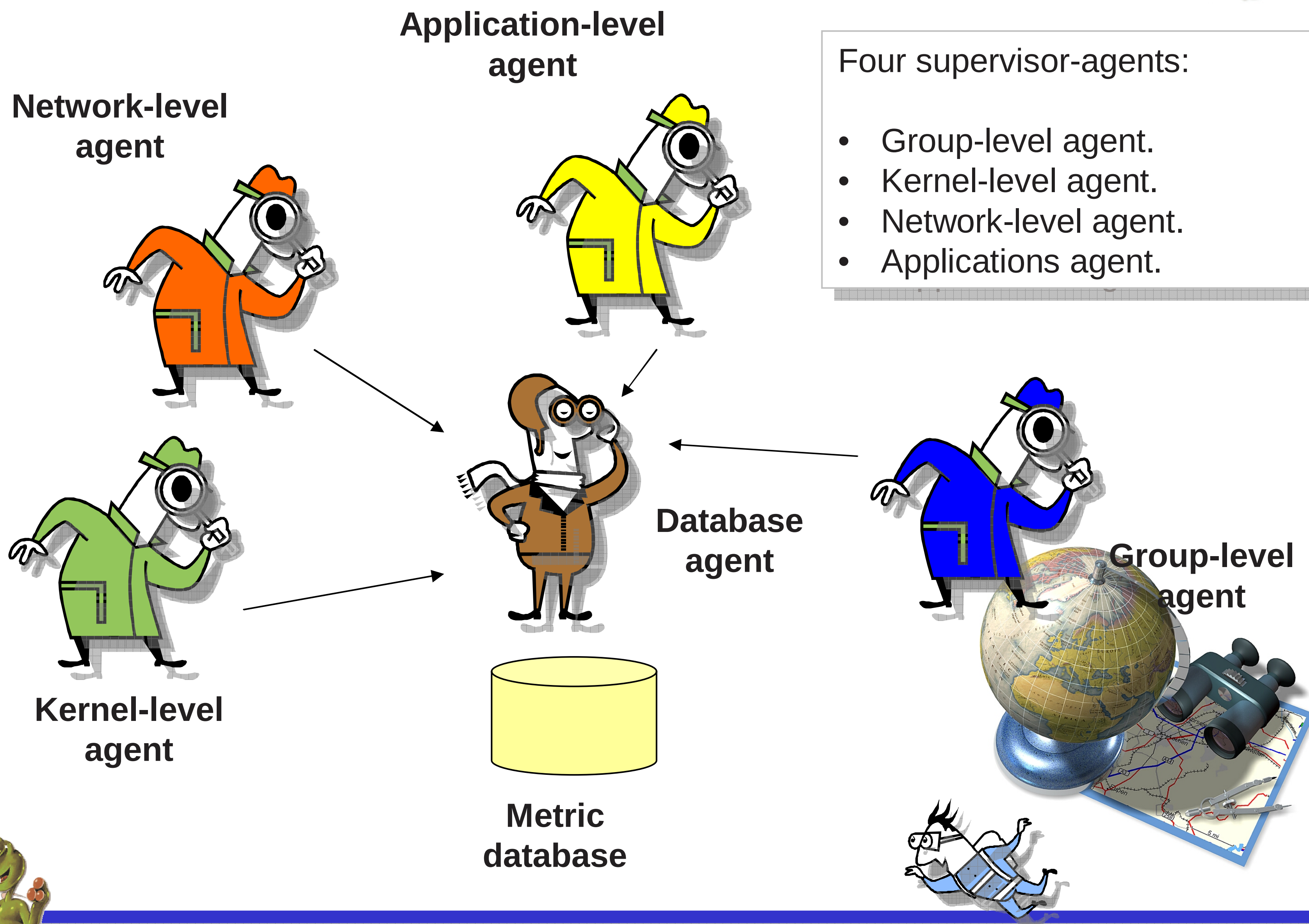

PalmVx
DSys &
Mobile
Agents



Kernel level. These tests aim to identify the hardware characteristics of the device. **Network level**. These tests aim to retrieve the current status of an ad-hoc network, from the viewpoint of a device. **Application level**. These tests aim to monitor the system's utilisation at a given time. **Group level**. These tests aim to group devices according to common characteristics that they may share. For example, all devices with the same operating system (OS) may require to run a specific test, in a specific way.

Four supervisor-agents:

- Group-level agent.
- Kernel-level agent.
- Network-level agent.
- Applications agent.

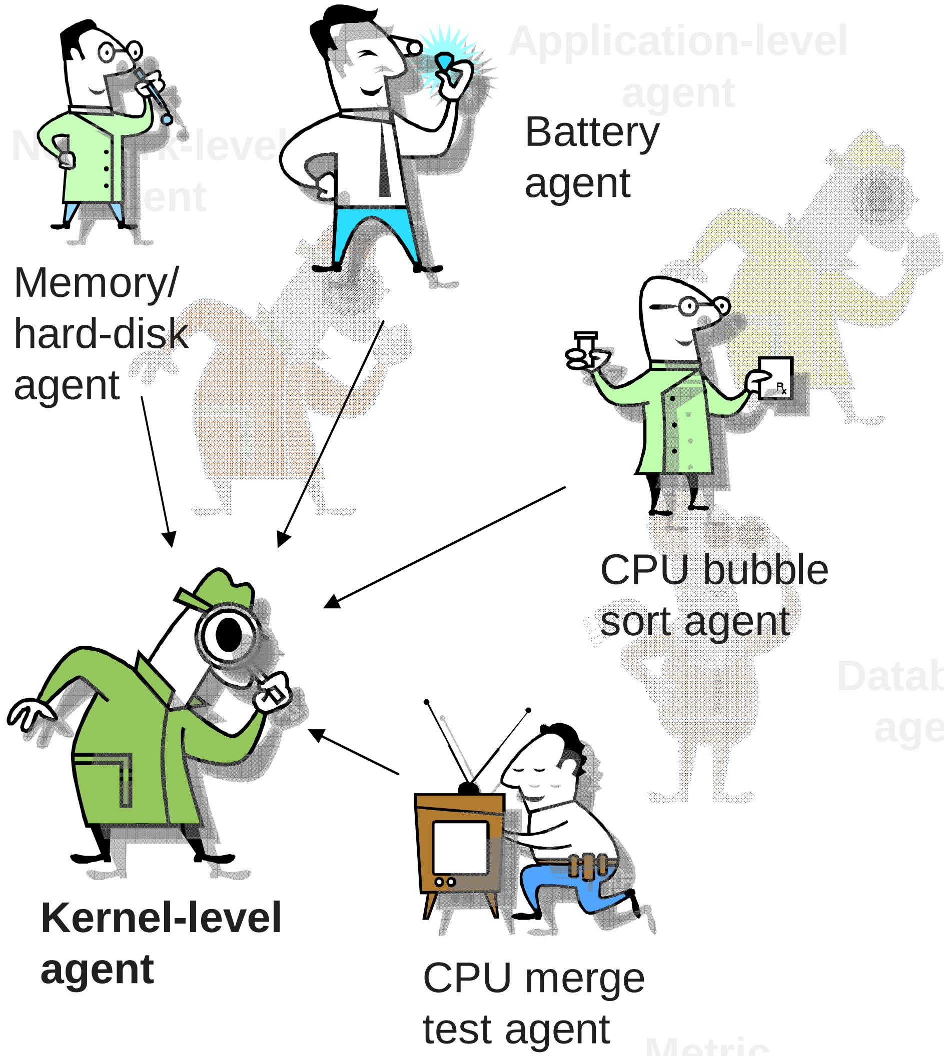


Supervisor-agents

Group-level agent. It does not supervise any other agents, and conducts the test on its own.

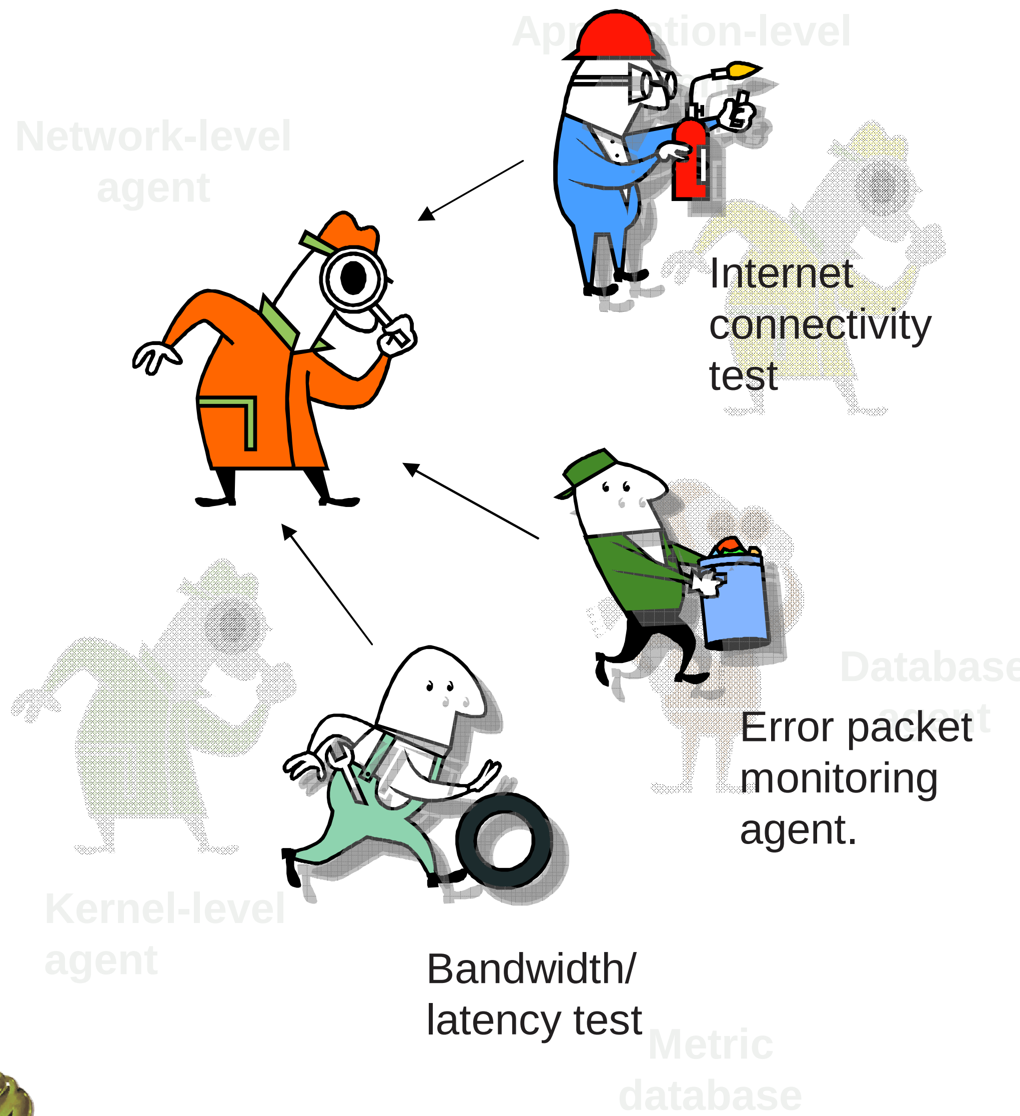
- Check IP address of system, or if there are more than one address.
- Check user to enable authentication.
- Determine Java Runtime Version to establish what Java classes are supported.
- Check operating system, and in the case of Windows CE, execute slightly different performance tests.

Group-level Agent



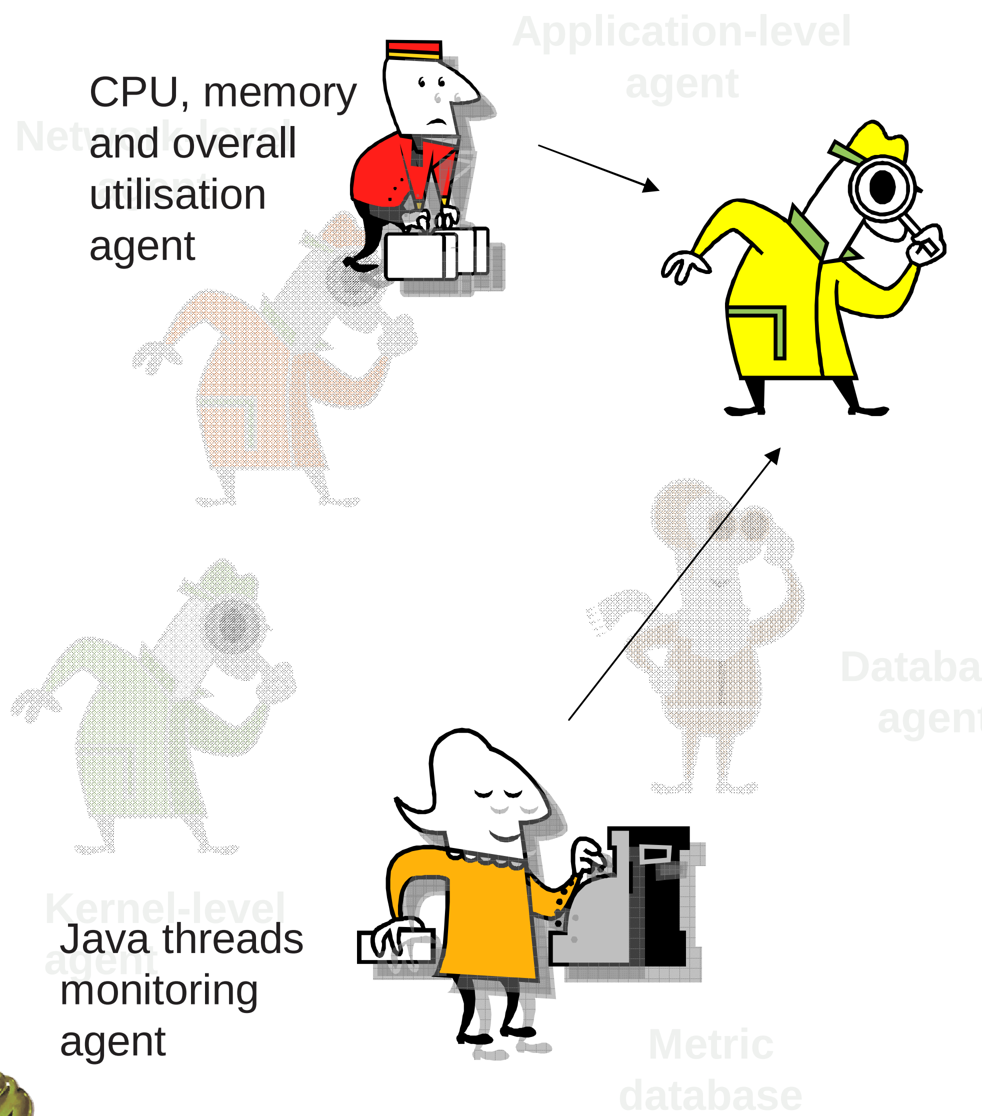
Kernel-level agent. Kernel-level agents can create an instance of the following agents: memory and battery monitoring agent, memory and hard-drive test agent, CPU bubble sort test agent, and CPU merge test agent.

Kernel-level Agent



Network-level agent. The network-level agent can create an instance of the following agents: the Internet connectivity test agent, the bandwidth and latency agent, and the error packets monitoring agent. The Internet connectivity test agent attempts to download an HTML page from the Internet.

Kernel-level Agent



Application-level agent. Can create instances of the following agents: CPU, memory, and overall utilisation agent, and Java threads monitoring agent. The CPU, memory, and overall utilisation agent is responsible for obtaining current CPU and memory usage per process running in the system, as well as total system utilisation. It calls native code via JNI to achieve this. The Java threads monitoring agent is responsible for obtaining the total number of Java threads running and their amount of CPU utilisation. This allows monitoring of the resources used by Java threads on devices. Information gained from these agents, can be used to determine whether, at a given time, a device is capable of routing or not.

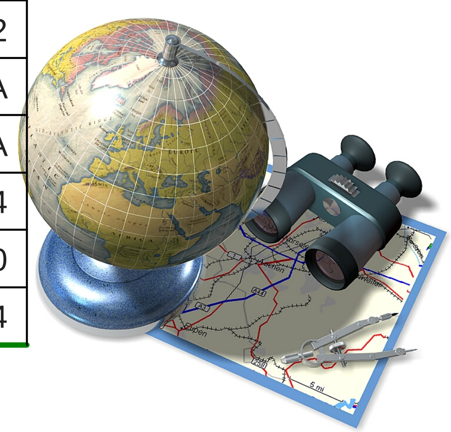
Application-level Agent

PDA. Hewlett Packard iPAQ h5400 series, 450MHz CPU, 64Mb RAM, wireless LAN card; is running Pocket PC 2002.
Laptop. Dell Inspiron, PIII 1100 MHz CPU, 512 MB RAM, Dell TrueMobile wireless LAN card; is running Windows 2000 Professional®.
Server. Dell Desktop PC, PIII 450MHz CPU, 256MB RAM, connected to Dell TrueMobile wireless base station, also connected to broadband router providing Internet access; is running Windows 2000 Server®.
Workstation. P4, 2GHz CPU, 512 MB, Windows XP Professional.

Tests:

CPU bubble sort test (multiple complexity – 1D, 2D, 3D, and 4D).
Memory and hard-drive test.
CPU merge test.
Memory and battery monitoring test.
Internet connectivity test.
Bandwidth and latency test.
Error packets monitoring.
CPU, memory, and overall utilisation test.
Java threads monitoring.
Group-level test.

Metric no.	*Test*	*PDA*	*Laptop*	*Server*	*Workstation*
1	1D	63.66	2.79	5.22	1.12
2	2D	69.74	4.45	8.90	2.00
3	3D	76.00	4.36	12.35	1.78
4	4D	77.04	5.11	14.38	1.88
5	800,000	76.34	1.69	7.38	0.74
6	1 file	84.95	1.35	12.9	1.2
7	1 KB	75.91	3.99	16.10	2.80
8	16 files	82.19	26.77	16.46	6.48
9	16 KB	92.44	4.75	19.26	3.34
10	1m	65.68	47.63	55.65	N/A
11	10m	87.63	47.67	54.87	N/A
12	Internet con	98.93	2.46	1.85	1.12
13	Net Error %	N/A	66.5	N/A	N/A
14	Utilisation	N/A	N/A	50	N/A
15	Average	79.20	16.88	21.15	2.24
16	Battery	50	0	0	0
17	Metric	79.20	16.88	21.15	2.24

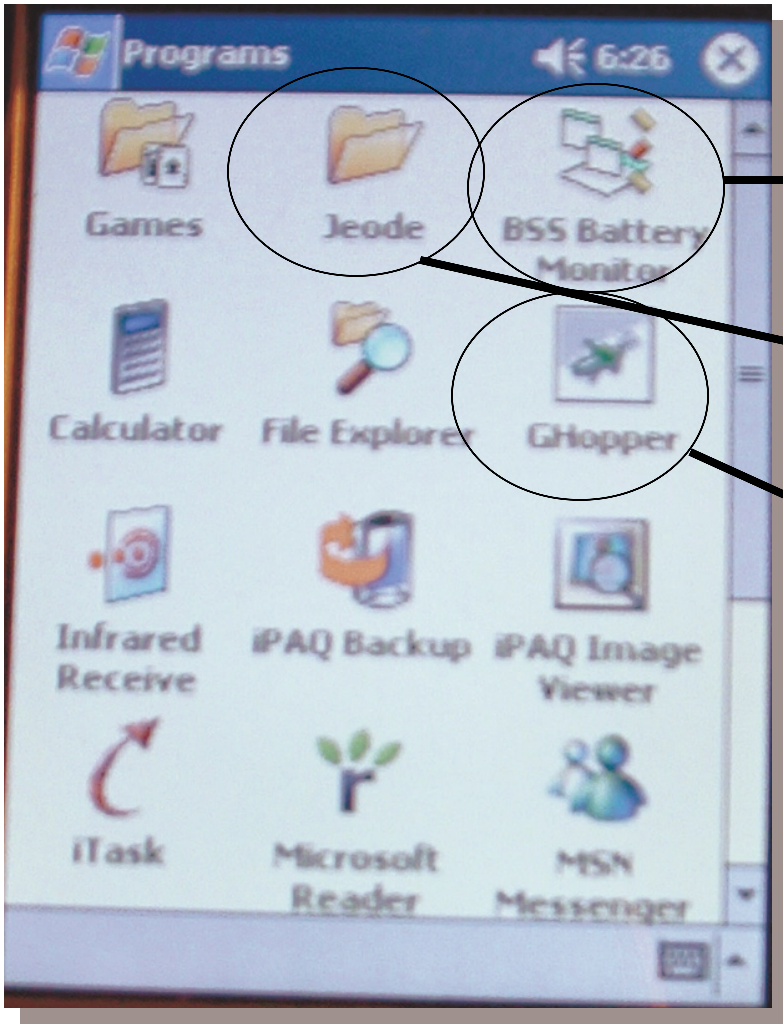


Sample Runs

Background

The great advantage of using Java is that the program will run on any type of machine. In this example a Pocket PC is used as a server, and a notebook as a client. The network connection is over a wireless network.

Java programs on any device which supports Java



Battery monitor - written in VB

Java Virtual Environment

Grasshopper Mobile Agent migration

Conclusions

Bill Buchanan

- Ad-hoc Routing is useful in mobile situations, such as in emergency situations.
- Agent-based systems allow for optimised routing, with different routes for different types of traffic.
- Agents allow for changes to the route and prediction.
- Agents allow for a certain quality of service for a route.