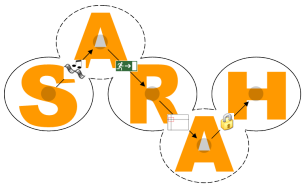


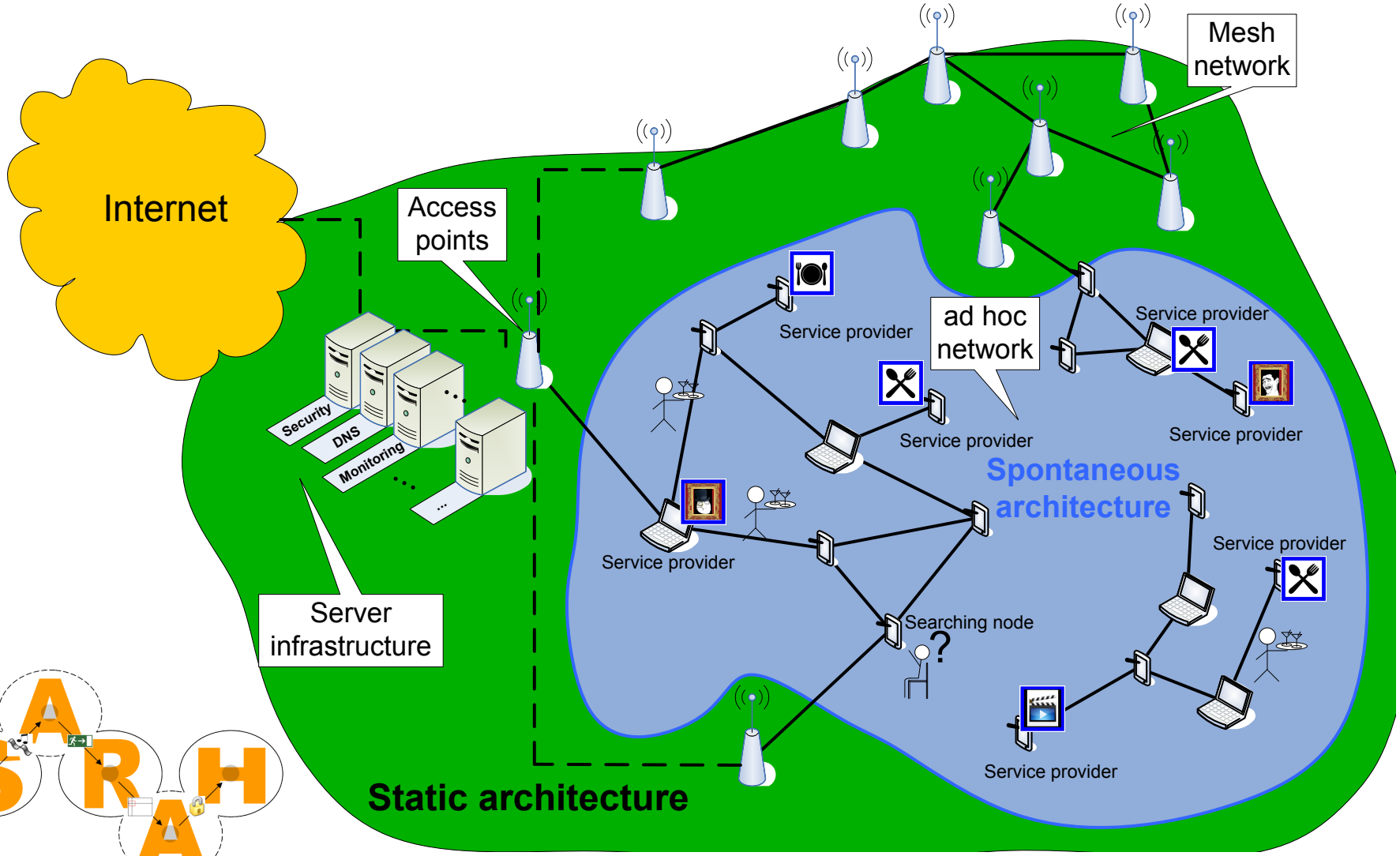
Contributions for Advanced Service Discovery in Ad hoc Networks

Tom Leclerc

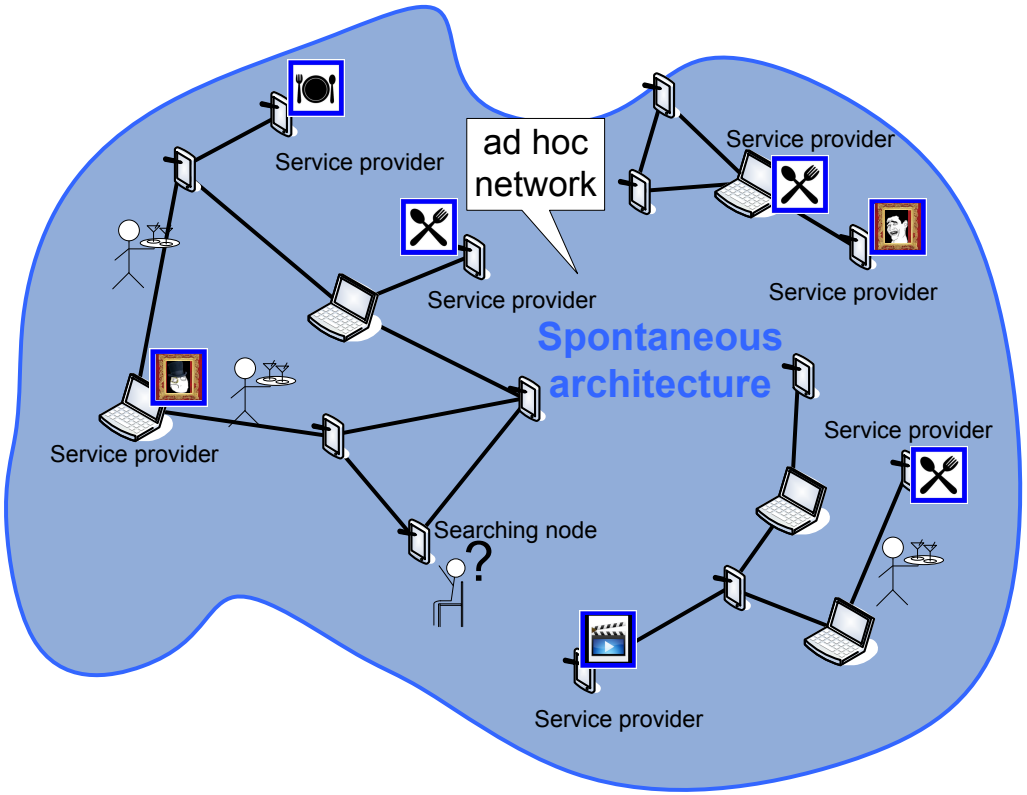
Thèse pour l'obtention du Doctorat de l'Université Henri Poincaré - Nancy 1
réalisée sous la direction d'André Schaff et Laurent Ciarletta
au sein de l'équipe MADYNES
présentée et soutenue publiquement au LORIA, Vandœuvre-lès-Nancy, France

le 24 novembre 2011

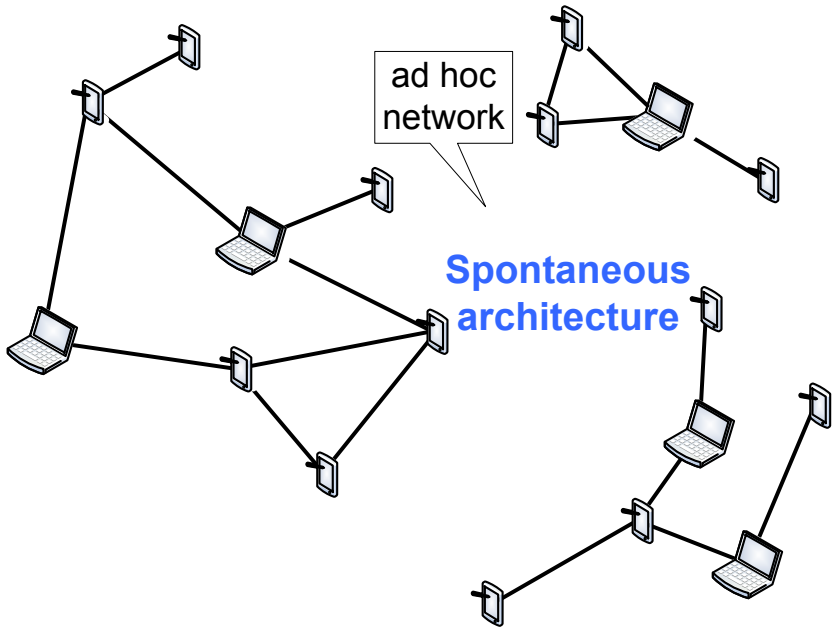
Museum visit

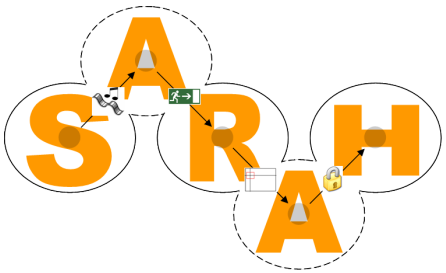


Museum visit



Museum visit





Ad hoc networks

Principles :

- Spontaneous networks
- Nodes are independent and play all roles : emitter, relay and receiver

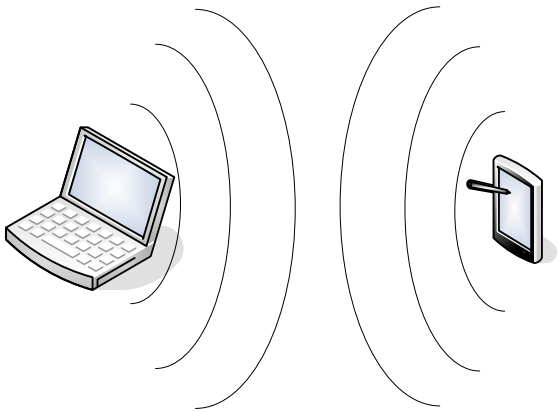
Properties :

- Dynamic (mobility, usage)
- Without any infrastructure
- No central control point
- Uncontrolled environment

Ad hoc networks

Principles :

- Spontaneous networks
- Nodes are independent and play all roles : emitter, relay and receiver



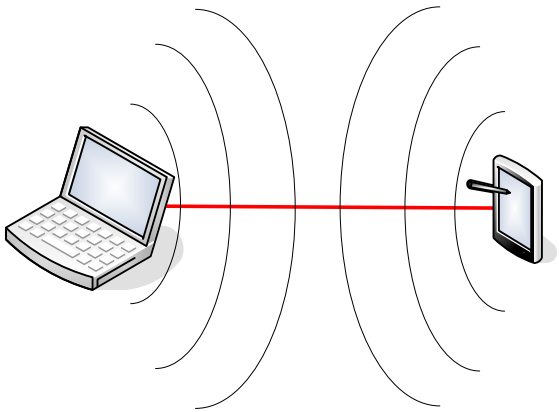
Properties :

- Dynamic (mobility, usage)
- Without any infrastructure
- No central control point
- Uncontrolled environment

Ad hoc networks

Principles :

- Spontaneous networks
- Nodes are independent and play all roles : emitter, relay and receiver



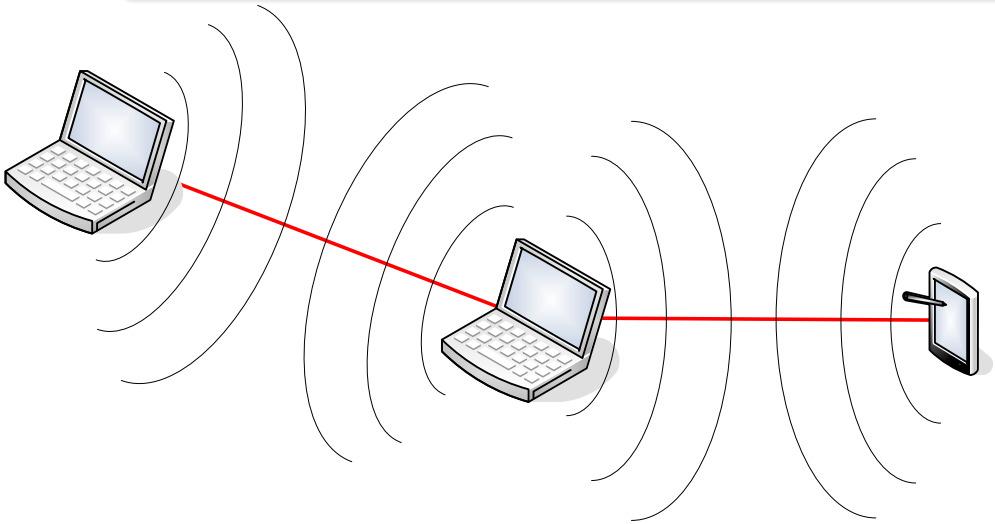
Properties :

- Dynamic (mobility, usage)
- Without any infrastructure
- No central control point
- Uncontrolled environment

Ad hoc networks

Principles :

- Spontaneous networks
- Nodes are independent and play all roles : emitter, relay and receiver



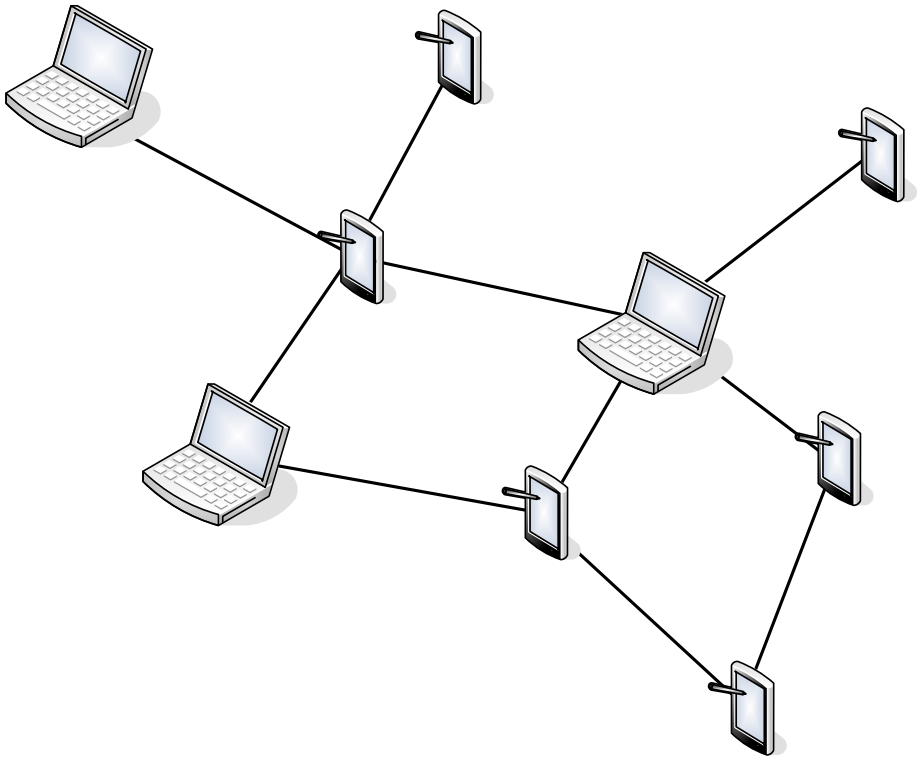
Properties :

- Dynamic (mobility, usage)
- Without any infrastructure
- No central control point
- Uncontrolled environment

Ad hoc networks

Principles :

- Spontaneous networks
- Nodes are independent and play all roles : emitter, relay and receiver



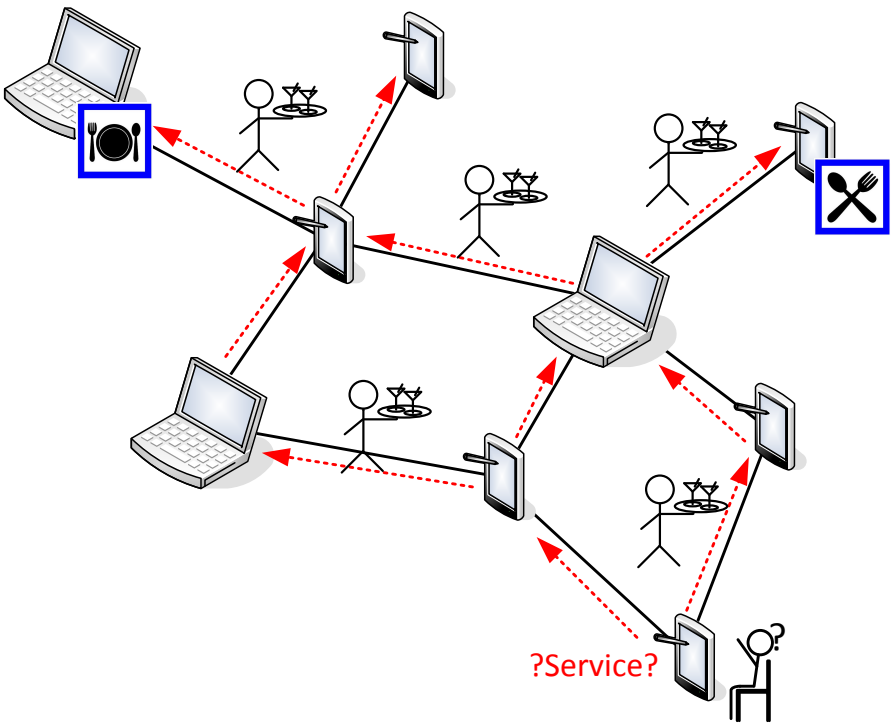
Properties :

- Dynamic (mobility, usage)
- Without any infrastructure
- No central control point
- Uncontrolled environment

Service discovery

Find services in a network :

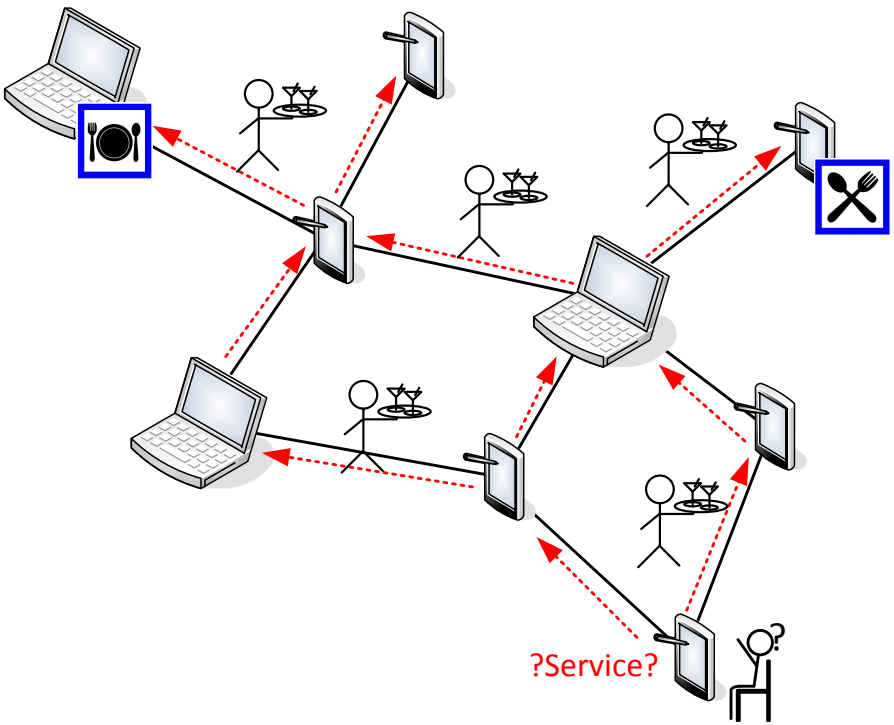
- In an automated way
- Knowing only a basic set of information
- For a remote usage



Service discovery

Find services in a network :

- In an automated way
- Knowing only a basic set of information
- For a remote usage



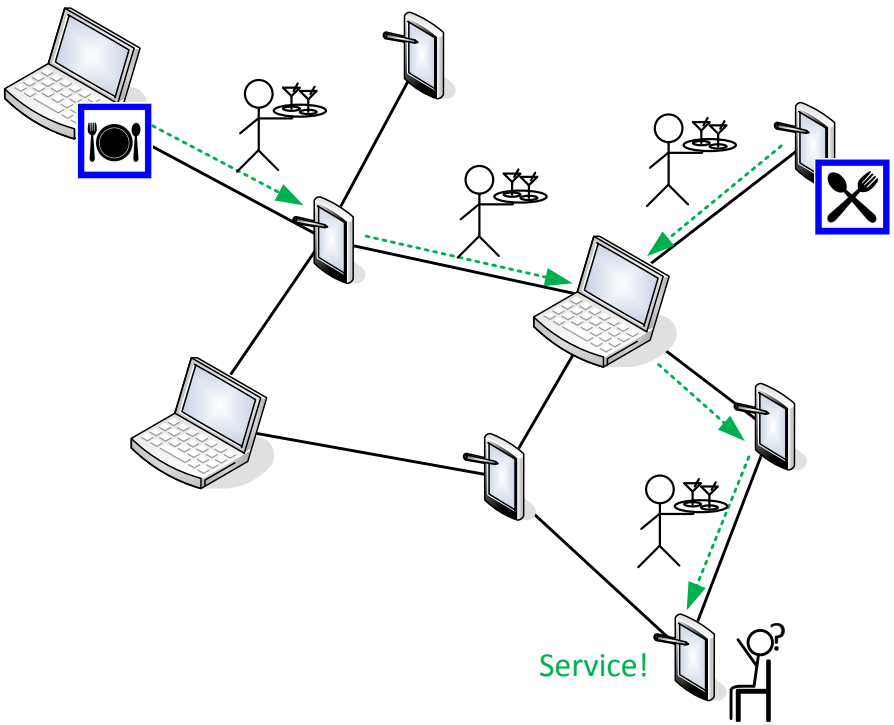
Discovery mechanisms

- Active
  - Send a service request
  - Await matching response messages

Service discovery

Find services in a network :

- In an automated way
- Knowing only a basic set of information
- For a remote usage



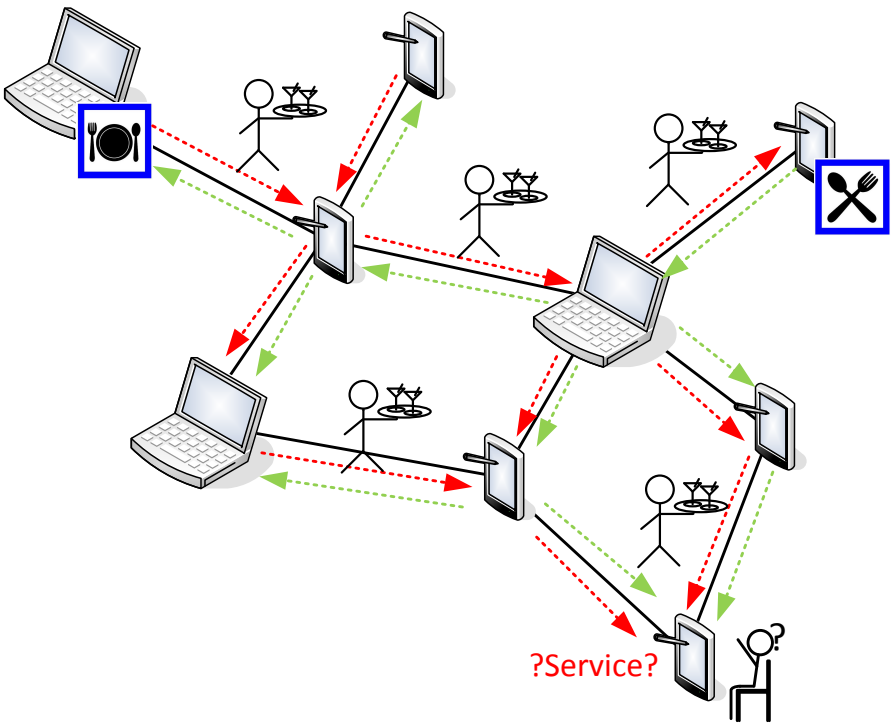
Discovery mechanisms

- Active
  - Send a service request
  - Await matching response messages

Service discovery

Find services in a network :

- In an automated way
- Knowing only a basic set of information
- For a remote usage



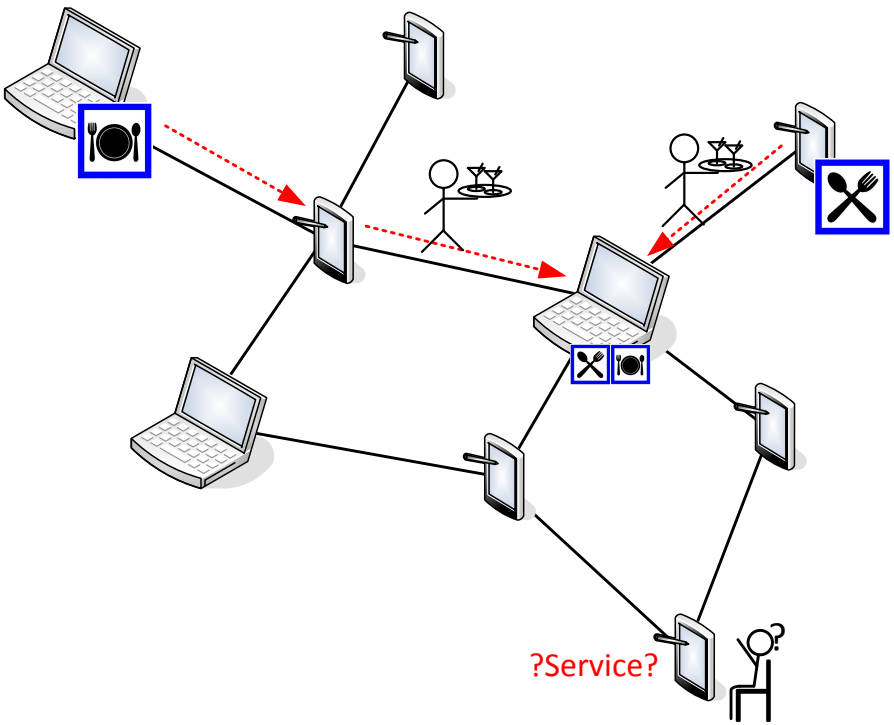
Discovery mechanisms

- Active
- Passive
  - Services are announced periodically
  - Client listen and collect services over time

Service discovery

Find services in a network :

- In an automated way
- Knowing only a basic set of information
- For a remote usage



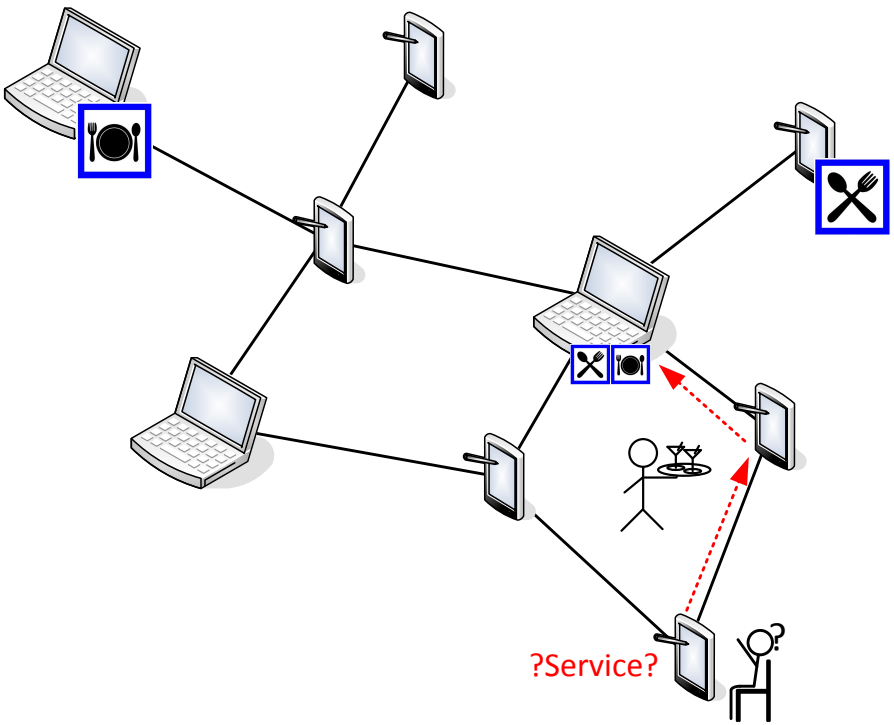
Discovery mechanisms

- Active
- Passive
- Service directories
  - Intermediary nodes
  - Collect and store service announces and requests
  - Permit local discovery

Service discovery

Find services in a network :

- In an automated way
- Knowing only a basic set of information
- For a remote usage



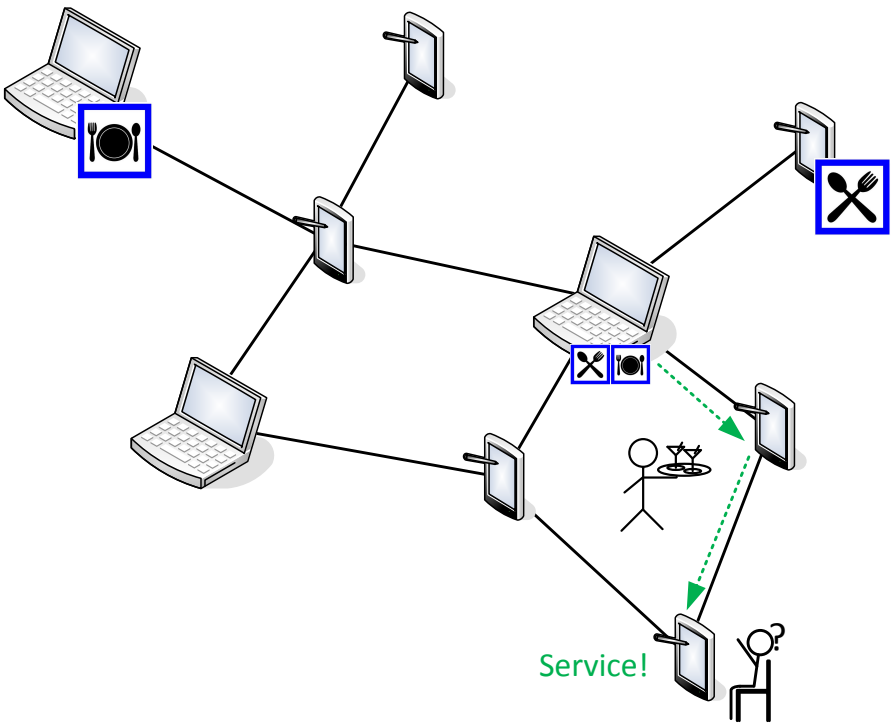
Discovery mechanisms

- Active
- Passive
- Service directories
  - Intermediary nodes
  - Collect and store service announces and requests
  - Permit local discovery

Service discovery

Find services in a network :

- In an automated way
- Knowing only a basic set of information
- For a remote usage



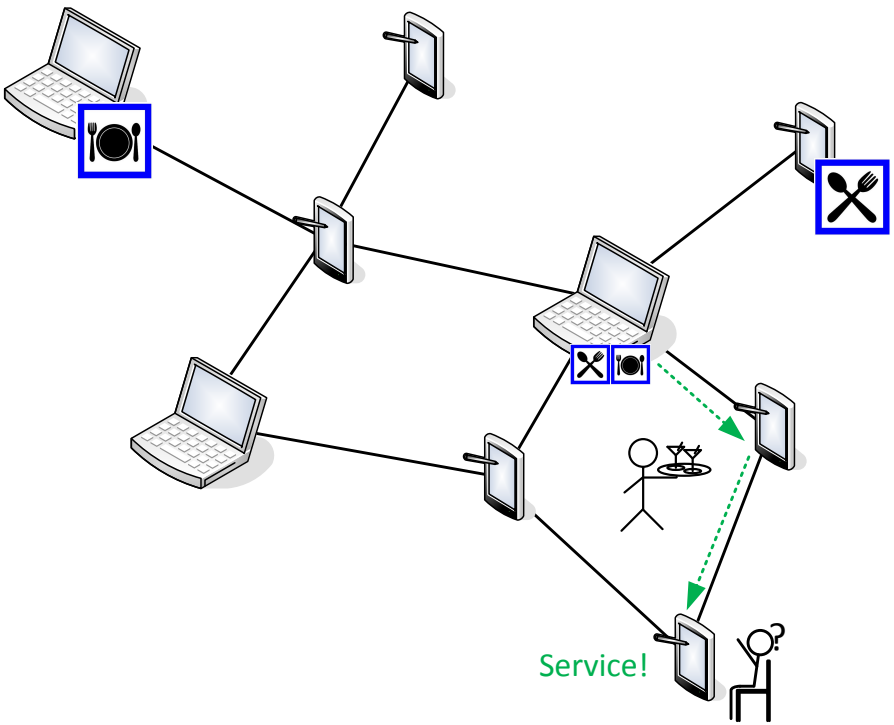
Discovery mechanisms

- Active
- Passive
- Service directories
  - Intermediary nodes
  - Collect and store service announces and requests
  - Permit local discovery

Service discovery

Find services in a network :

- In an automated way
- Knowing only a basic set of information
- For a remote usage



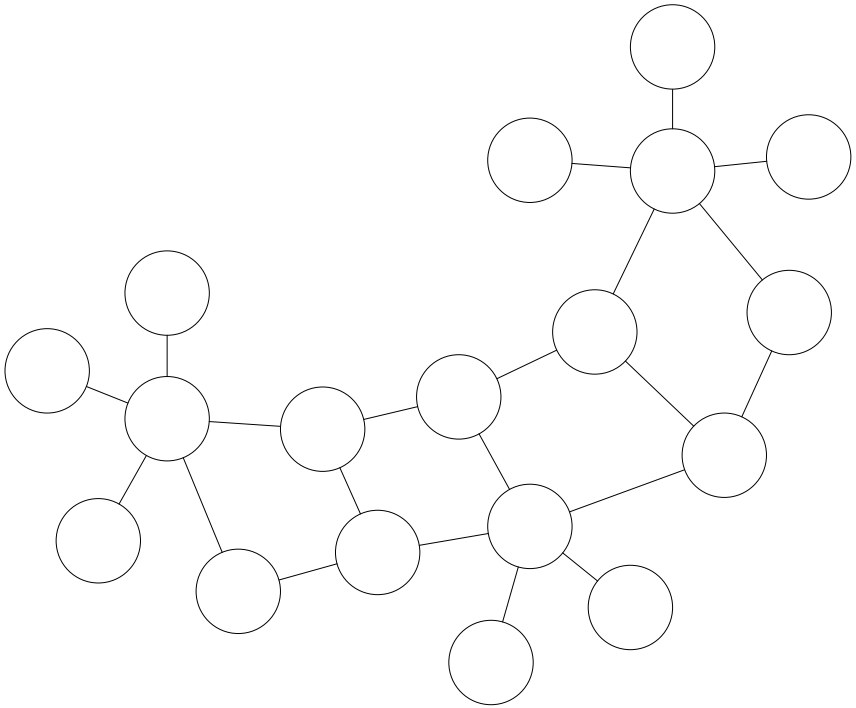
Discovery mechanisms

- Active
- Passive
- Service directories
- Hybrid/Mix

Routing protocols

Find routes to nodes :

- Proactive : Maintain route information up-to-date
- Reactive : Obtain routes only on demand

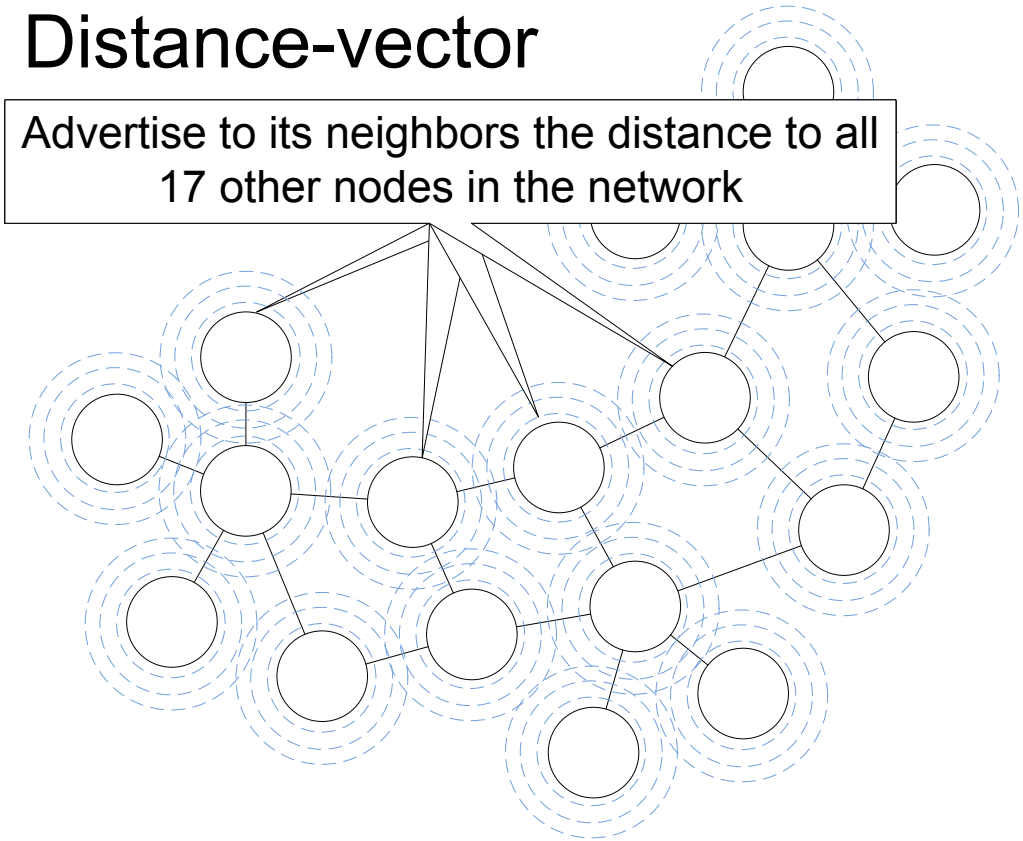


Routing protocols

Find routes to nodes :

- Proactive : Maintain route information up-to-date
- Reactive : Obtain routes only on demand

Distance-vector



Routing strategies

- Distance Vector
  - Distance to a destination
  - Vector(direction) towards destination
  - Announce known routes to neighbors

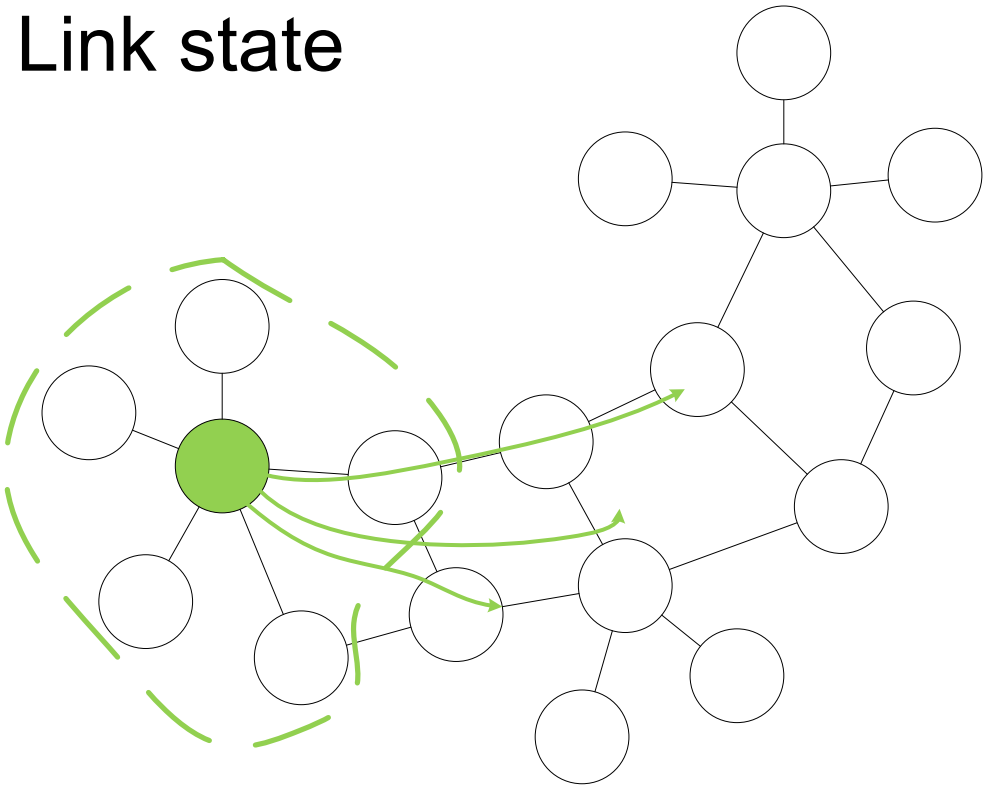
$\Rightarrow$ Known info : Next hop

Routing protocols

Find routes to nodes :

- Proactive : Maintain route information up-to-date
- Reactive : Obtain routes only on demand

Link state



Routing strategies

- Distance Vector
- Link State
  - Disseminate announces containing the list of neighors in all the network

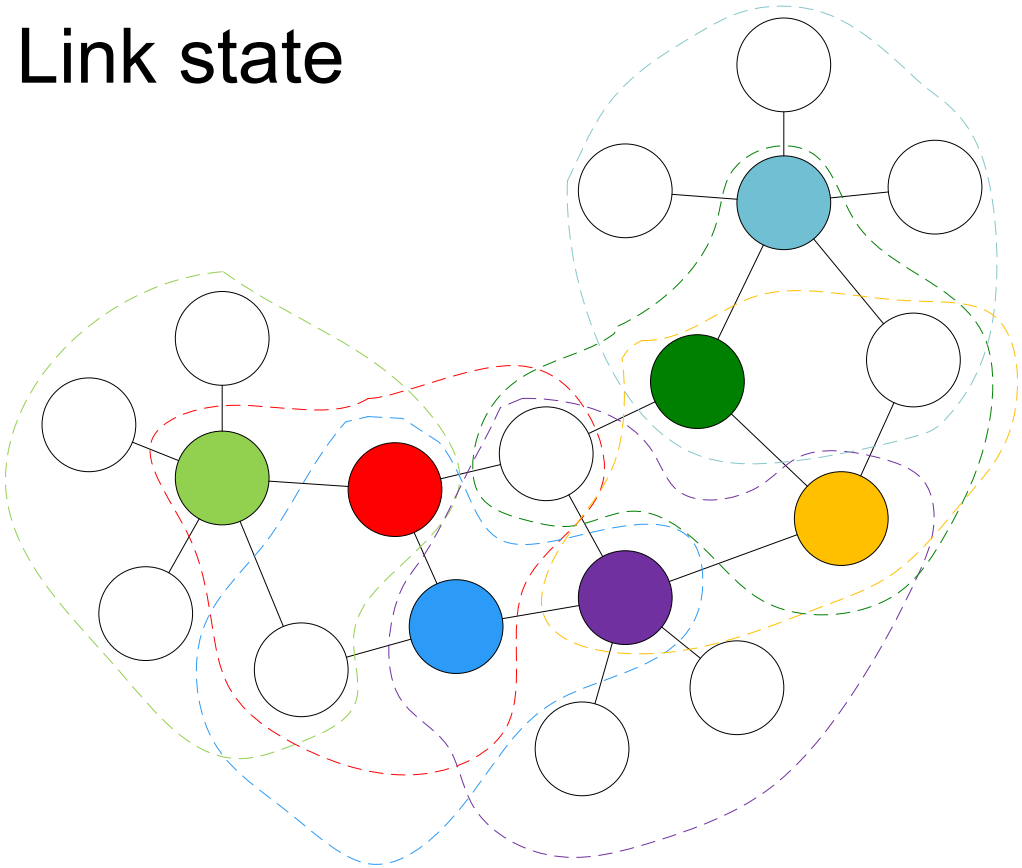
$\Rightarrow$ Known info : Complete topology

Routing protocols

Find routes to nodes :

- Proactive : Maintain route information up-to-date
- Reactive : Obtain routes only on demand

Link state



Routing strategies

- Distance Vector
- Link State
  - Disseminate announces containing the list of neighors in all the network

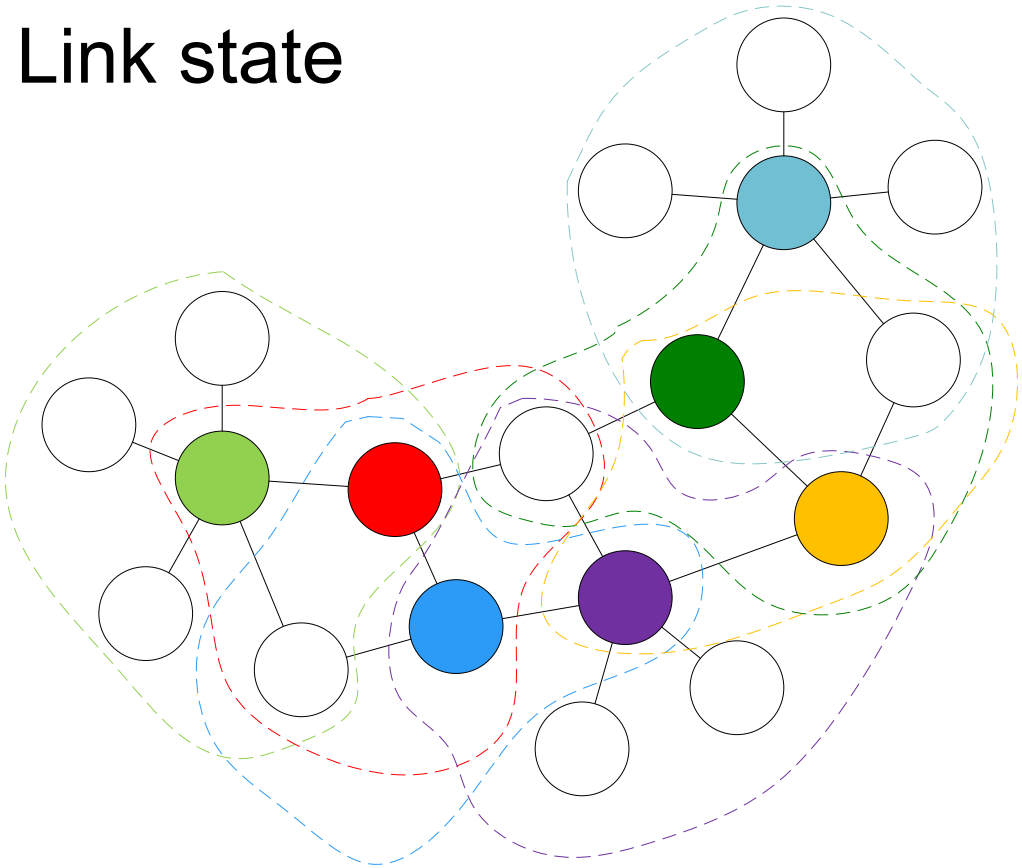
$\Rightarrow$ Known info : Complete topology

Routing protocols

Find routes to nodes :

- Proactive : Maintain route information up-to-date
- Reactive : Obtain routes only on demand

Link state

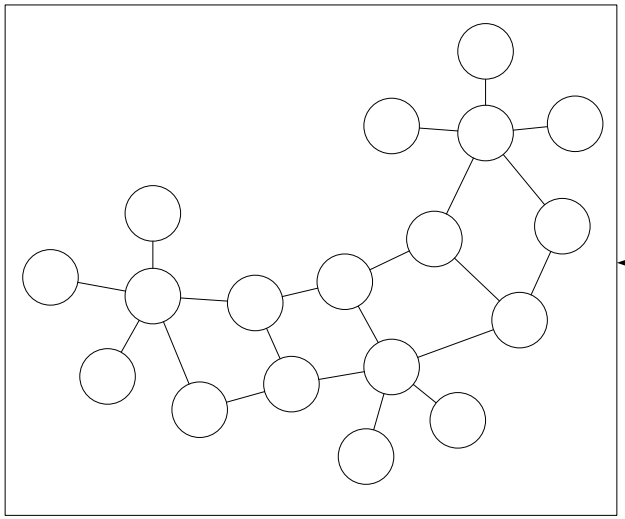


Routing strategies

- Distance Vector
- Link State
- Hybrid/Mix

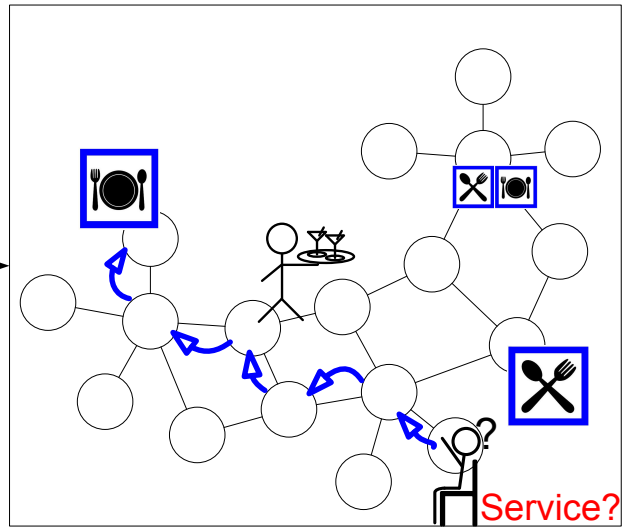
Problematic

Ad hoc network



?

(Advanced) Service discovery



What we want to achieve :

Find the best service

- Stability
- Distance
- Quality
- Availability

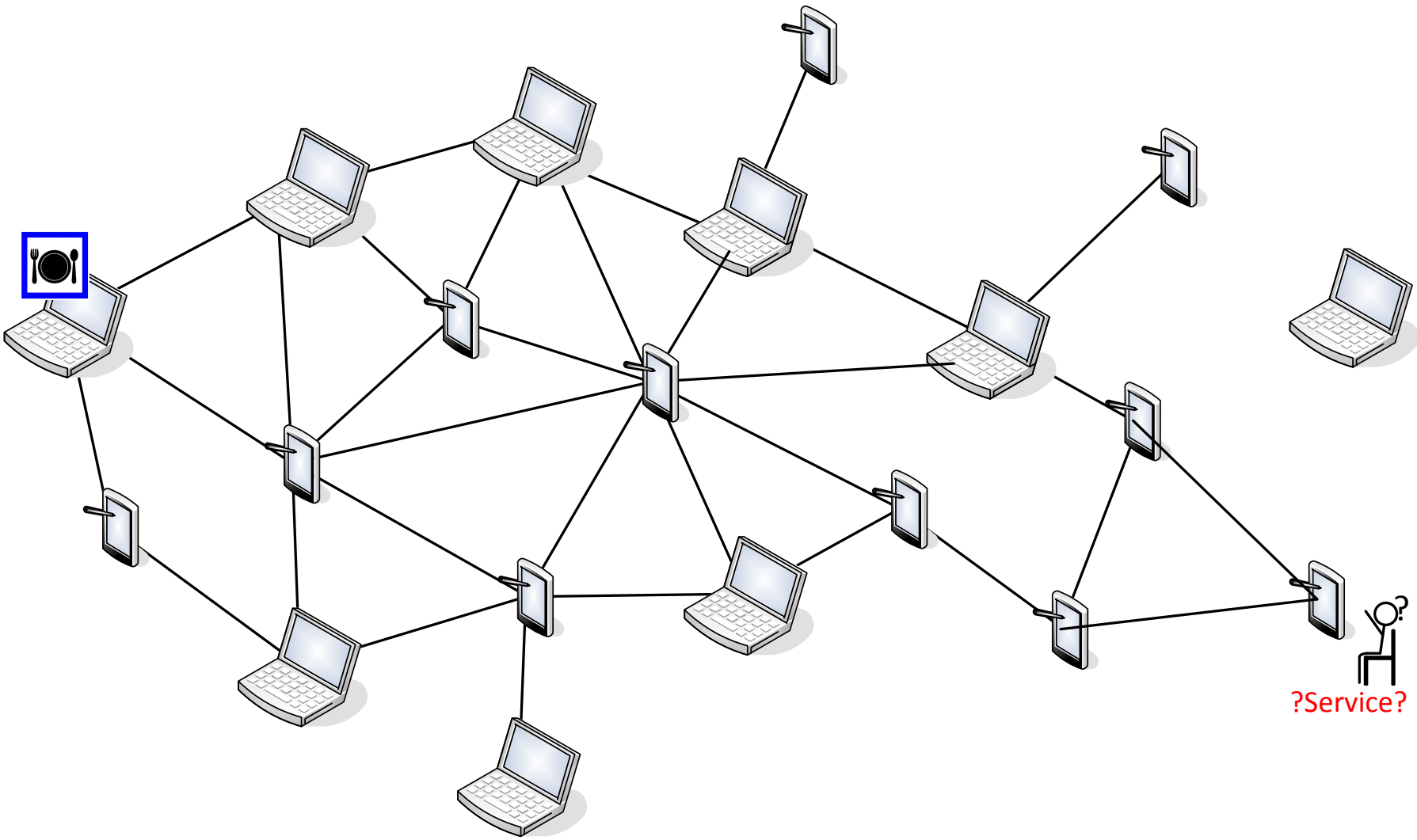
Find services efficiently

- Bandwidth used
- Number of nodes involved
- Use all available information wisely

How can we achieve both ? $\Rightarrow$ Find the right balance

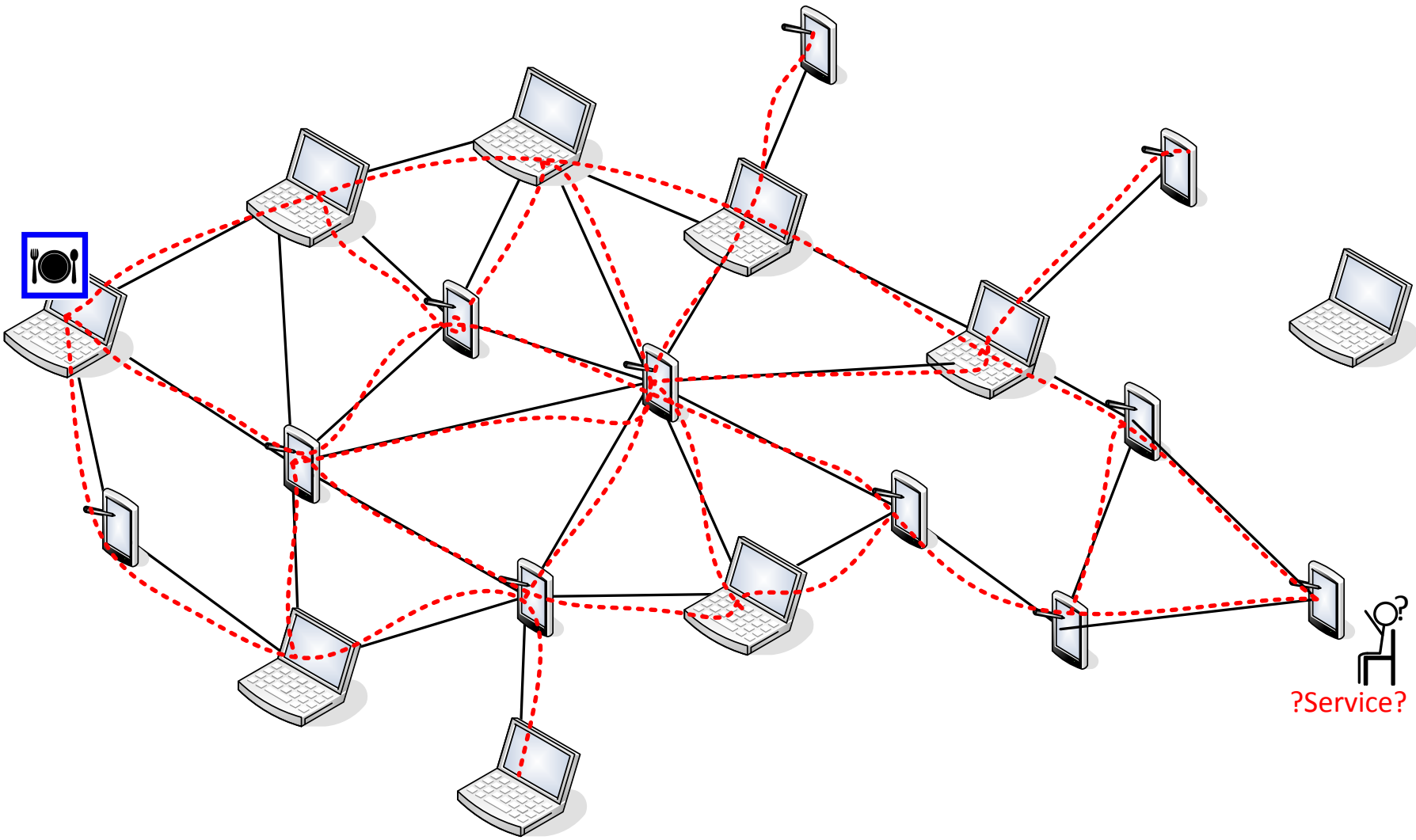
The challenges in ad hoc networks :

⇛ Dissemination



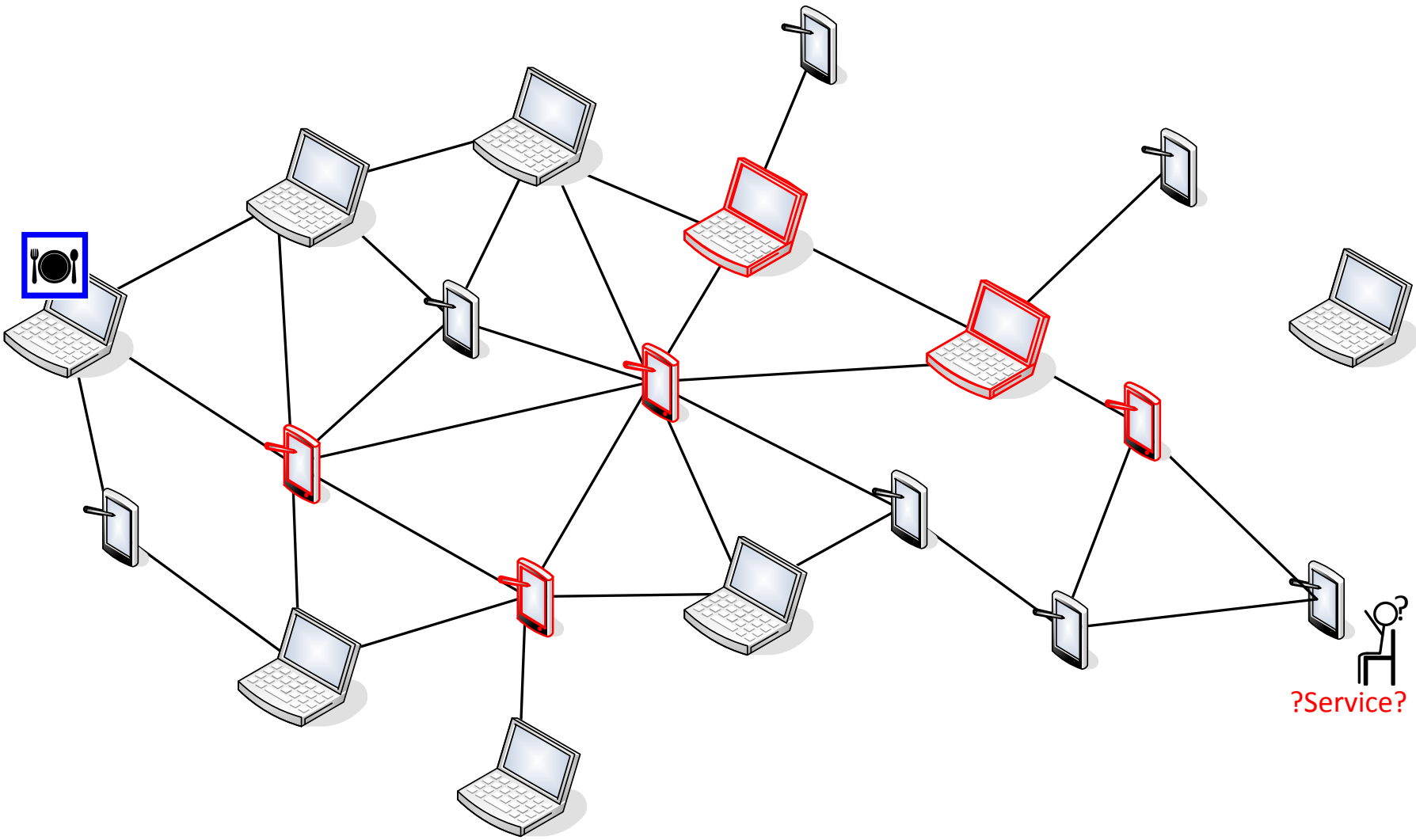
The challenges in ad hoc networks :

⇛ Dissemination



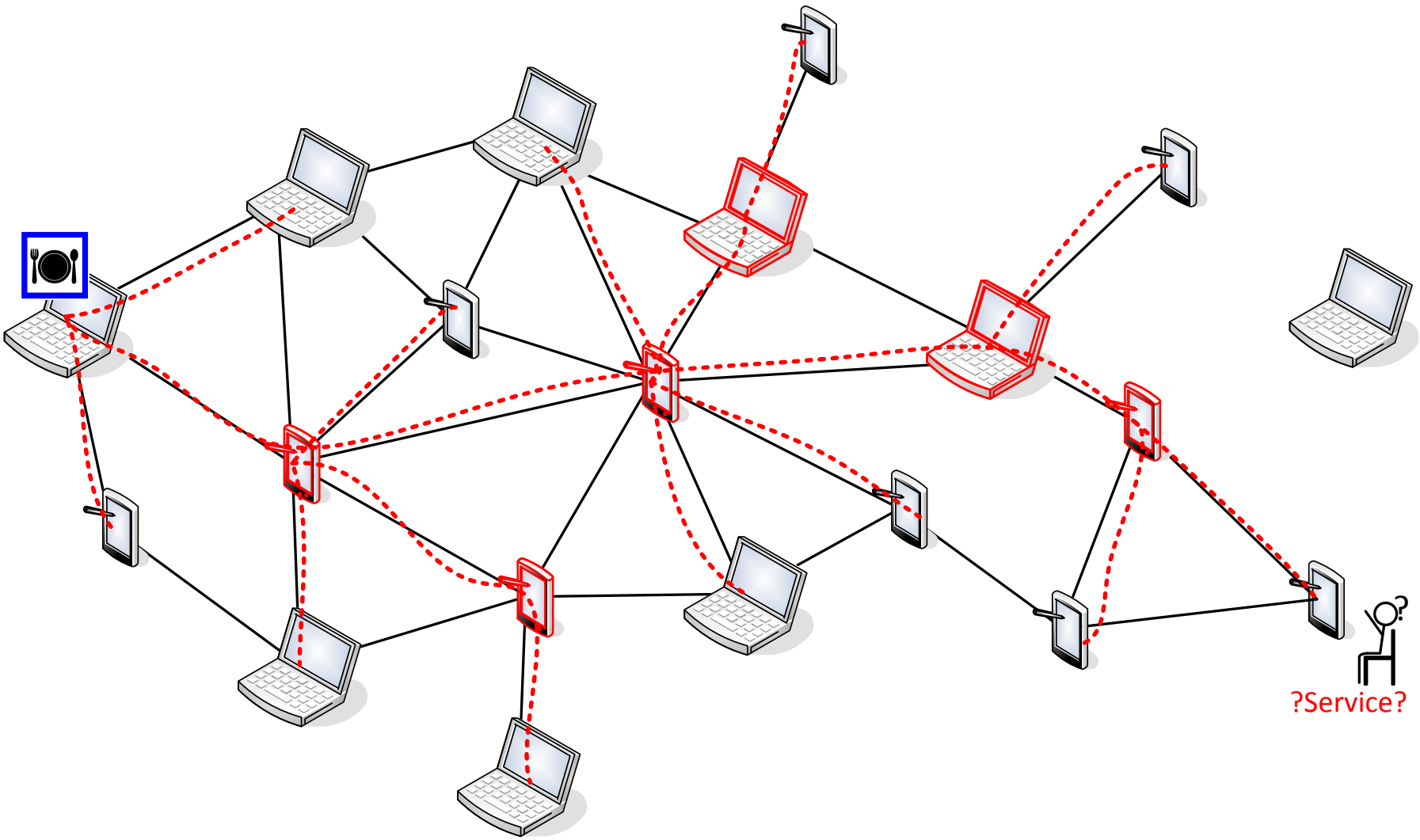
The challenges in ad hoc networks :

⇛ Dominating nodes[Blum 04]



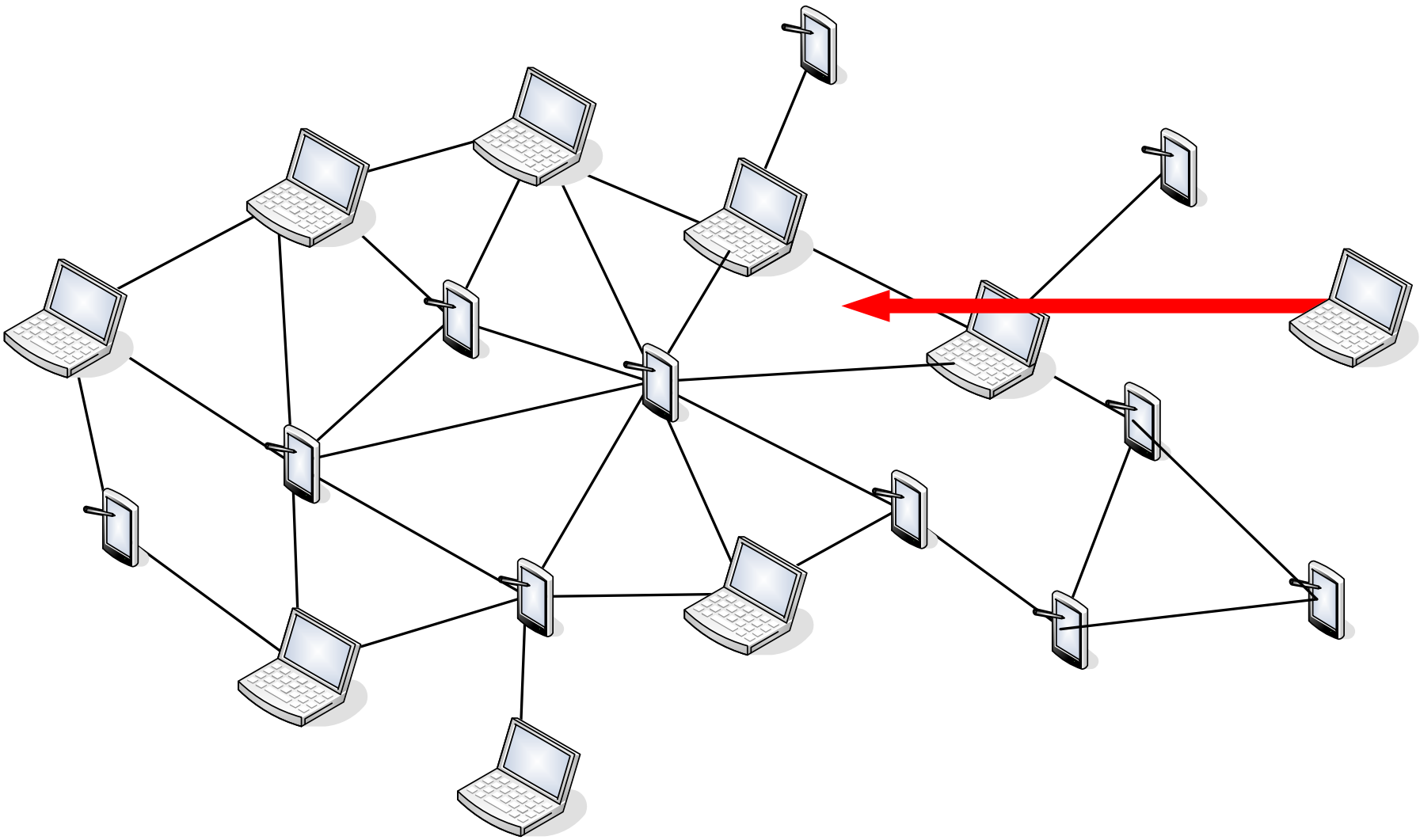
The challenges in ad hoc networks :

⇛ **Dominating nodes[Blum 04]**



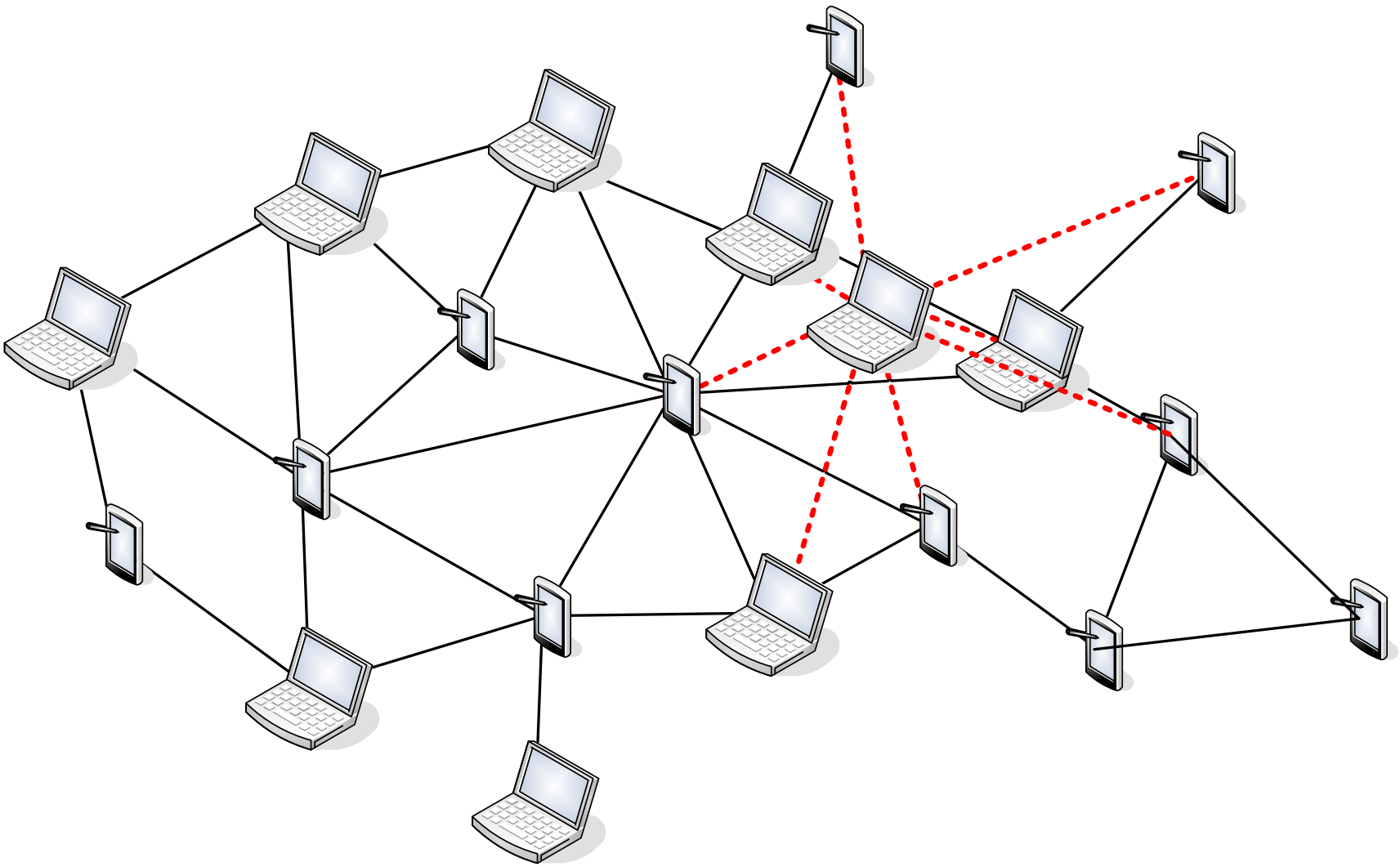
The challenges in ad hoc networks :

⇛ **Dynamicity**



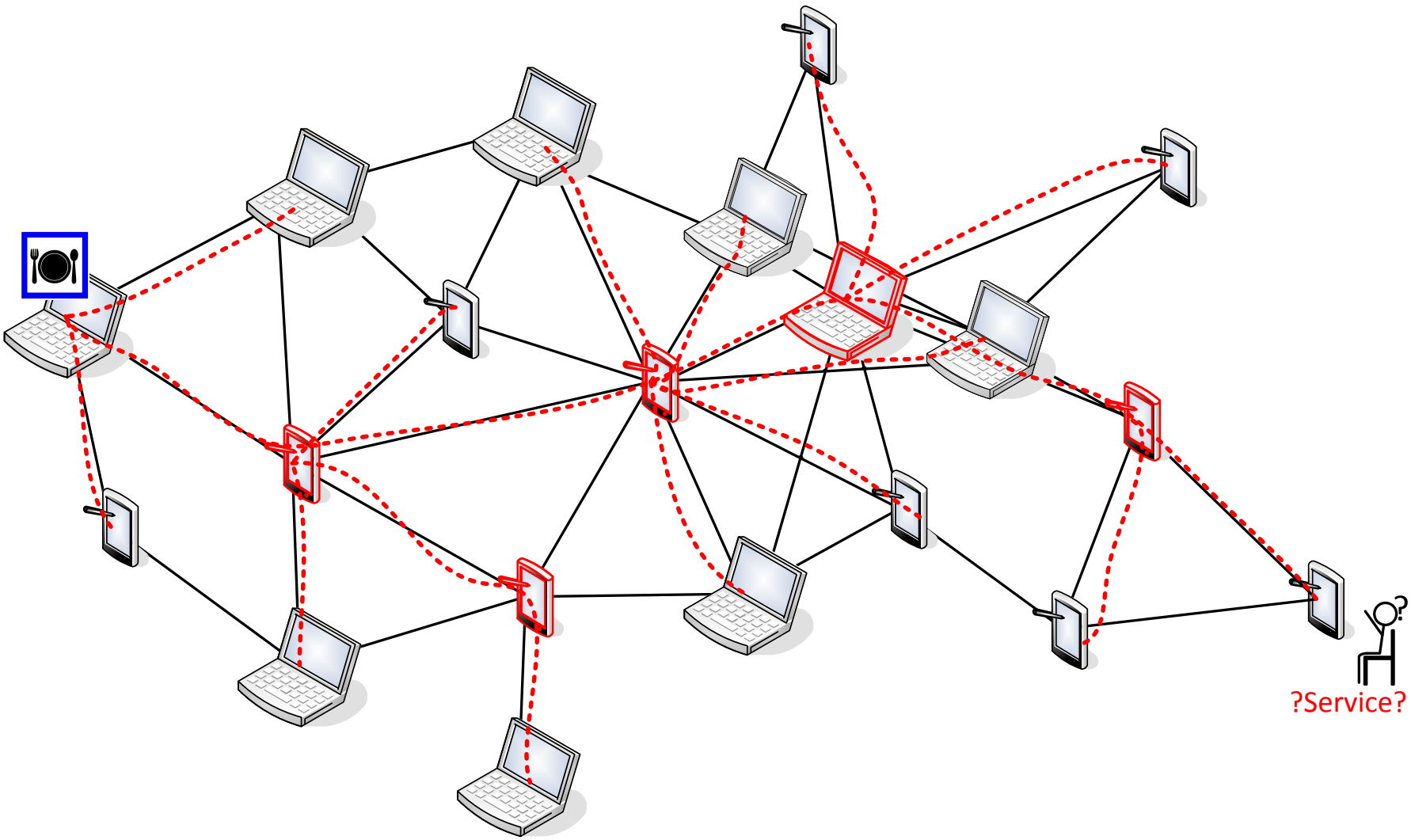
The challenges in ad hoc networks :

⇛ **Dynamicity**



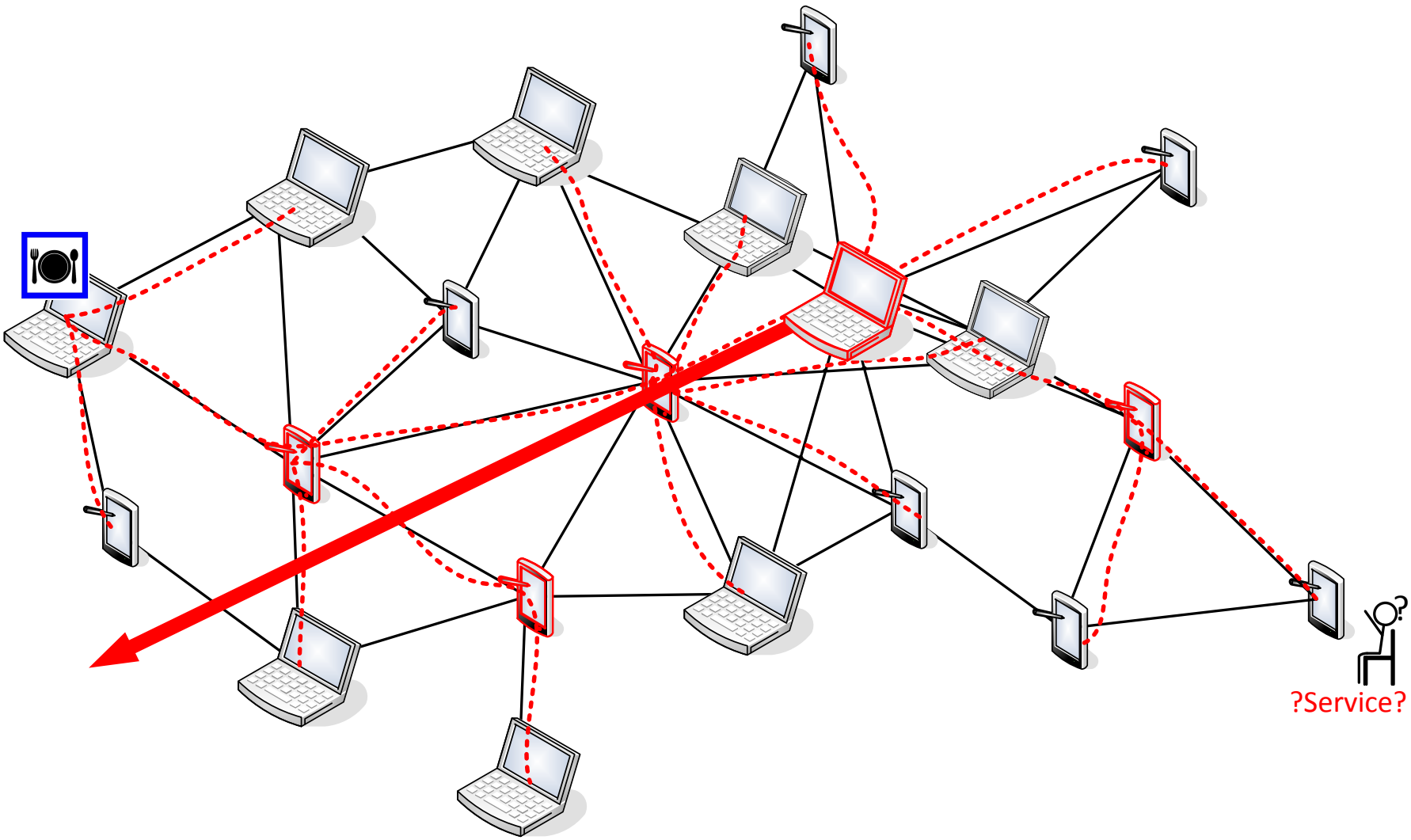
The challenges in ad hoc networks :

⇛ **Dynamicity** ⇛ **Stability**



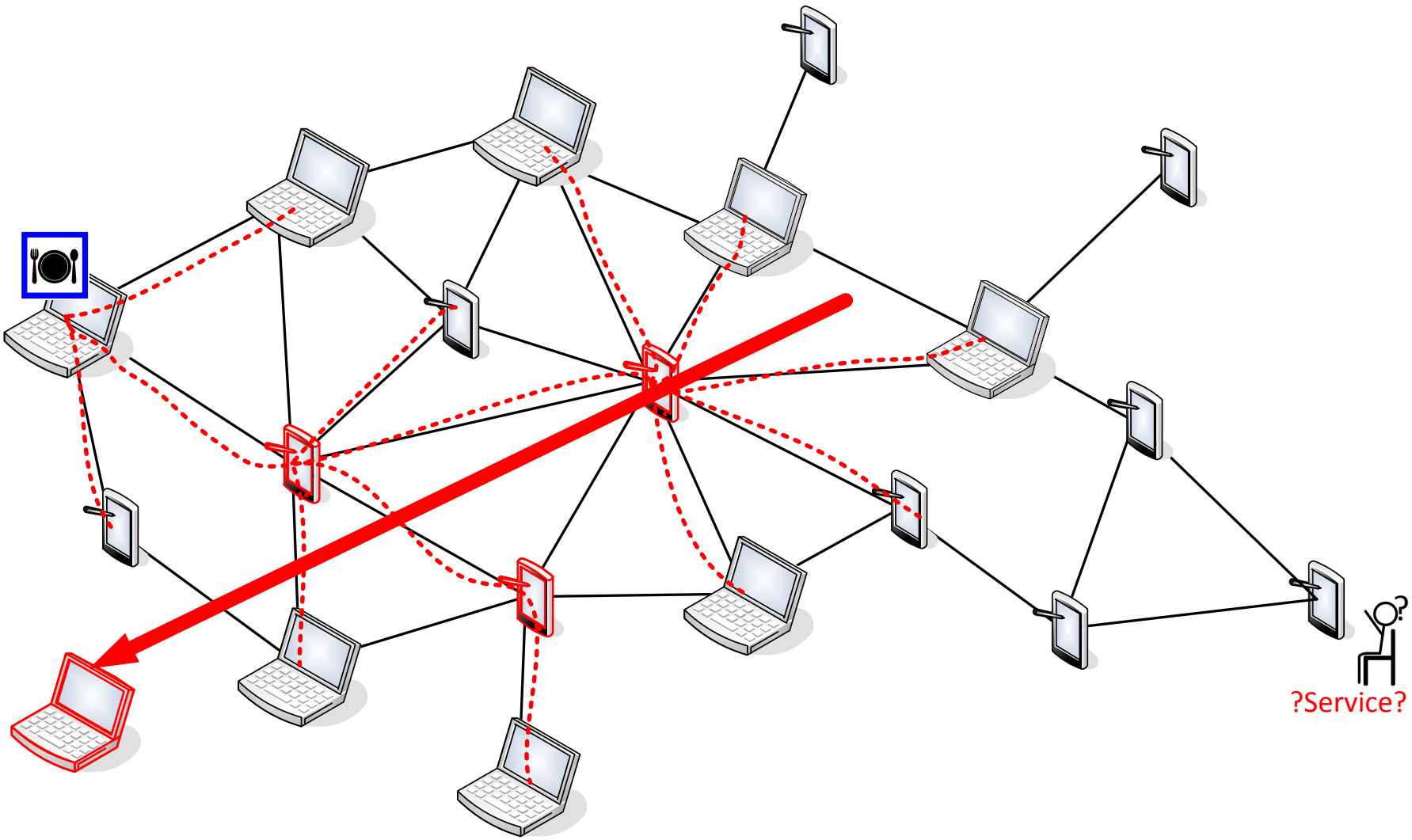
The challenges in ad hoc networks :

⇛ **Dynamicity** ⇛ **Stability**

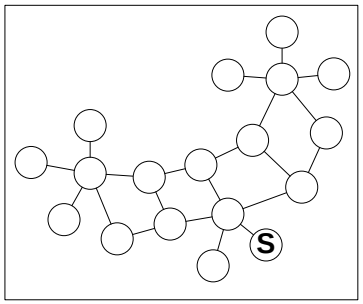


The challenges in ad hoc networks :

⇛ **Dynamicity** ⇛ **Stability**

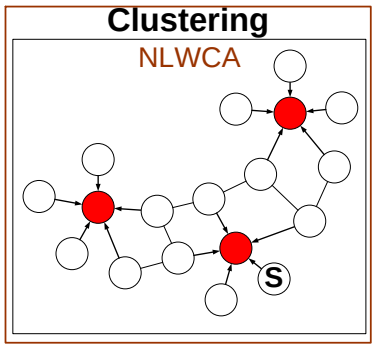


Contributions for Advanced Service Discovery



Contributions for Advanced Service Discovery

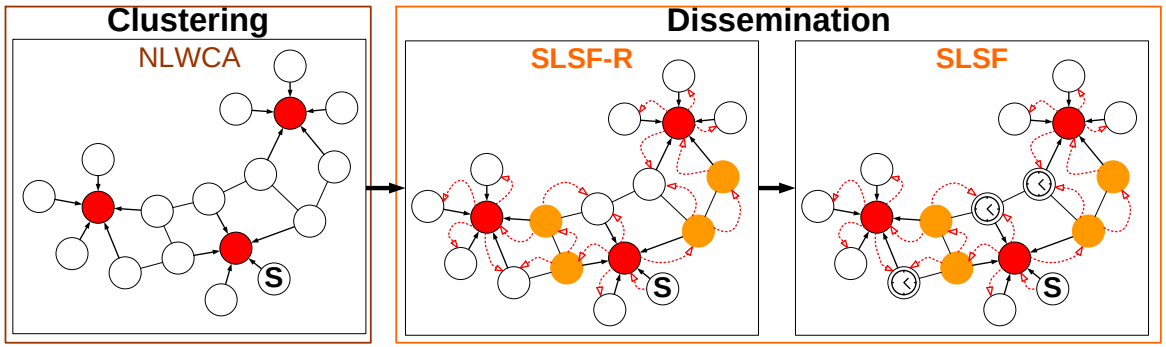
- Stability, Reference nodes



Collaborations :
A. Andronache, S. Rothkugel

Contributions for Advanced Service Discovery

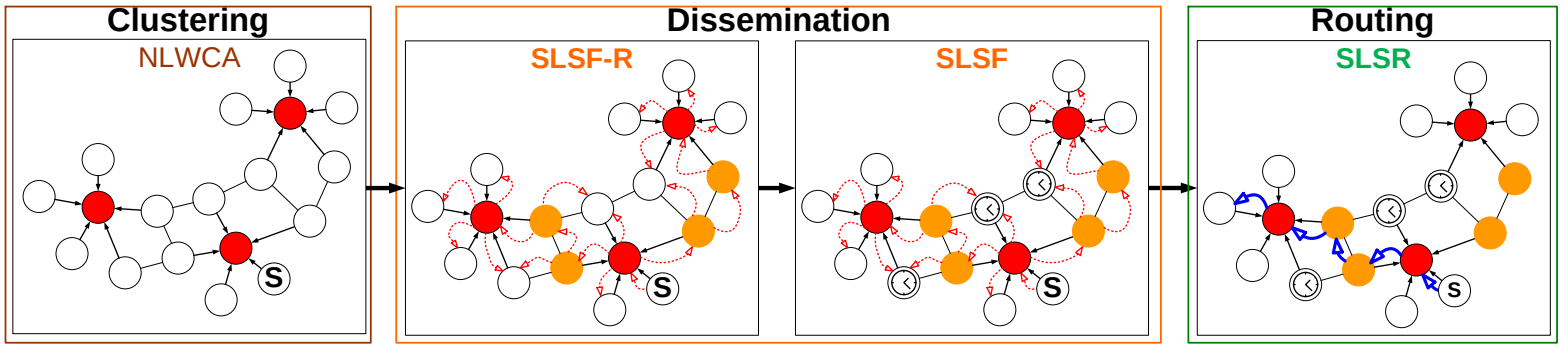
- Stability, Reference nodes
- Efficient message dissemination



Collaborations :
A. Andronache, S. Rothkugel

Contributions for Advanced Service Discovery

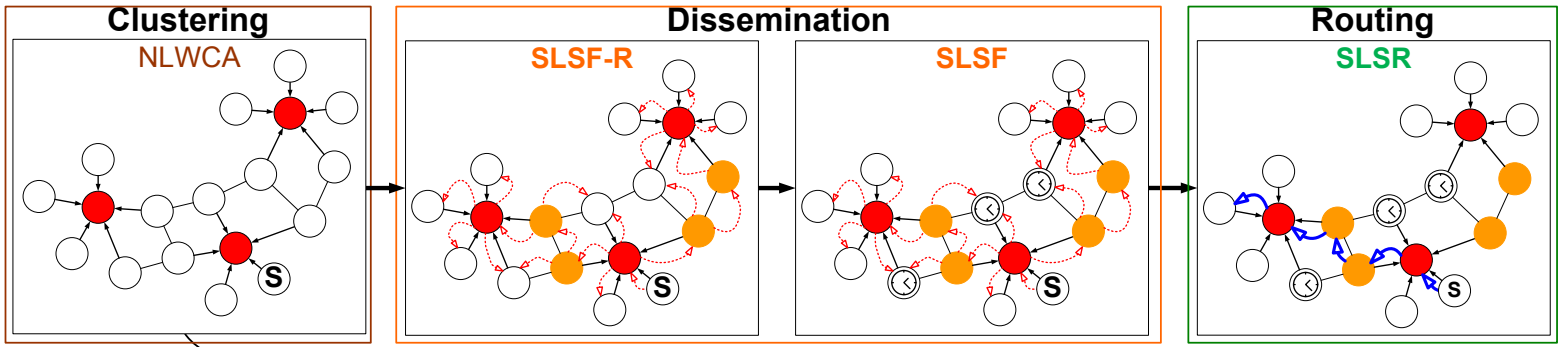
- Stability, Reference nodes
- Efficient message dissemination
- Routing

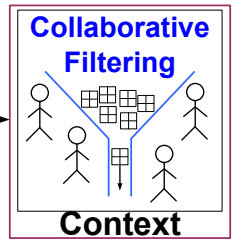


Collaborations :
A. Andronache, S. Rothkugel

Contributions for Advanced Service Discovery

- Stability, Reference nodes
- Efficient message dissemination
- Routing
- Context awareness

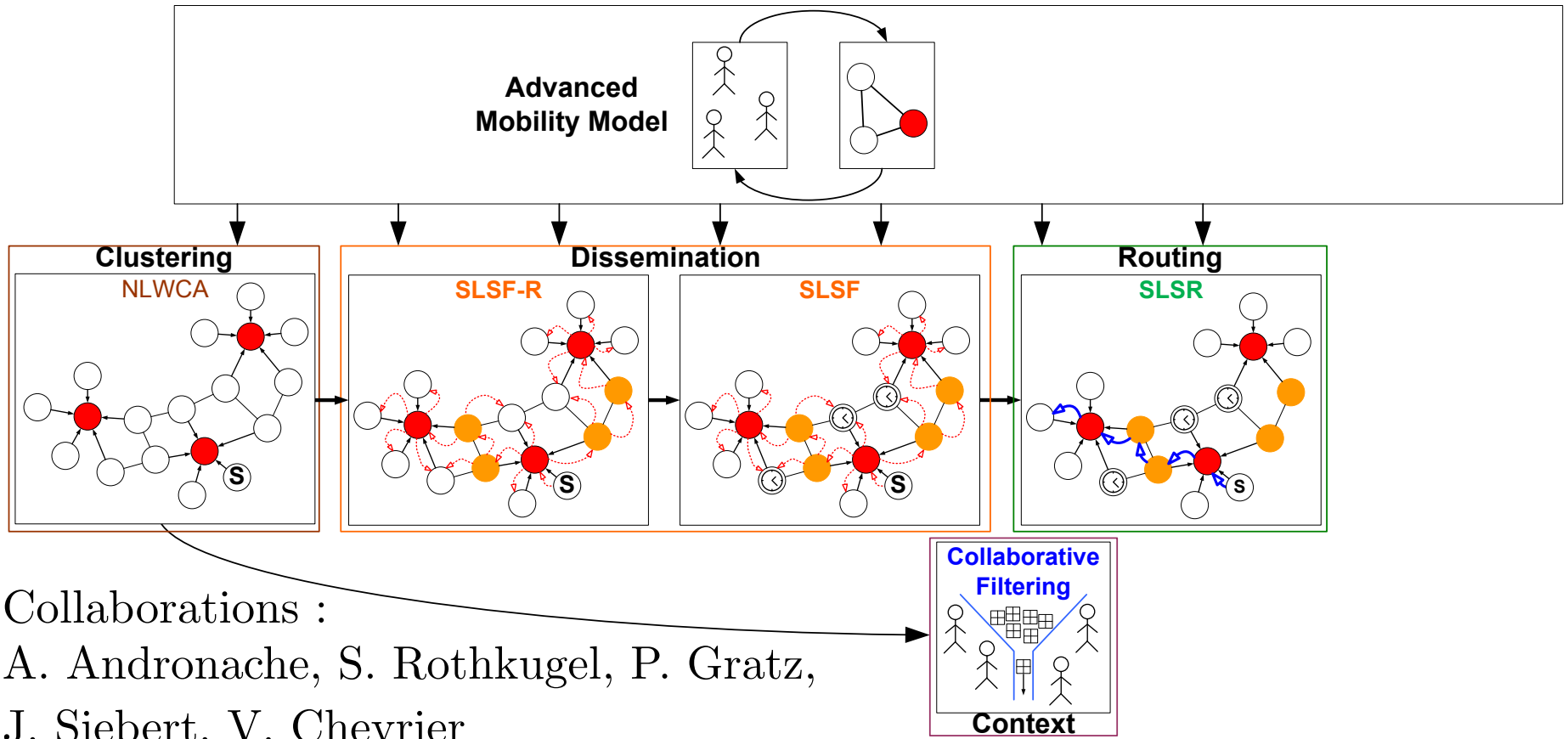




Collaborations :
A. Andronache, S. Rothkugel, P. Gratz

Contributions for Advanced Service Discovery

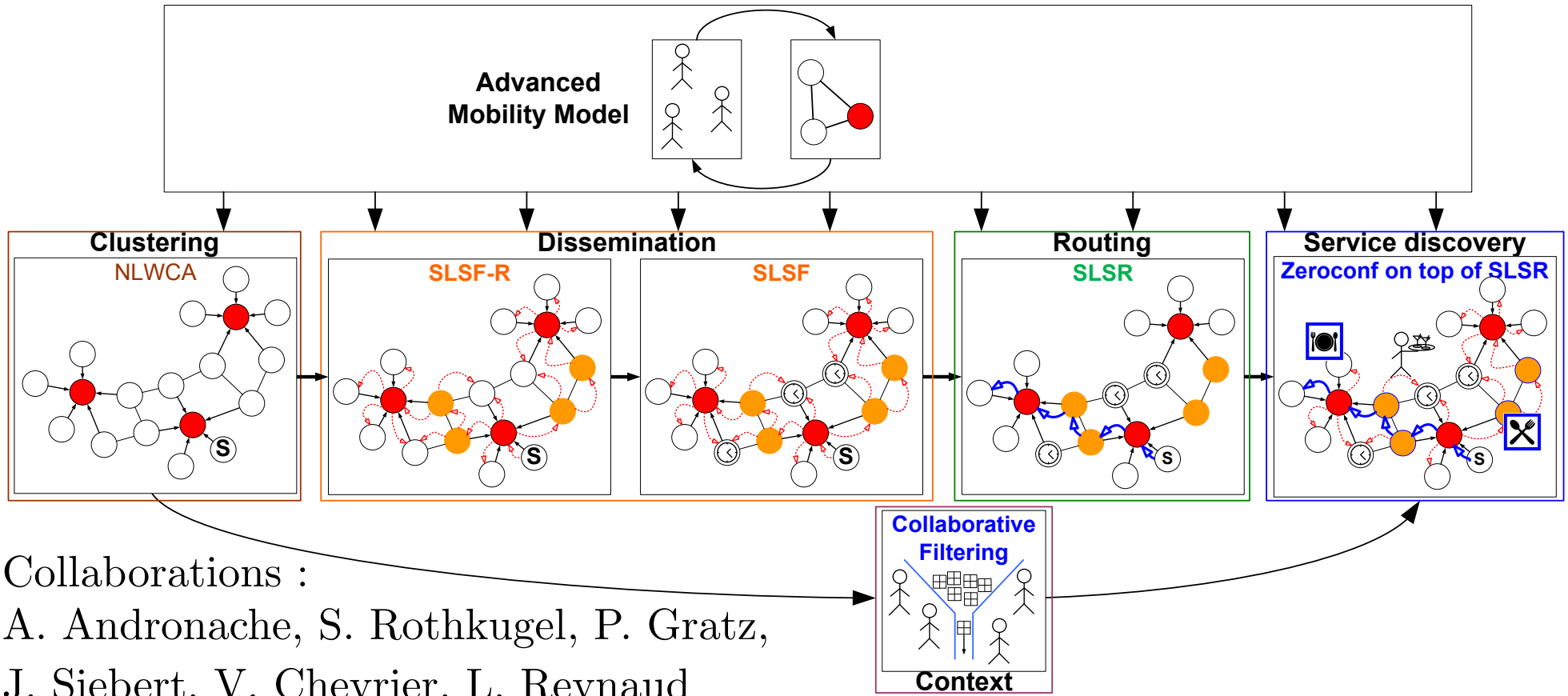
- Stability, Reference nodes
- Efficient message dissemination
- Routing
- Context awareness
- Mobility models, Research tools
- Experiments (Orange Labs)



Collaborations :
A. Andronache, S. Rothkugel, P. Gratz,
J. Siebert, V. Chevrier

Contributions for Advanced Service Discovery

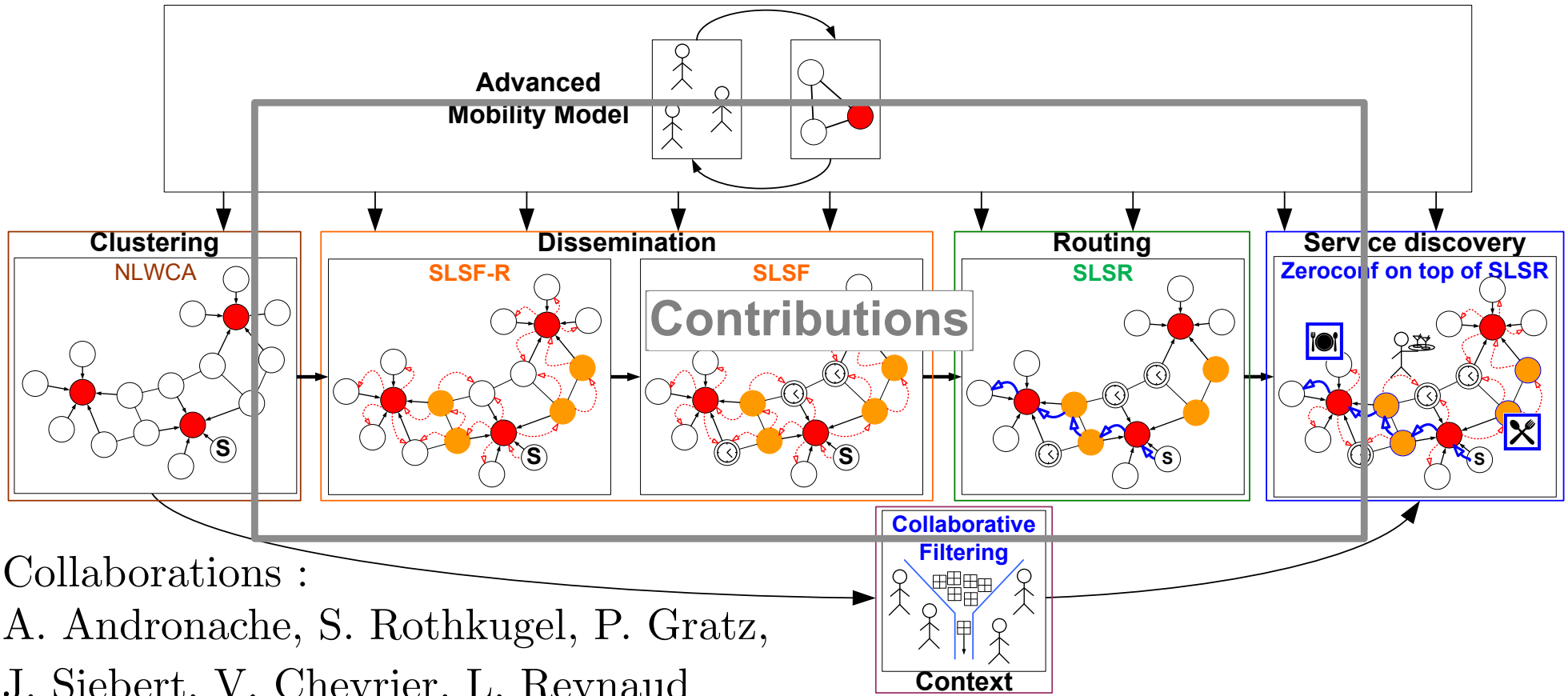
- Stability, Reference nodes
- Efficient message dissemination
- Routing
- Context awareness
- Mobility models, Research tools
- Experiments (Orange Labs)



Collaborations :
A. Andronache, S. Rothkugel, P. Gratz,
J. Siebert, V. Chevrier, L. Reynaud

Contributions for Advanced Service Discovery

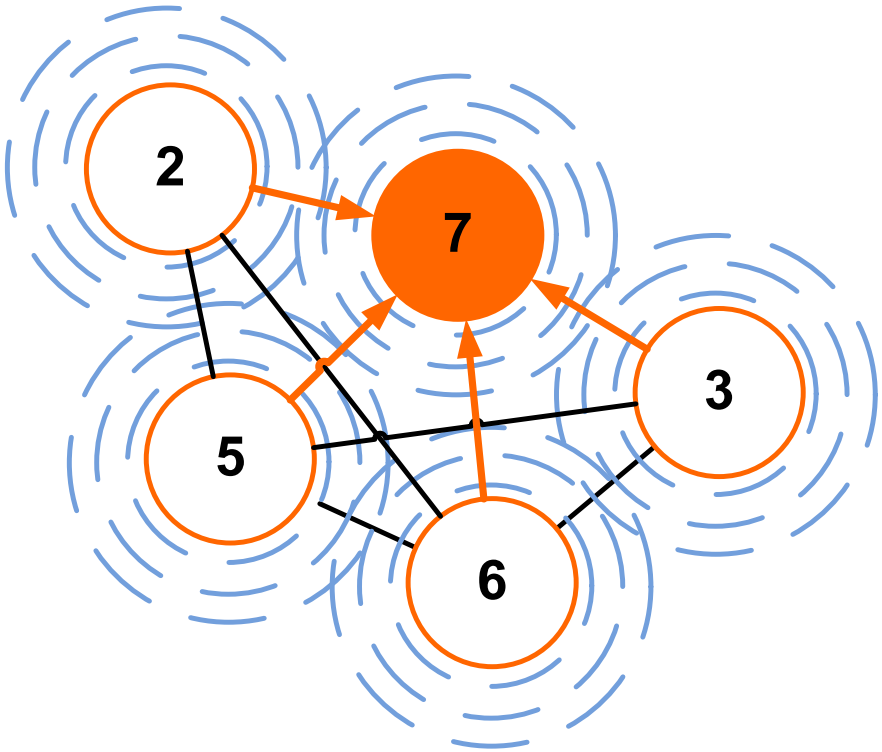
- Stability, Reference nodes
- Efficient message dissemination
- Routing
- Context awareness
- Mobility models, Research tools
- Experiments (Orange Labs)



Collaborations :
A. Andronache, S. Rothkugel, P. Gratz,
J. Siebert, V. Chevrier, L. Reynaud

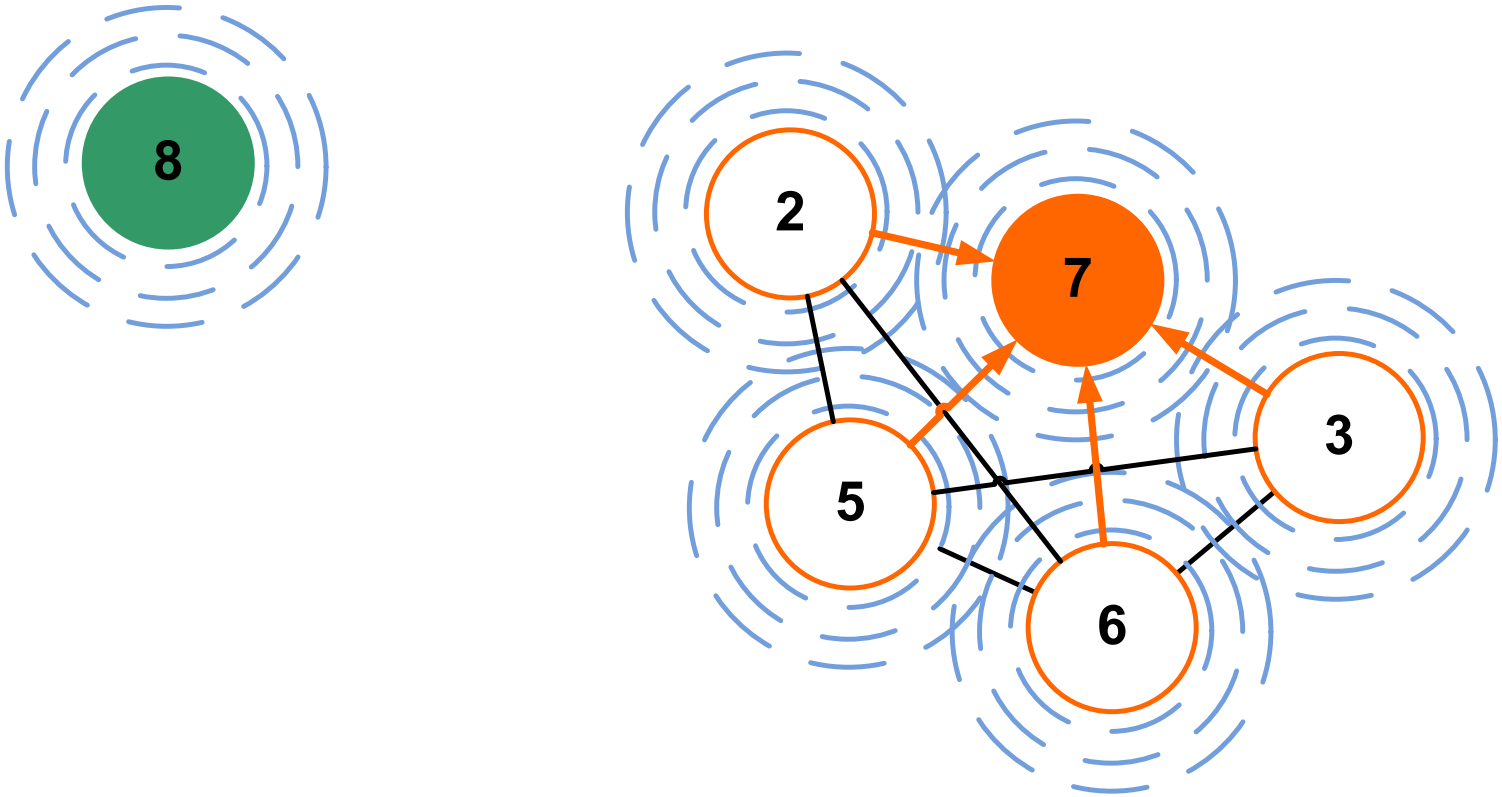
NLWCA[Andr 08b]

- Constructs one-hop clusters
  - Uses beacons to advertise its node weight
  - Clusterhead is the neighbor with highest node weight



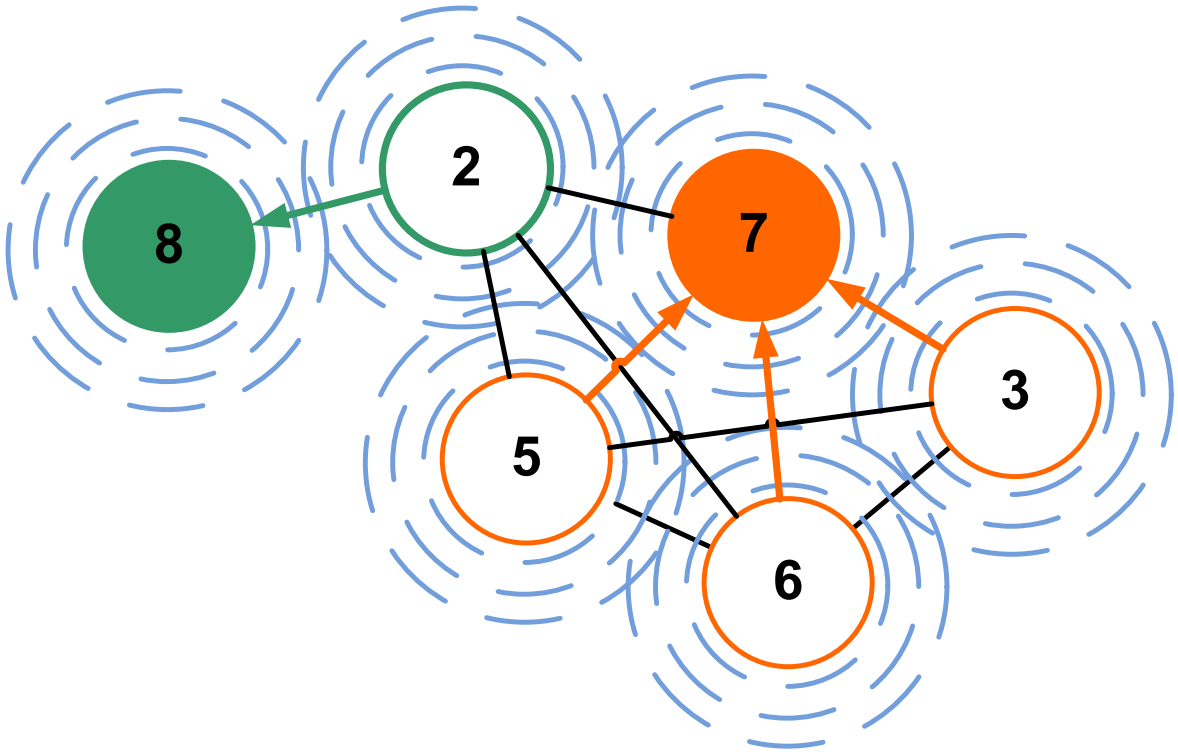
NLWCA[Andr 08b]

- Constructs one-hop clusters
- Assigns weights to the links
  - Reaching a given threshold the link is considered stable
  - Only stable neighbors are considered for clusterhead selection



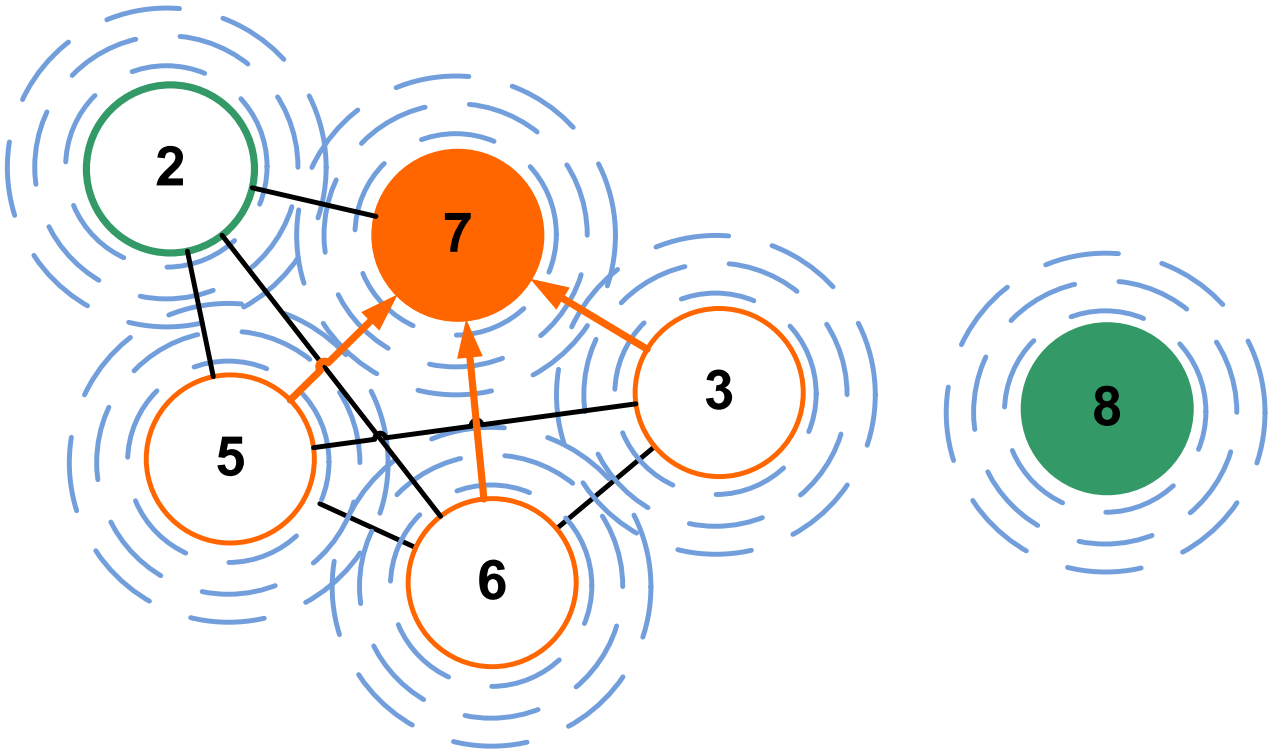
NLWCA[Andr 08b]

- Constructs one-hop clusters
- Assigns weights to the links
  - Reaching a given threshold the link is considered stable
  - Only stable neighbors are considered for clusterhead selection



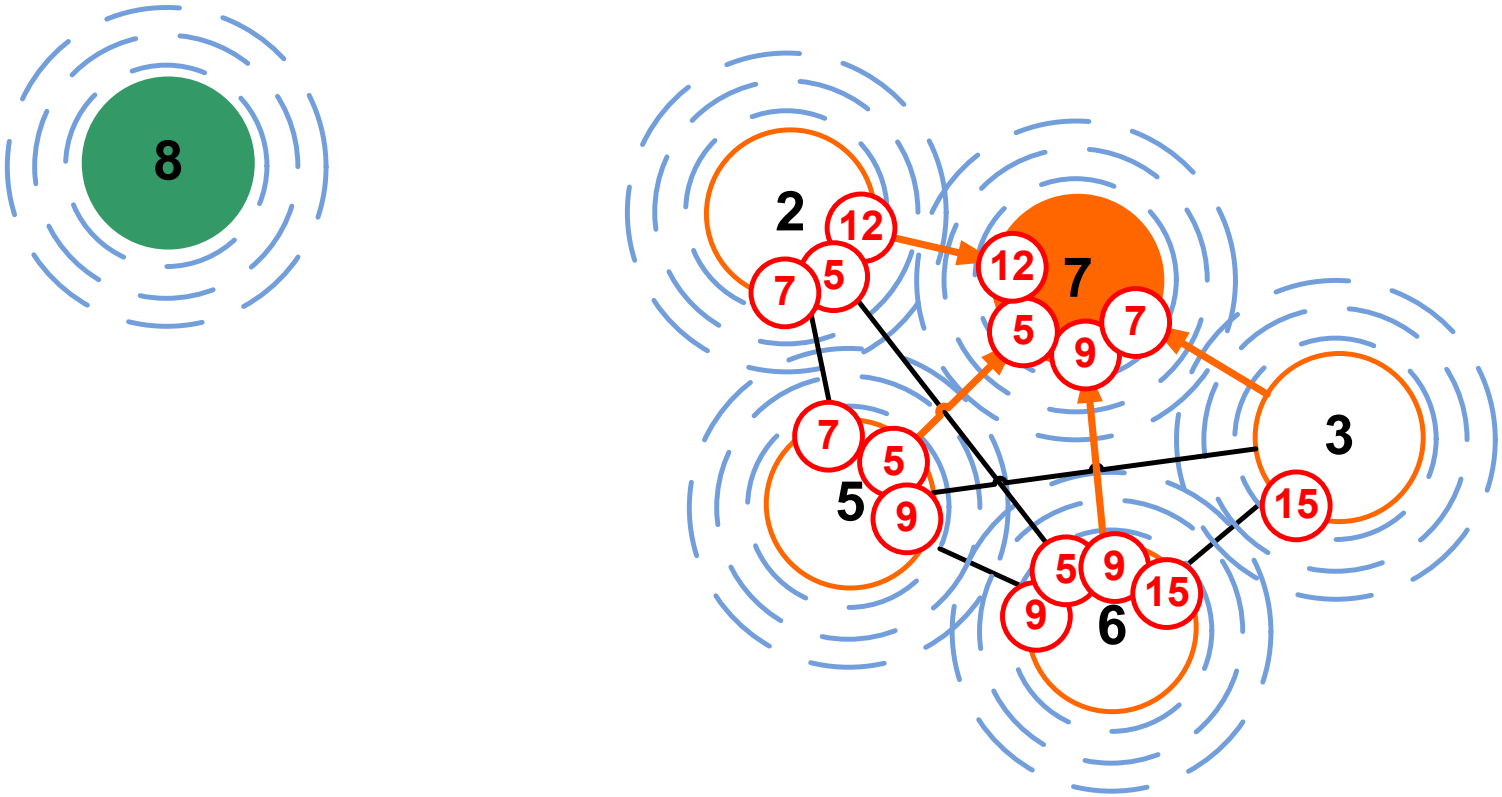
NLWCA[Andr 08b]

- Constructs one-hop clusters
- Assigns weights to the links
  - Reaching a given threshold the link is considered stable
  - Only stable neighbors are considered for clusterhead selection



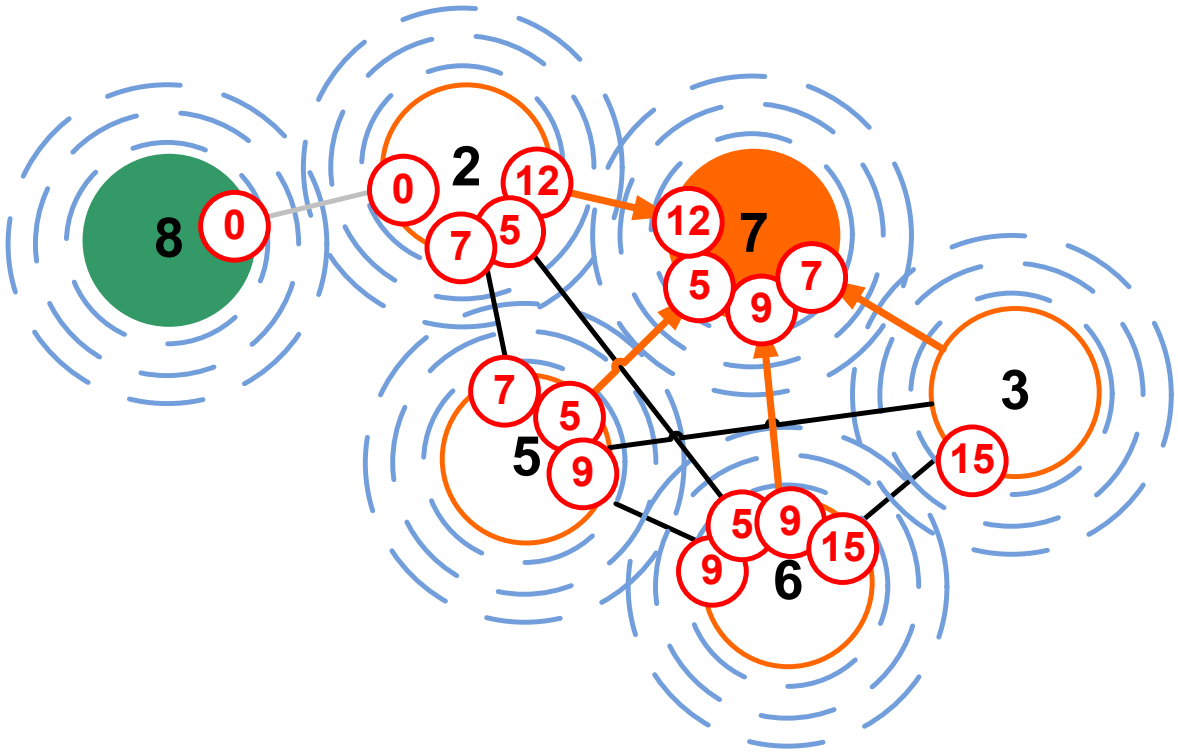
NLWCA[Andr 08b]

- Constructs one-hop clusters
- Assigns weights to the links
  - Reaching a given threshold the link is considered stable
  - Only stable neighbors are considered for clusterhead selection



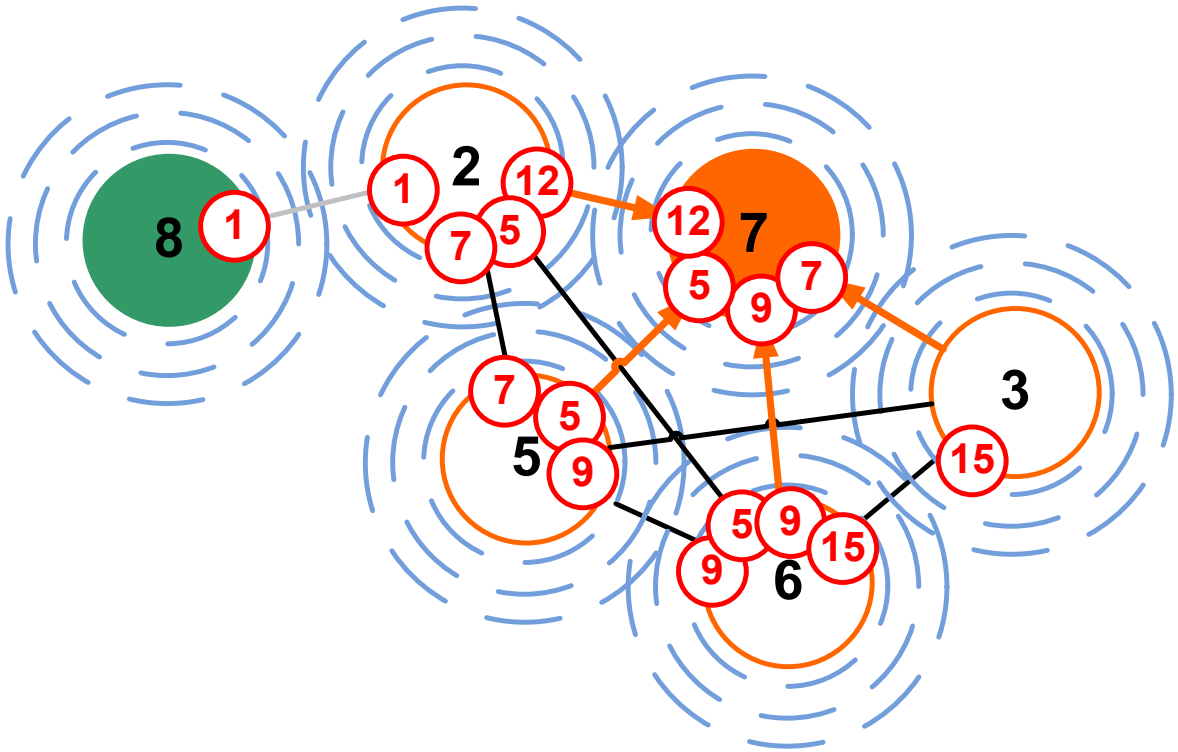
NLWCA[Andr 08b]

- Constructs one-hop clusters
- Assigns weights to the links
  - Reaching a given threshold the link is considered stable
  - Only stable neighbors are considered for clusterhead selection



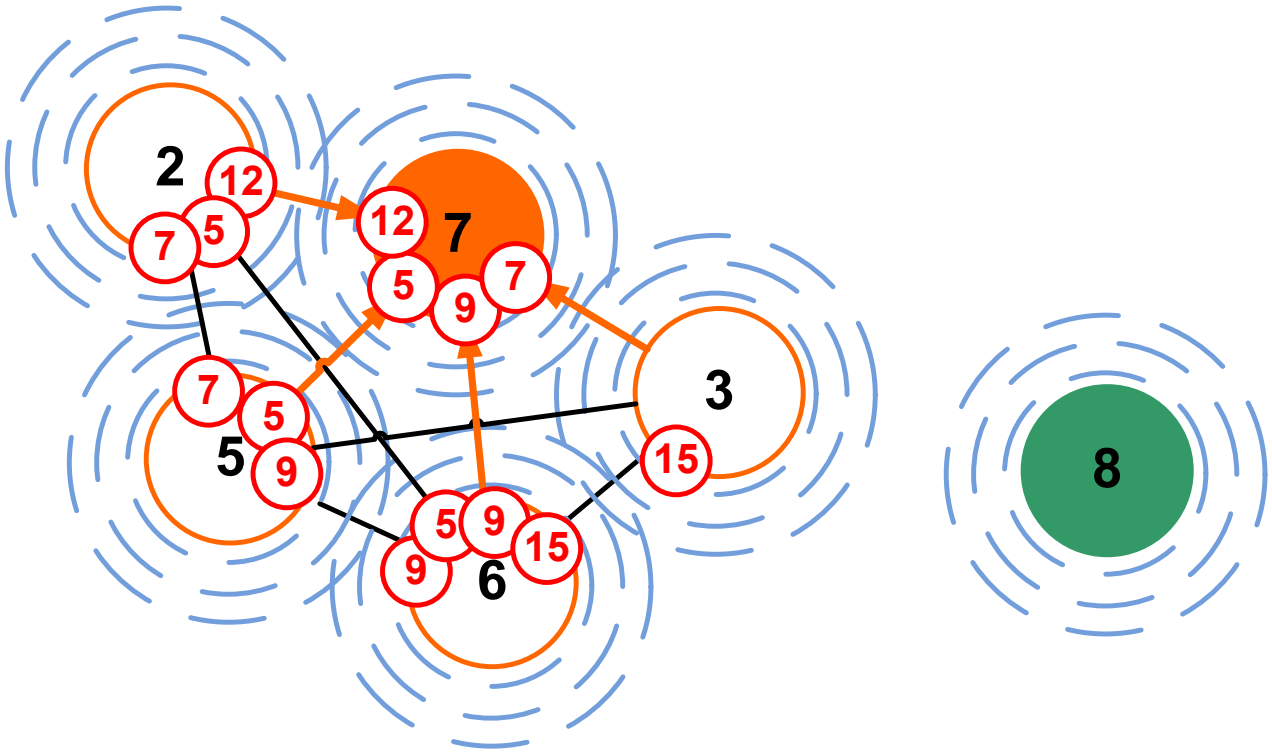
NLWCA[Andr 08b]

- Constructs one-hop clusters
- Assigns weights to the links
  - Reaching a given threshold the link is considered stable
  - Only stable neighbors are considered for clusterhead selection



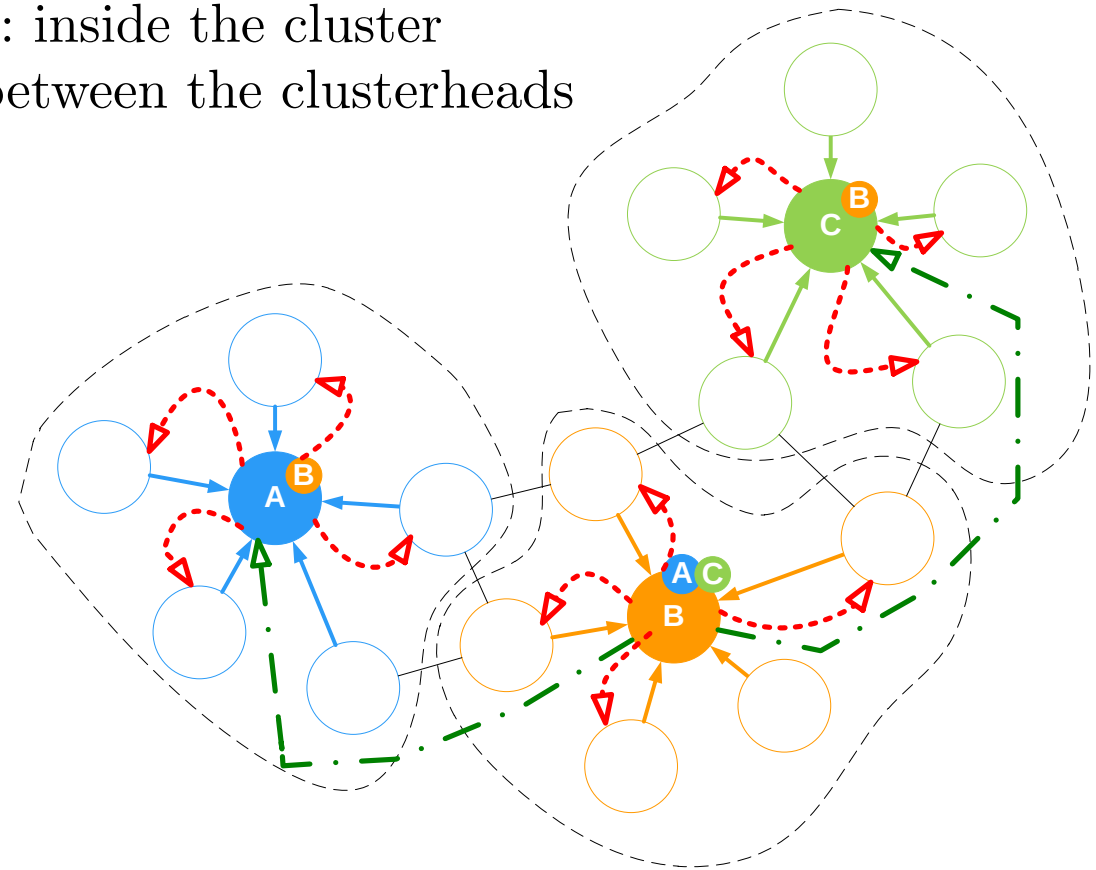
NLWCA[Andr 08b]

- Constructs one-hop clusters
- Assigns weights to the links
  - Reaching a given threshold the link is considered stable
  - Only stable neighbors are considered for clusterhead selection



WCPD[Andr 08a]

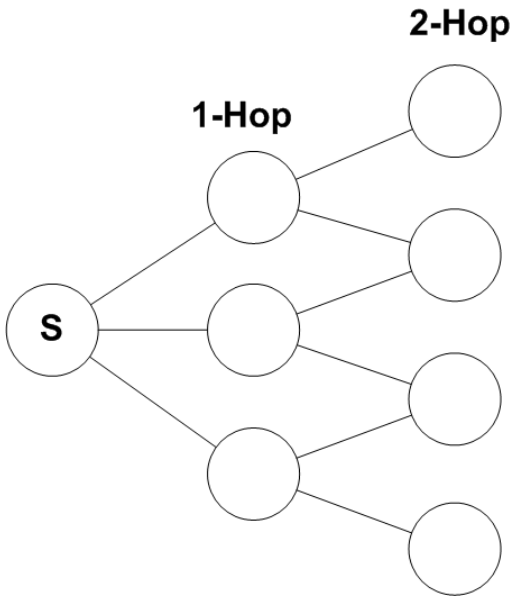
- Is on top of NLWCA
- Provides dissemination capabilities
- Disseminate a message
  - Broadcast : inside the cluster
  - Unicast : between the clusterheads



OLSR[Clau 03]

Dissemination in OLSR

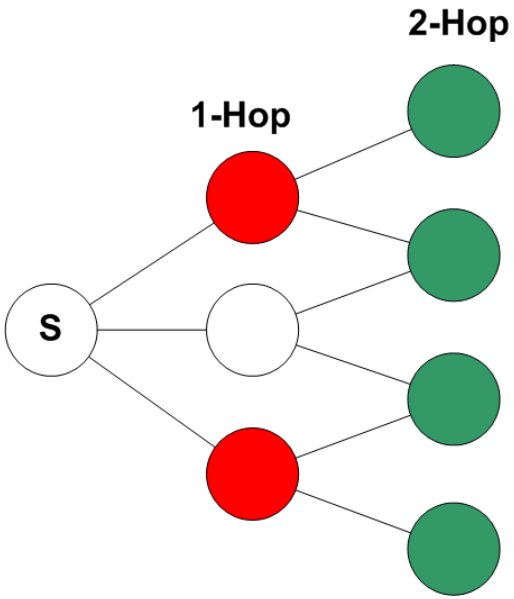
- Exchanges HELLO messages for 2-hop neighborhood discovery
- MPR (Multipoint Relay) : Smallest set of 1-hop neighbors to reach all 2-hop neighbors



OLSR[Clau 03]

Dissemination in OLSR

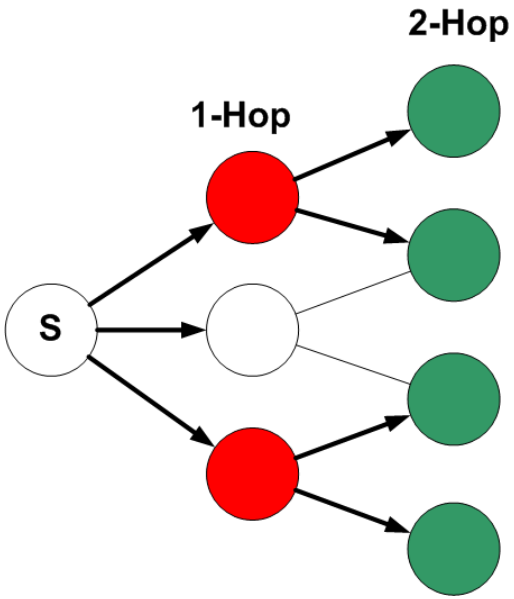
- Exchanges HELLO messages for 2-hop neighborhood discovery
- MPR (Multipoint Relay) : Smallest set of 1-hop neighbors to reach all 2-hop neighbors



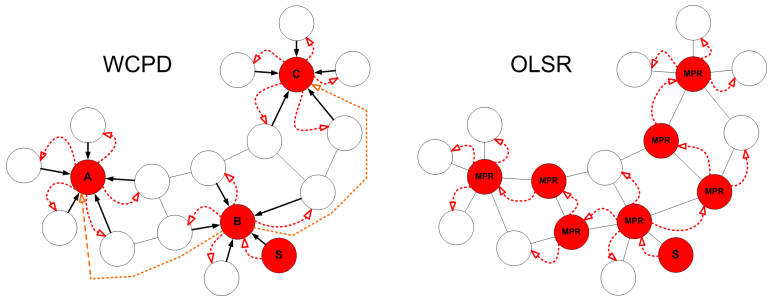
OLSR[Clau 03]

Dissemination in OLSR

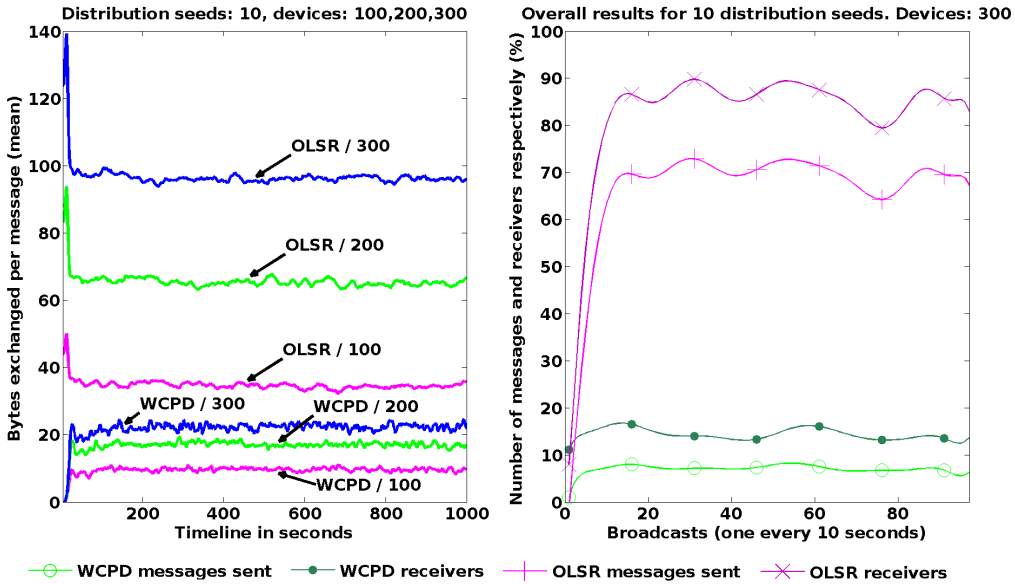
- Exchanges HELLO messages for 2-hop neighborhood discovery
- MPR (Multipoint Relay) : Smallest set of 1-hop neighbors to reach all 2-hop neighbors



Why this structure ? [Lecl 08]

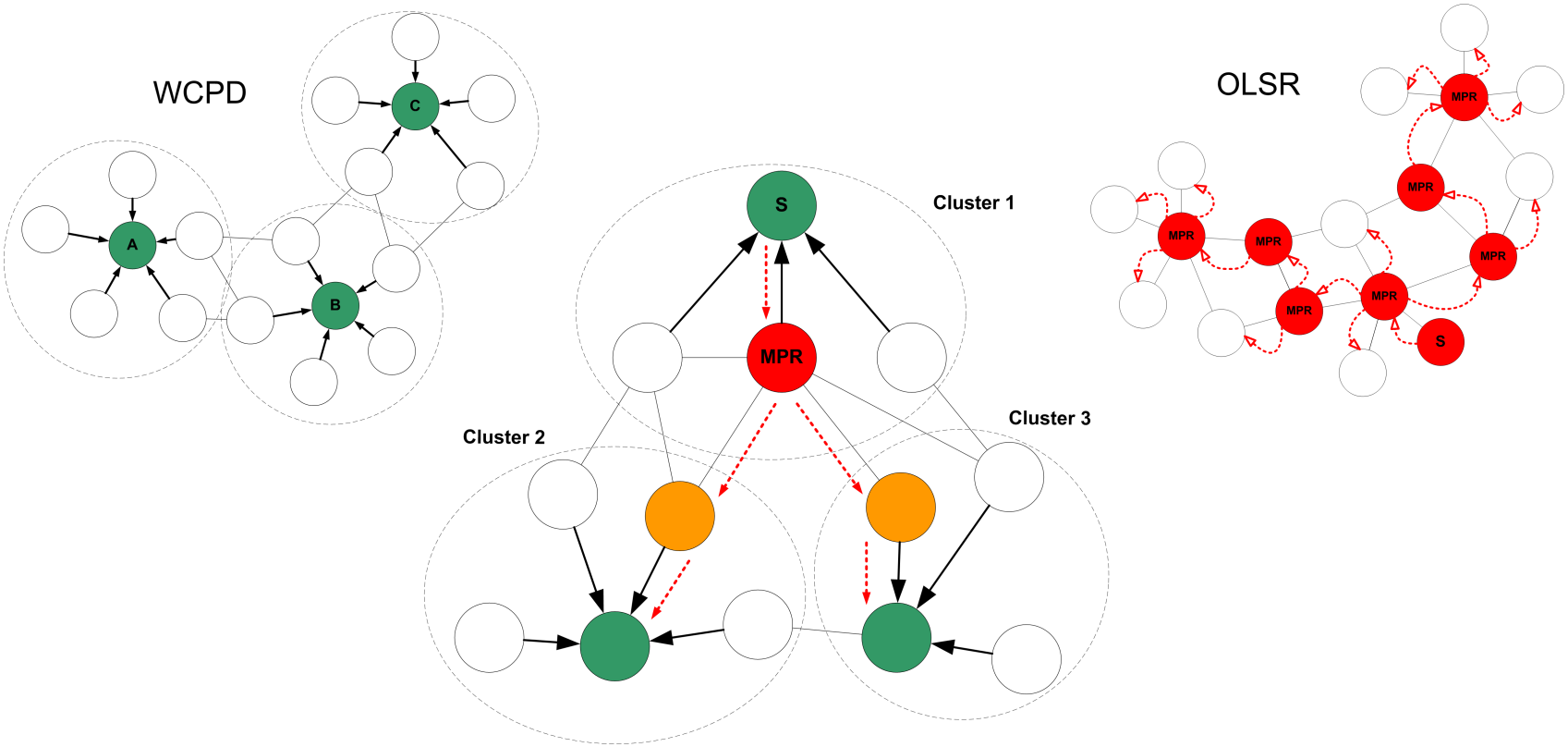


- OLSR uses high bandwidth but provides good reachability
- WCPD reaches few nodes but uses very low bandwidth



Why this structure ? [Lecl 08]

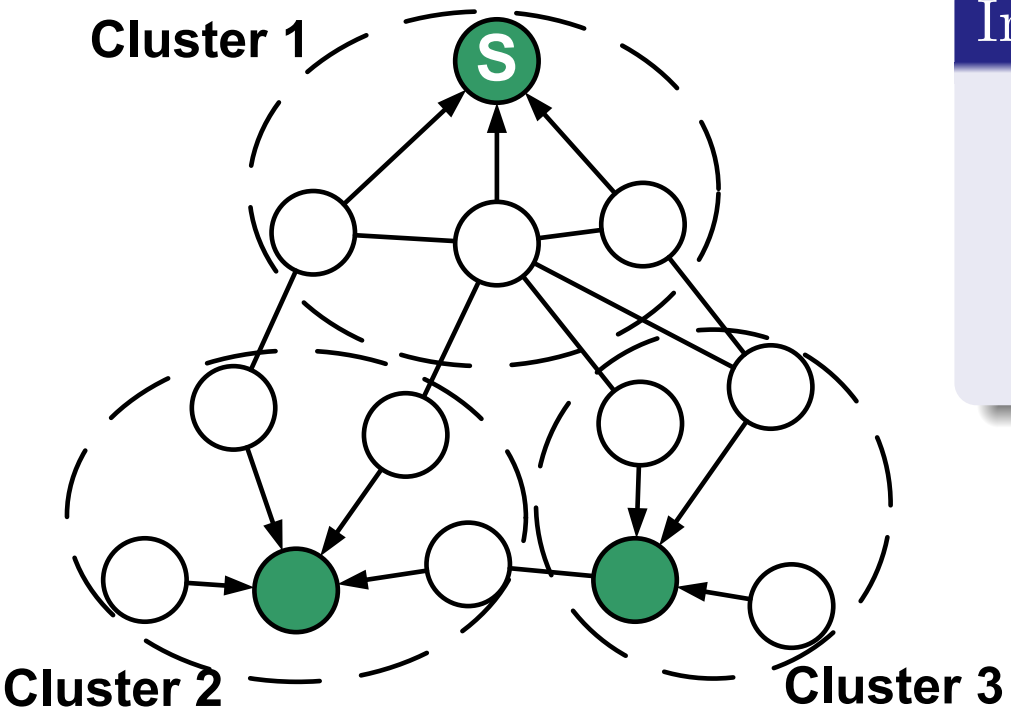
⇛ Idea : Combine OLSR and NLWCA



SLSF[Lecl 10b]

Use NLWCA as stable structure

- Reduce the number of forwarding nodes
- Reduce the structure setup overhead



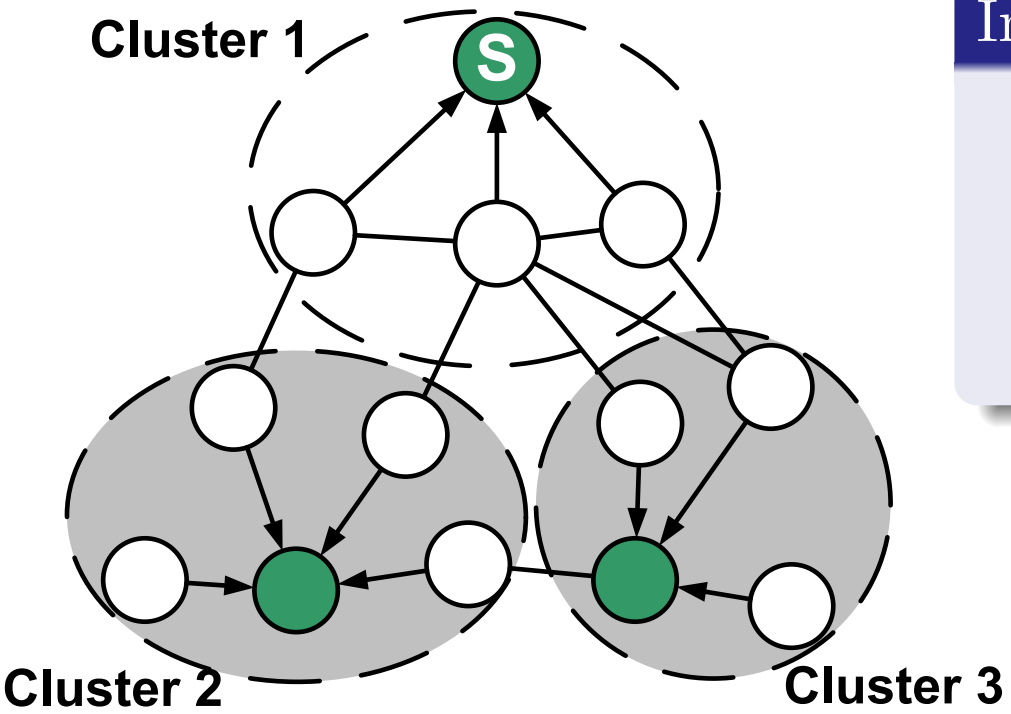
Inter Cluster Relay nodes (ICR)

- Smallest set of neighbors that reach all nearby clusterheads
- Only ICR nodes forward messages

SLSF[Lecl 10b]

Use NLWCA as stable structure

- Reduce the number of forwarding nodes
- Reduce the structure setup overhead



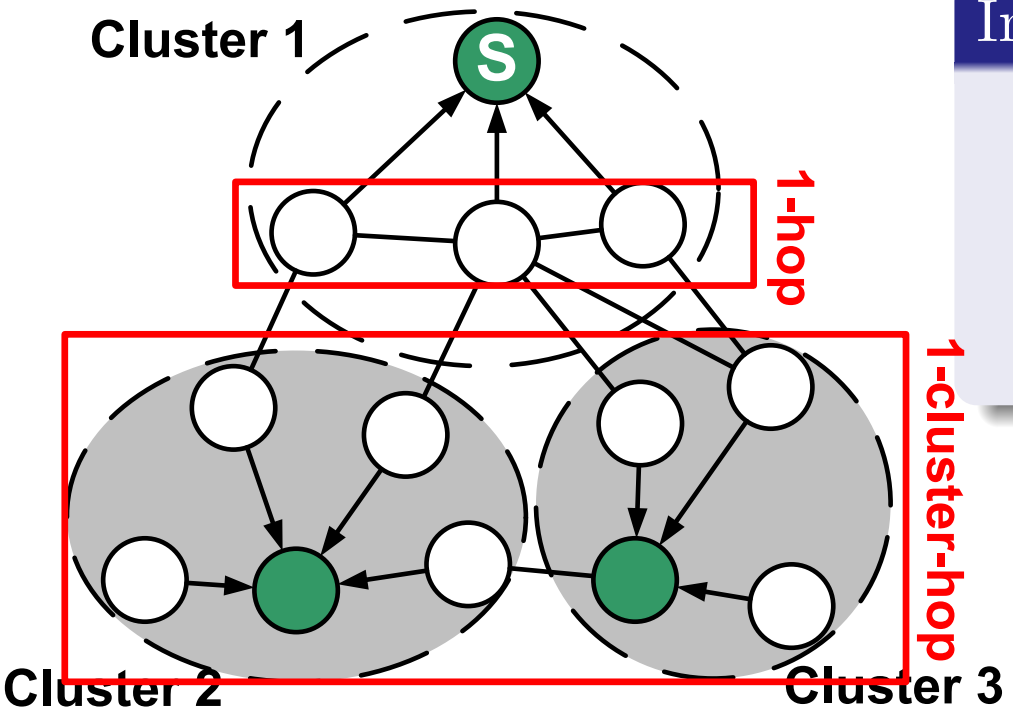
Inter Cluster Relay nodes (ICR)

- Smallest set of neighbors that reach all nearby clusterheads
- Only ICR nodes forward messages

SLSF[Lecl 10b]

Use NLWCA as stable structure

- Reduce the number of forwarding nodes
- Reduce the structure setup overhead



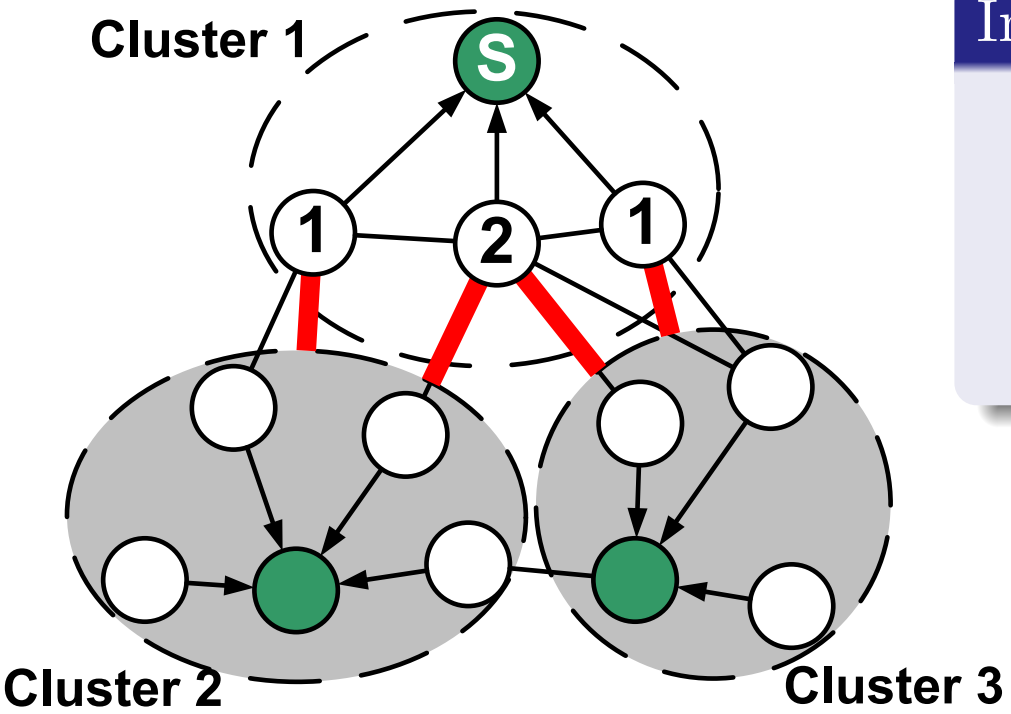
Inter Cluster Relay nodes (ICR)

- Smallest set of neighbors that reach all nearby clusterheads
- Only ICR nodes forward messages

SLSF[Lecl 10b]

Use NLWCA as stable structure

- Reduce the number of forwarding nodes
- Reduce the structure setup overhead



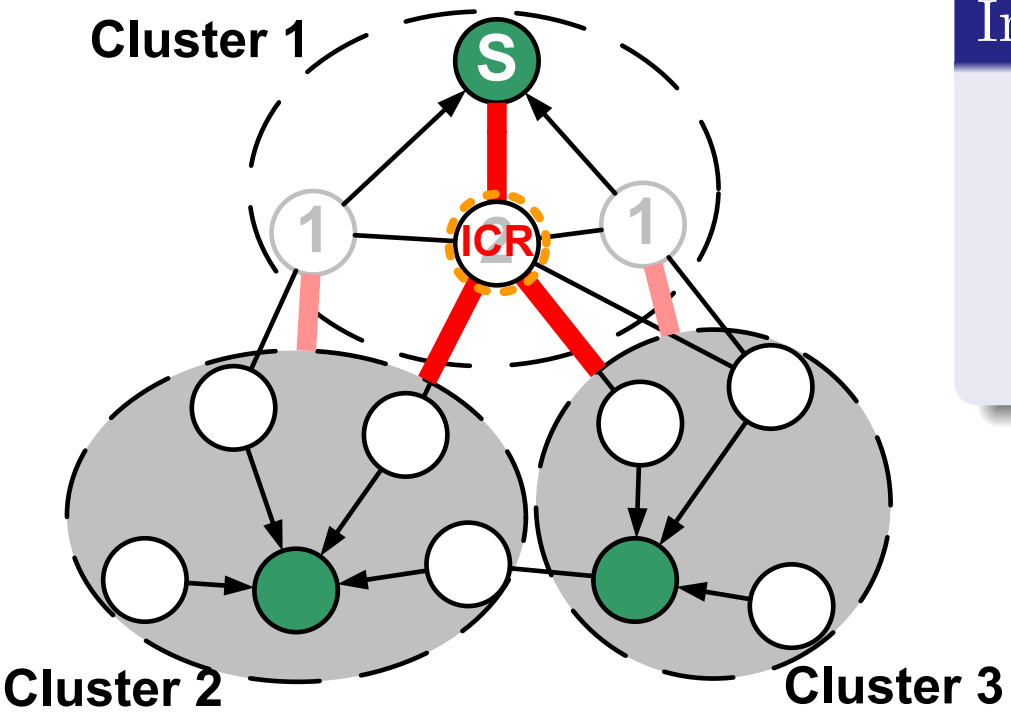
Inter Cluster Relay nodes (ICR)

- Smallest set of neighbors that reach all nearby clusterheads
- Only ICR nodes forward messages

SLSF[Lecl 10b]

Use NLWCA as stable structure

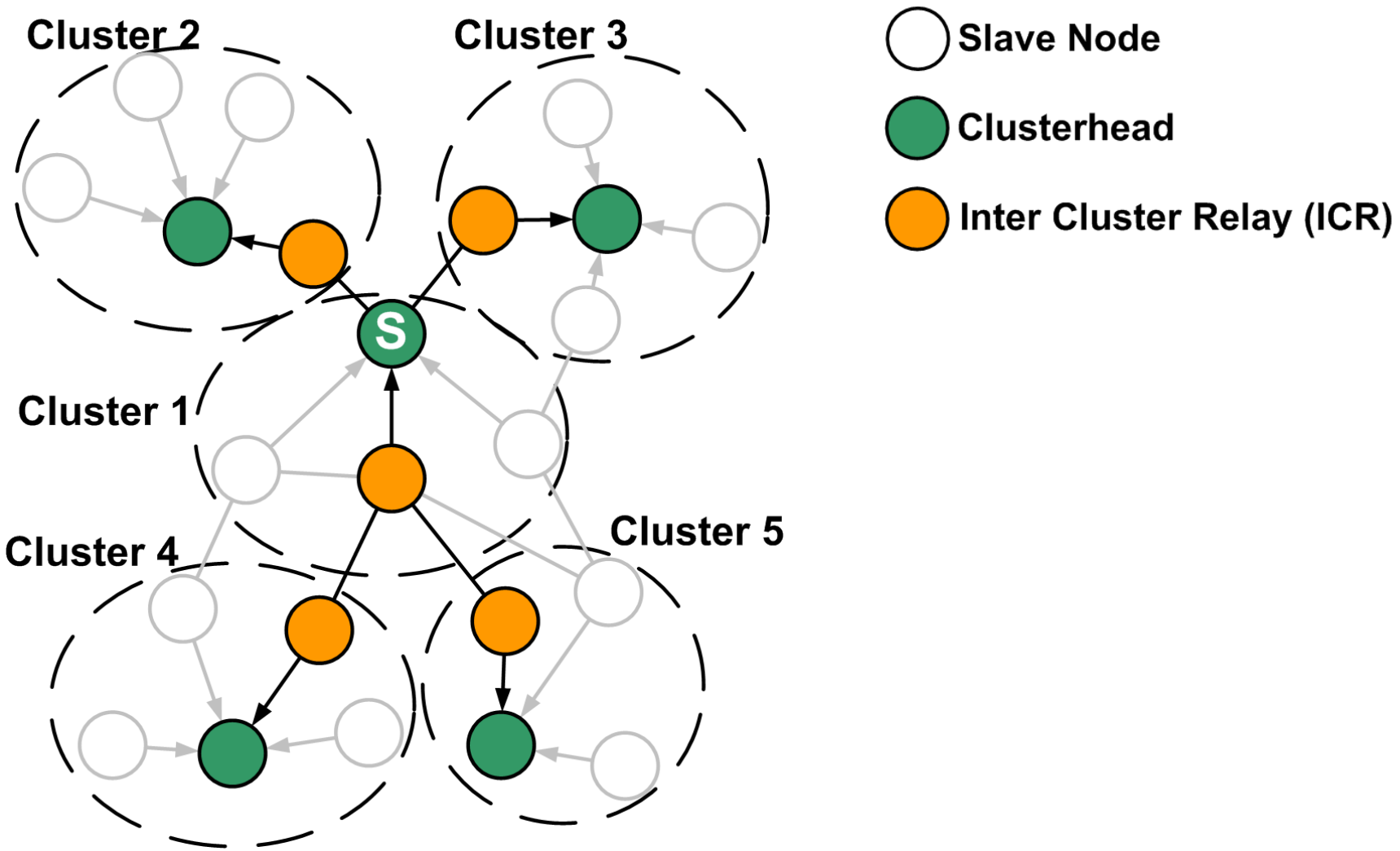
- Reduce the number of forwarding nodes
- Reduce the structure setup overhead



Inter Cluster Relay nodes (ICR)

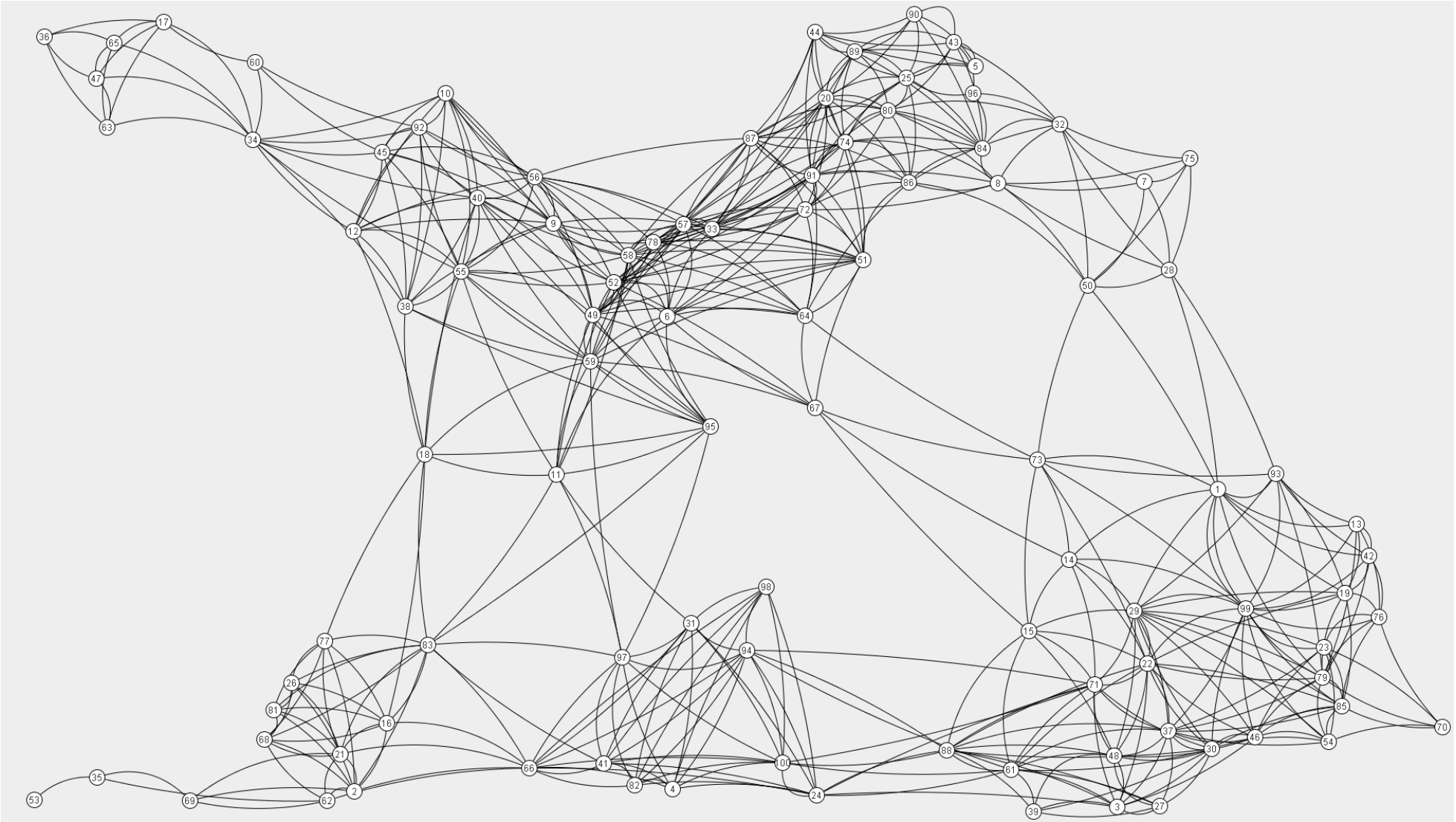
- Smallest set of neighbors that reach all nearby clusterheads
- Only ICR nodes forward messages

SLSF : Stable Linked Structure Flooding



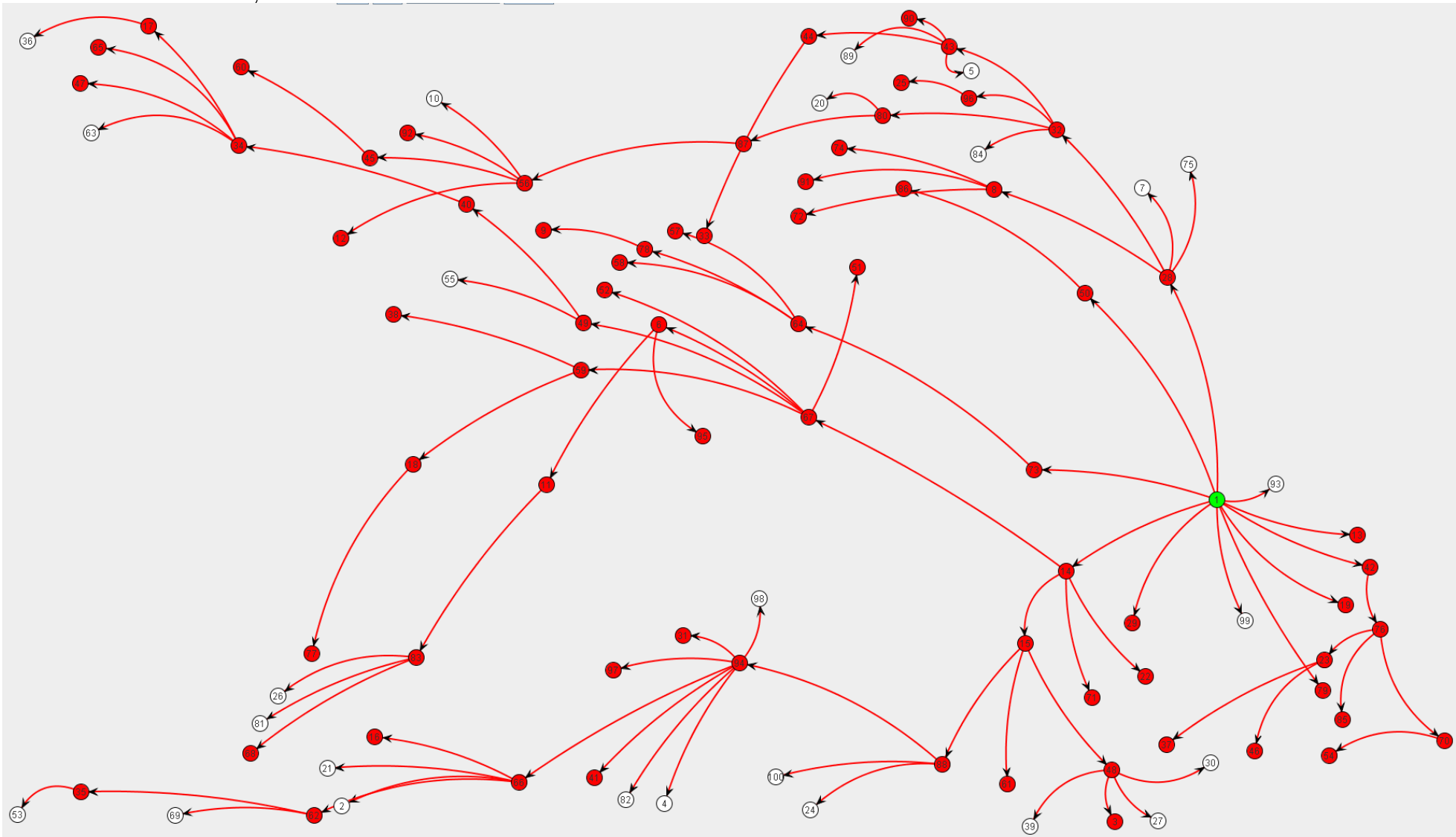
Static Simulation Results

Network with 100 nodes



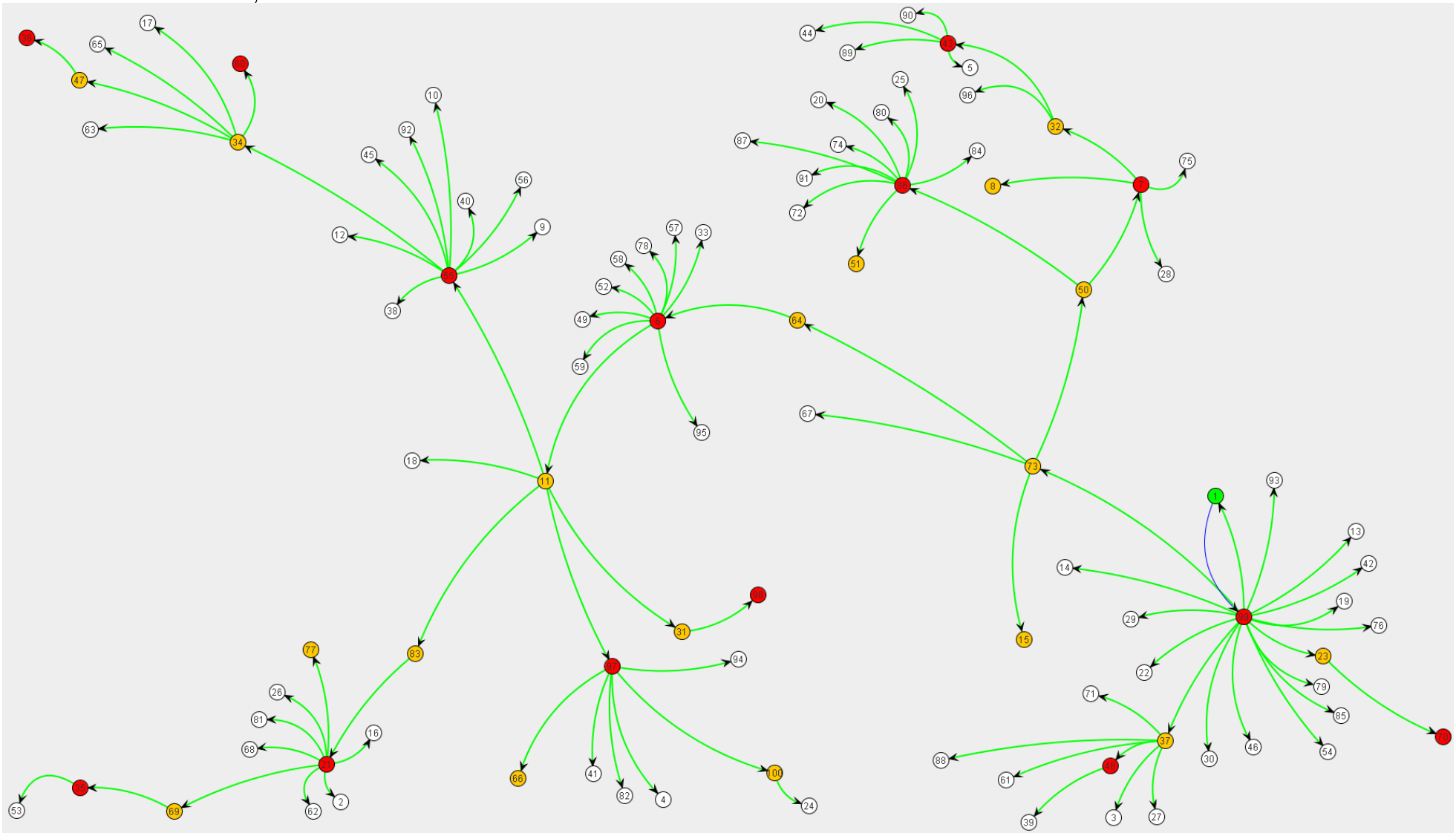
Static Simulation Results

100 nodes ; OLSR : 74 forwarders

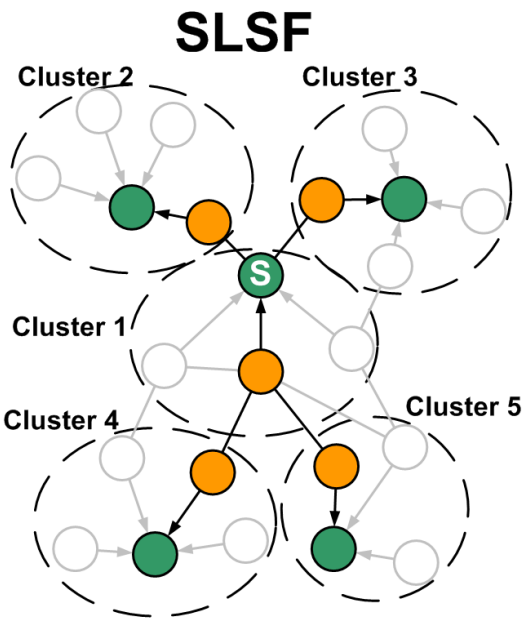


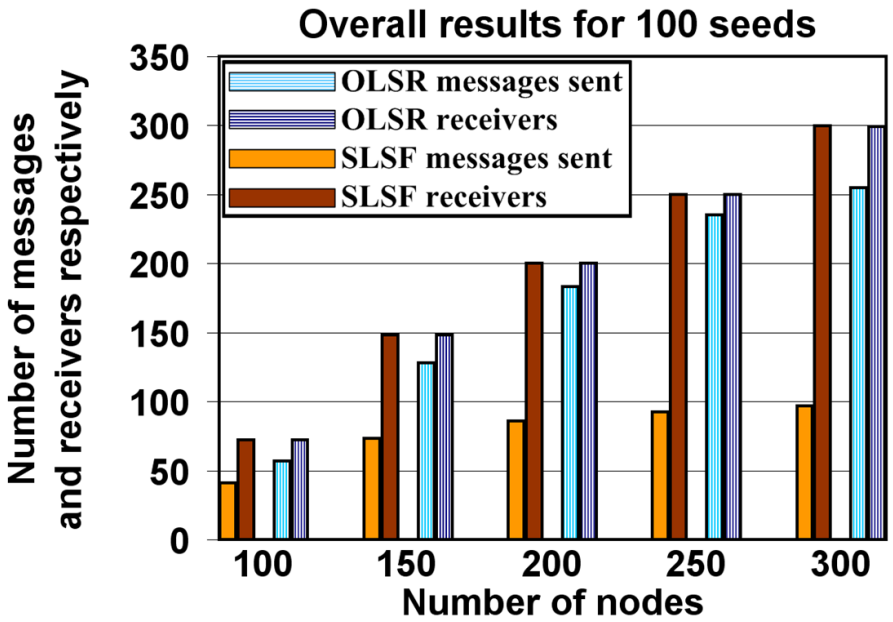
Static Simulation Results

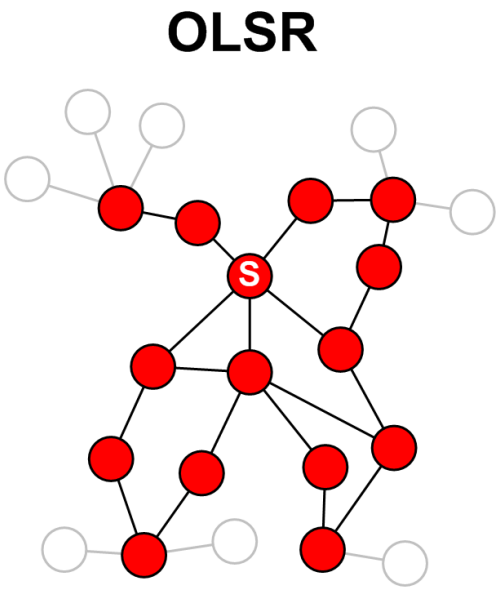
100 nodes ; SLSF : 33 forwarders



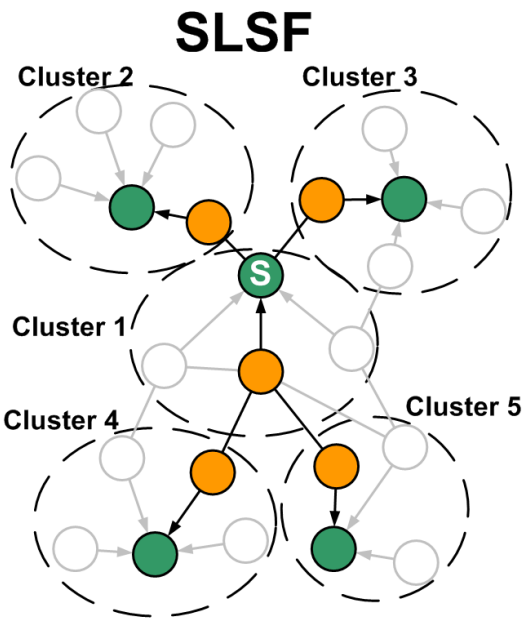
Static Simulation Results

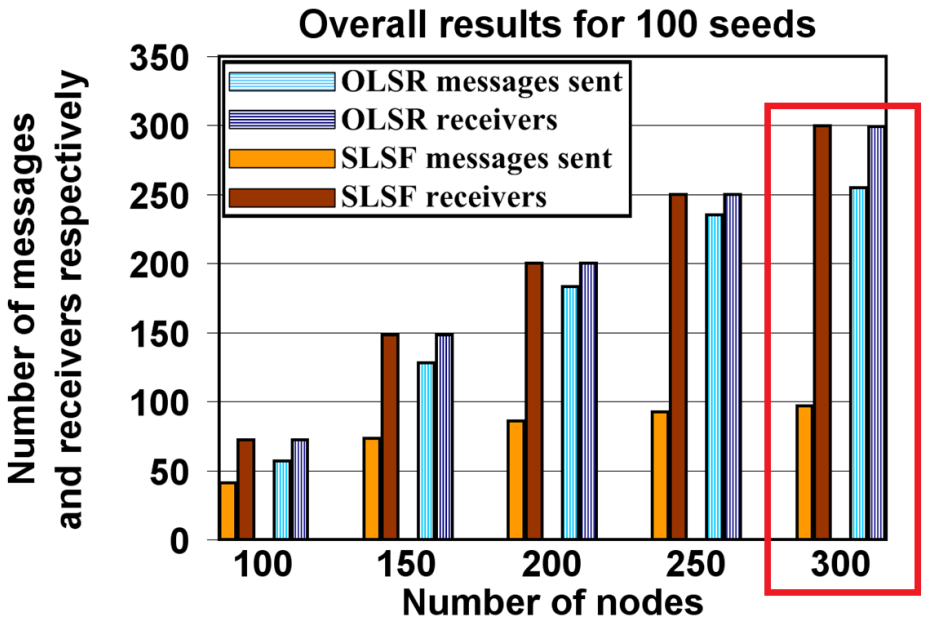


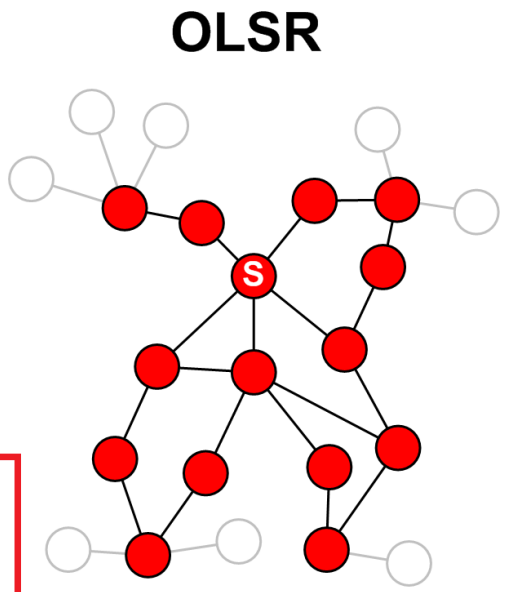




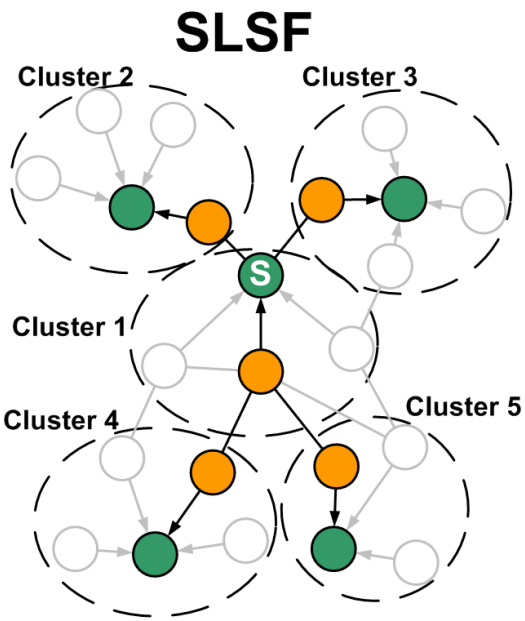
Static Simulation Results

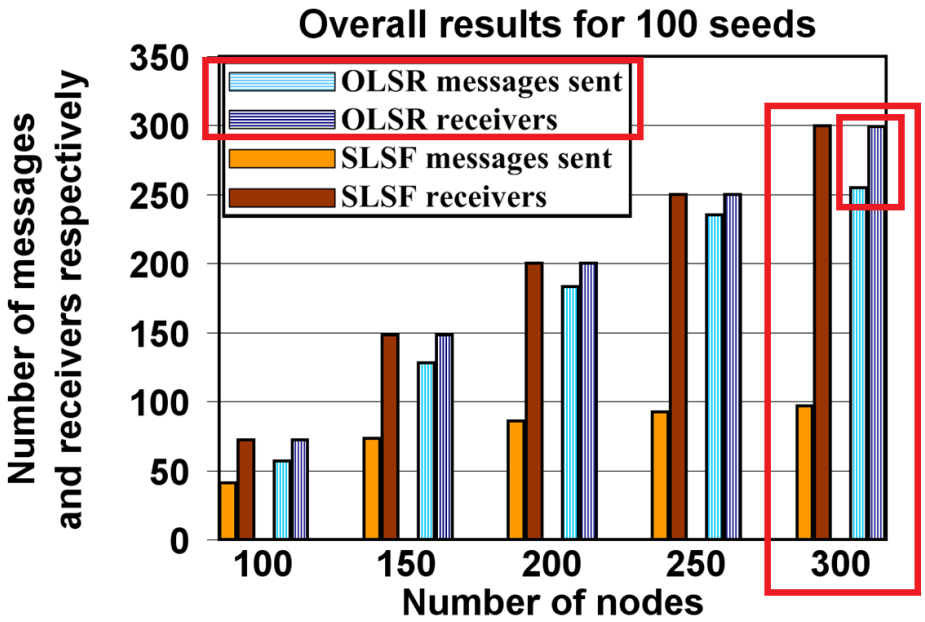


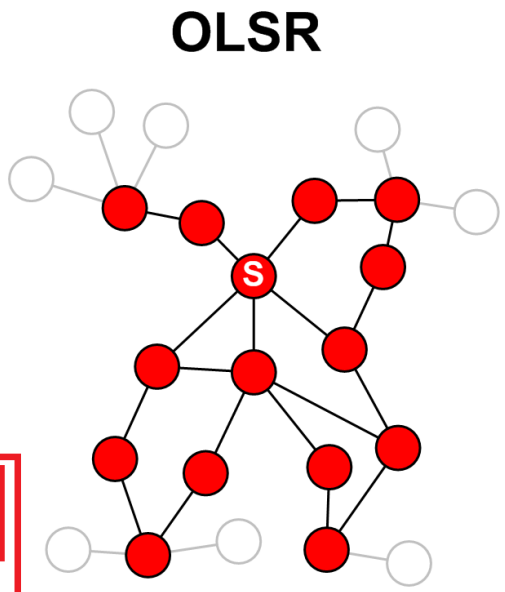




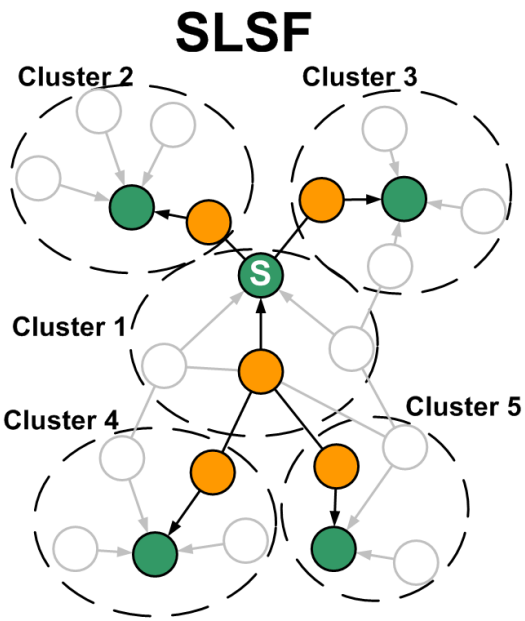
Static Simulation Results

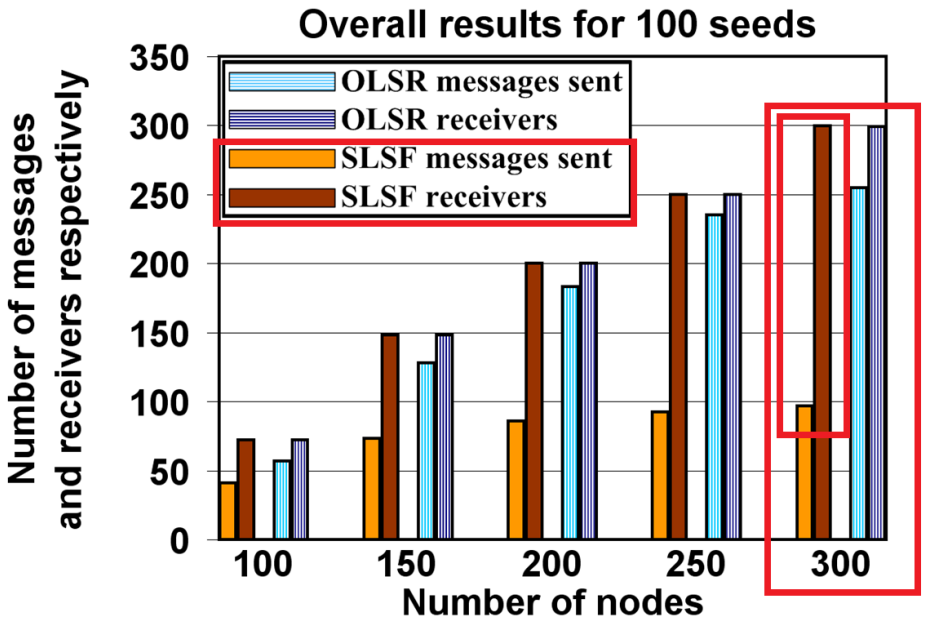


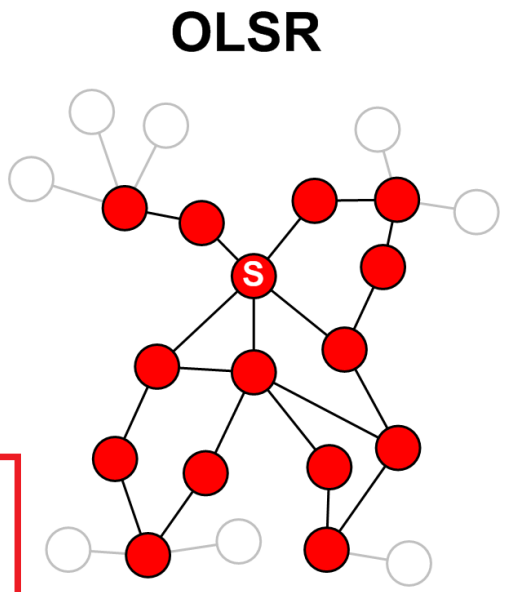




Static Simulation Results





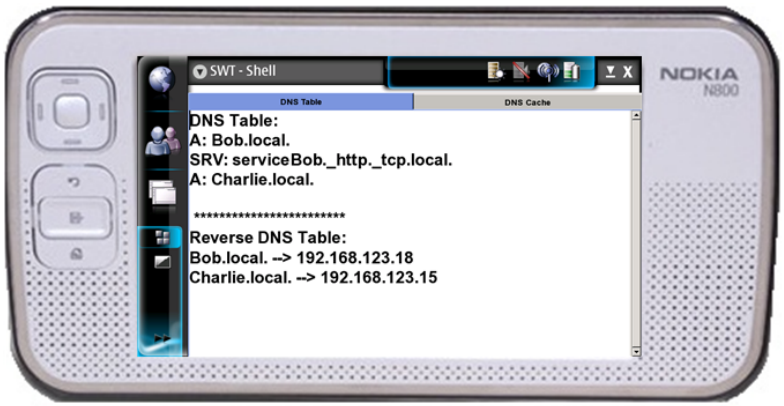


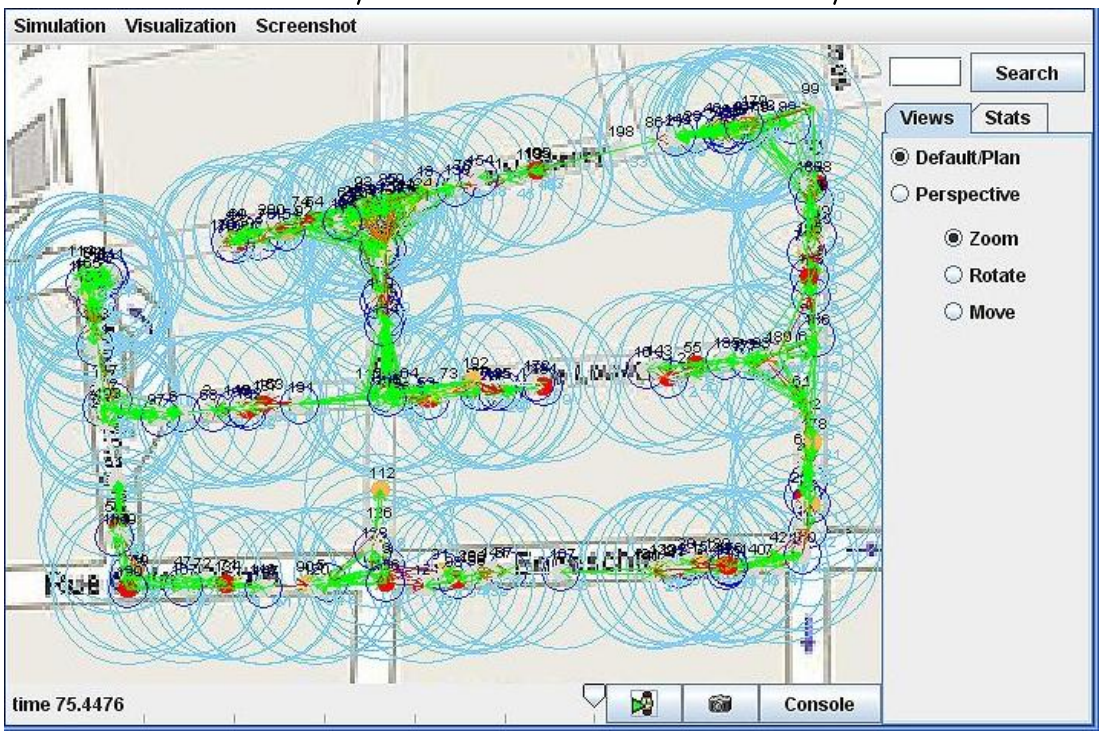
Simulations and settings

- JANE : Java Ad Hoc Network Development Environment [Gorg 07]
- Restricted random walk / 1000 seconds / 100-300 devices / 10 seeds

Different modes :

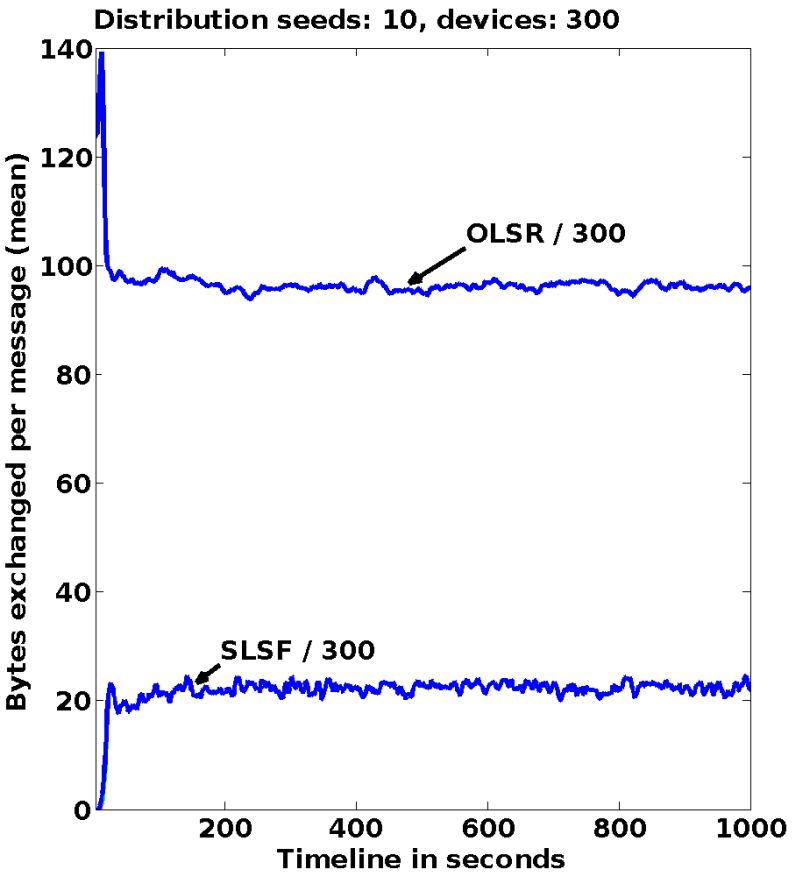
- Simulation
- Hybrid
- Platform



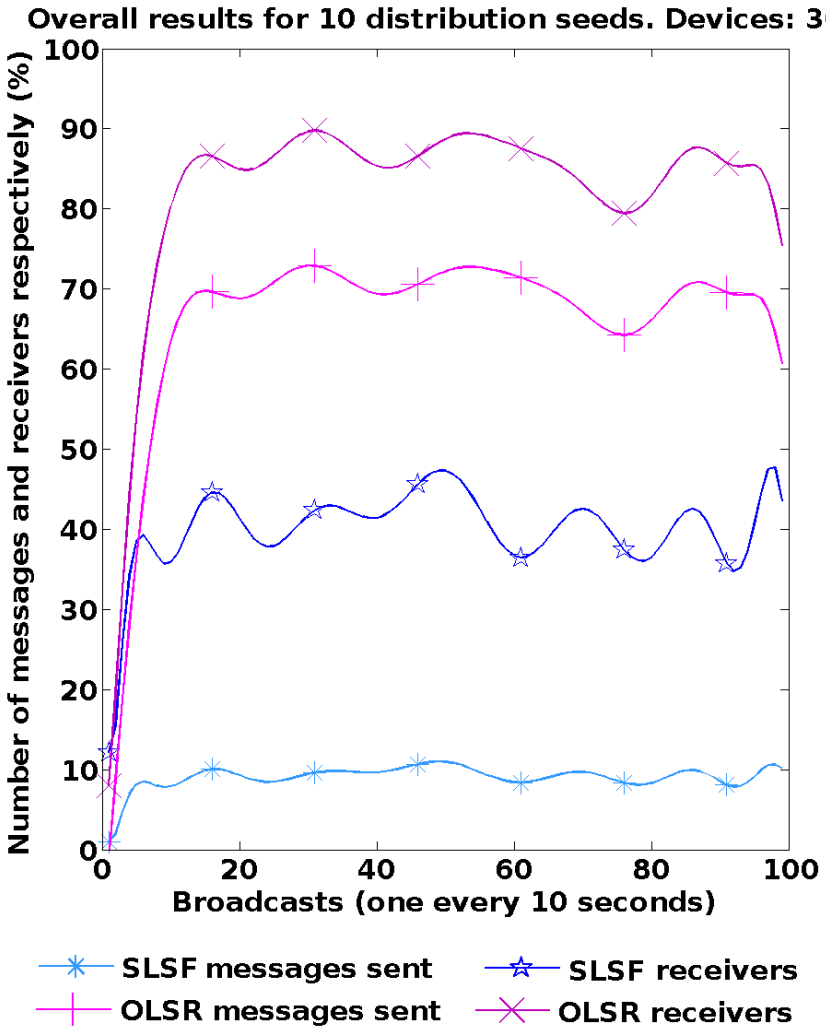


Dynamic Simulation Results

- Bandwidth usage



- Reachability



SLSF with Fault Recovery[Lecl 10c]

⇛ **Provide recovery only between nearby clusterheads**

Acknowledge messages between clusterheads

- Each message has a sequence number
- Beacon relay acknowledgements

Not ICR nodes :

- Await acknowledgements for messages
- React on lost messages
- Re-emit lost messages

SLSF with Fault Recovery Results

Restricted random walk / 1000 seconds / 100-300 devices / 10 seeds

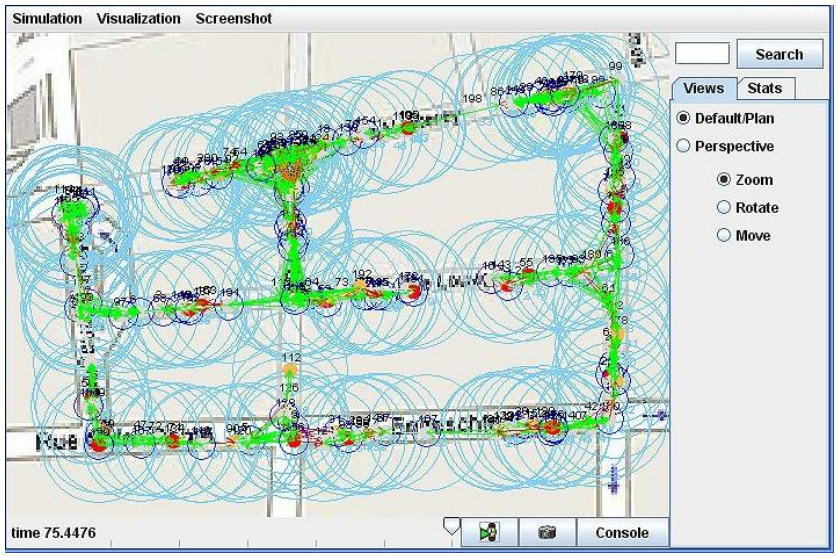
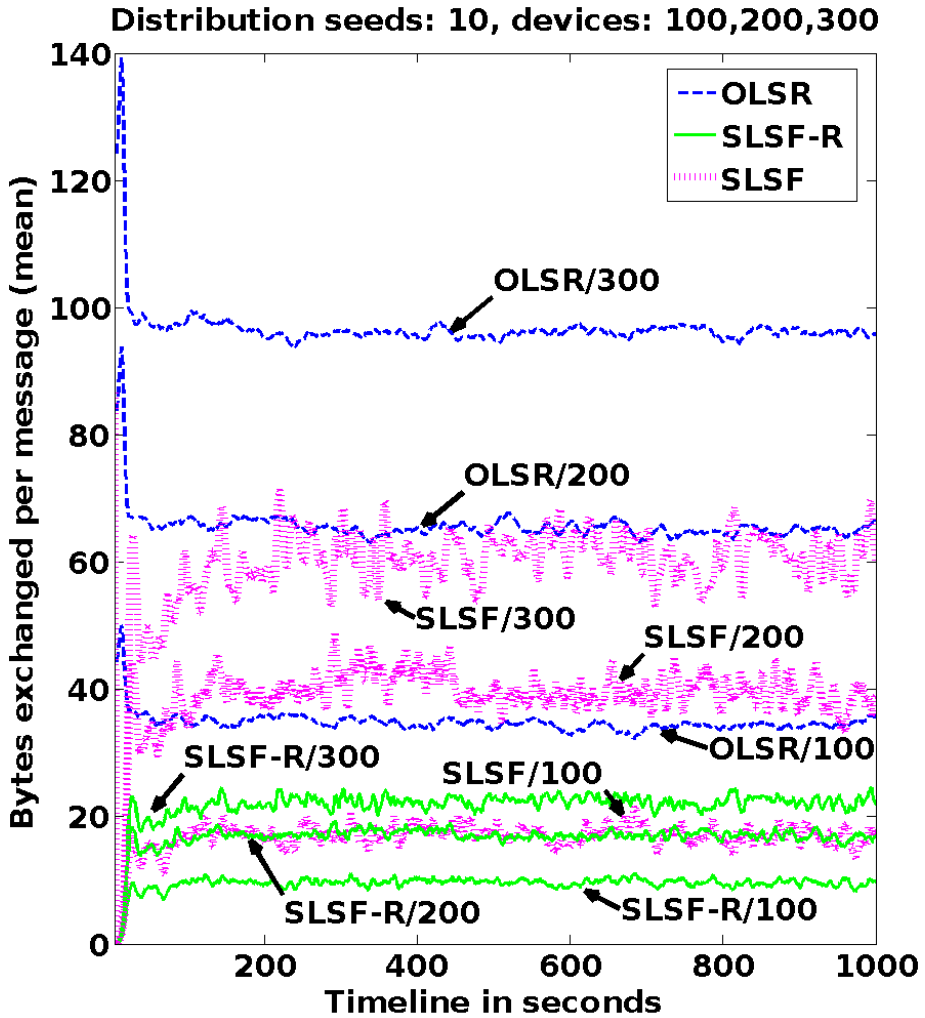
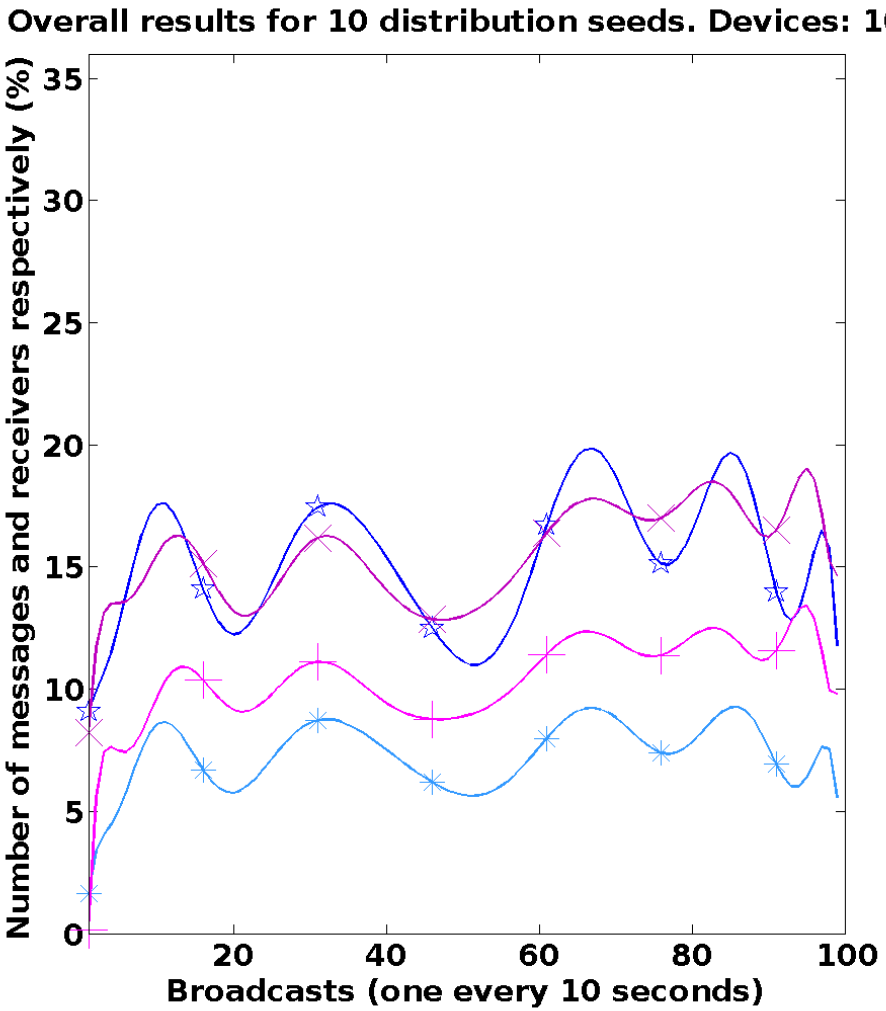
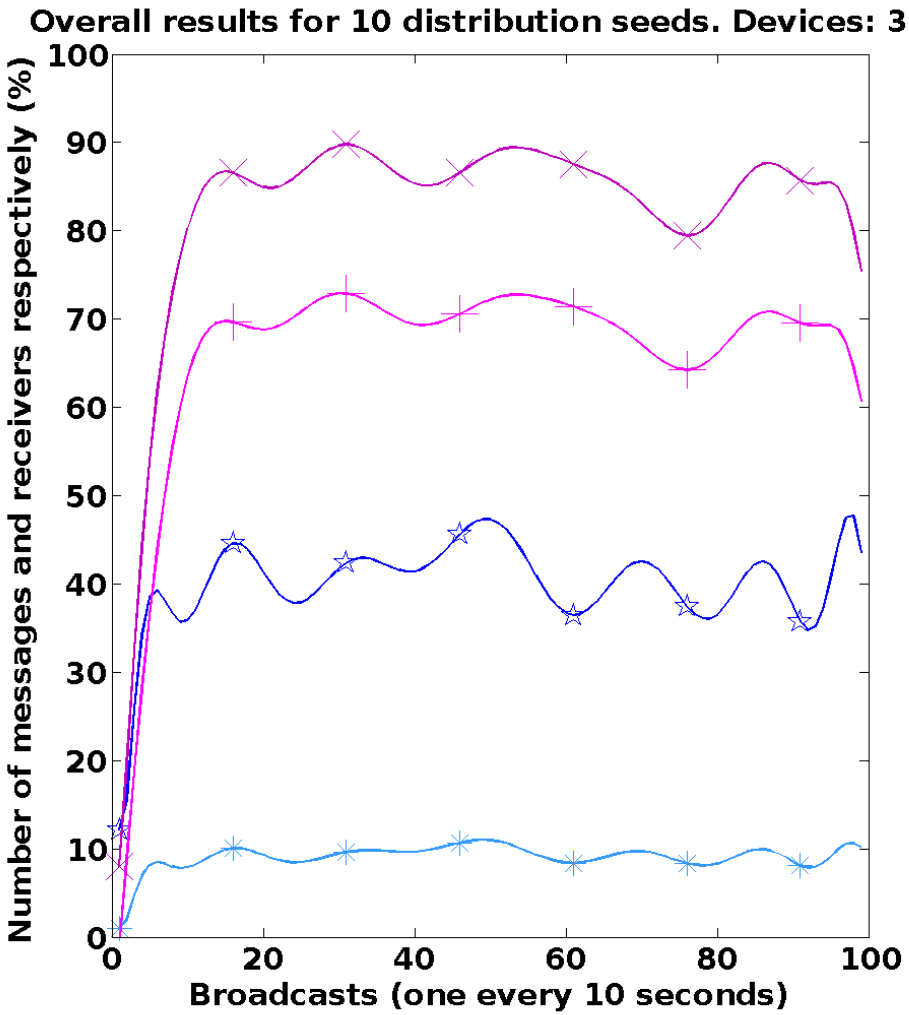


FIGURE: JANE Simulator

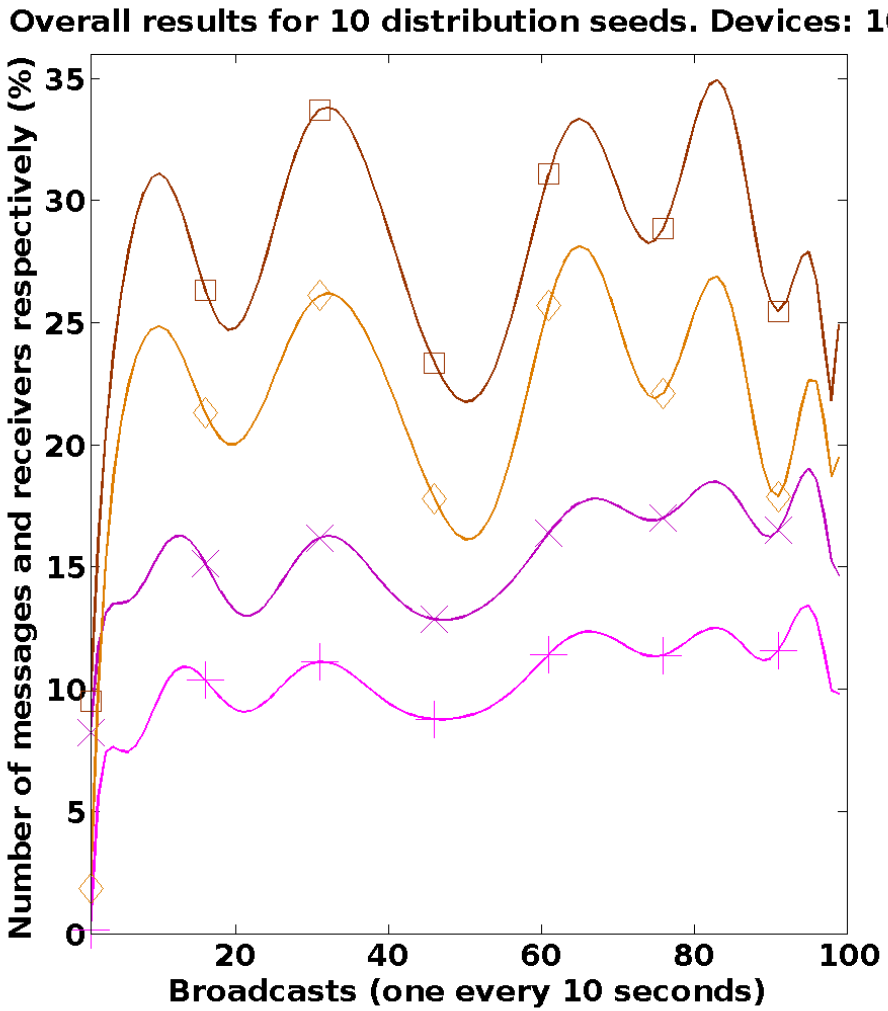


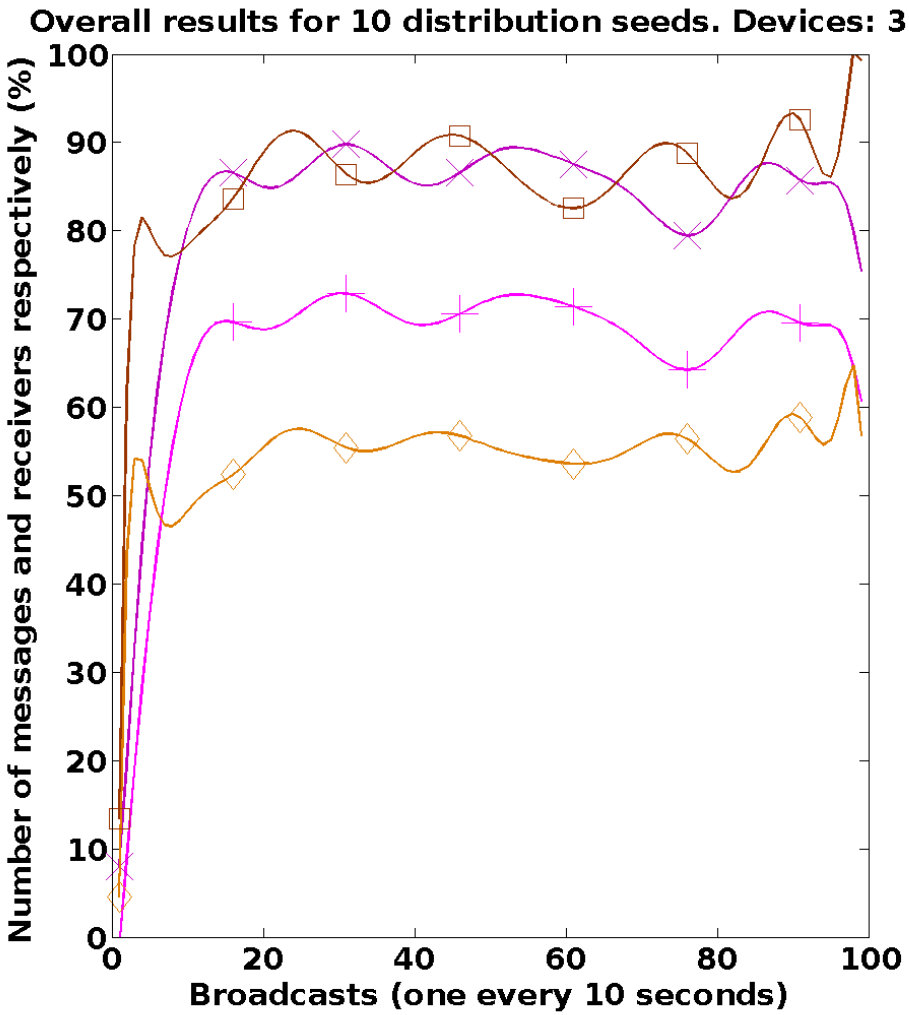
Dynamic Simulation Results





Dynamic Simulation Results





OLSR messages sent
OLSR receivers
SLSF-R messages sent
SLSF-R receivers
SLSF messages sent
SLSF receivers

SLSR

⇒ **Simplify the routing to its clusterheads**

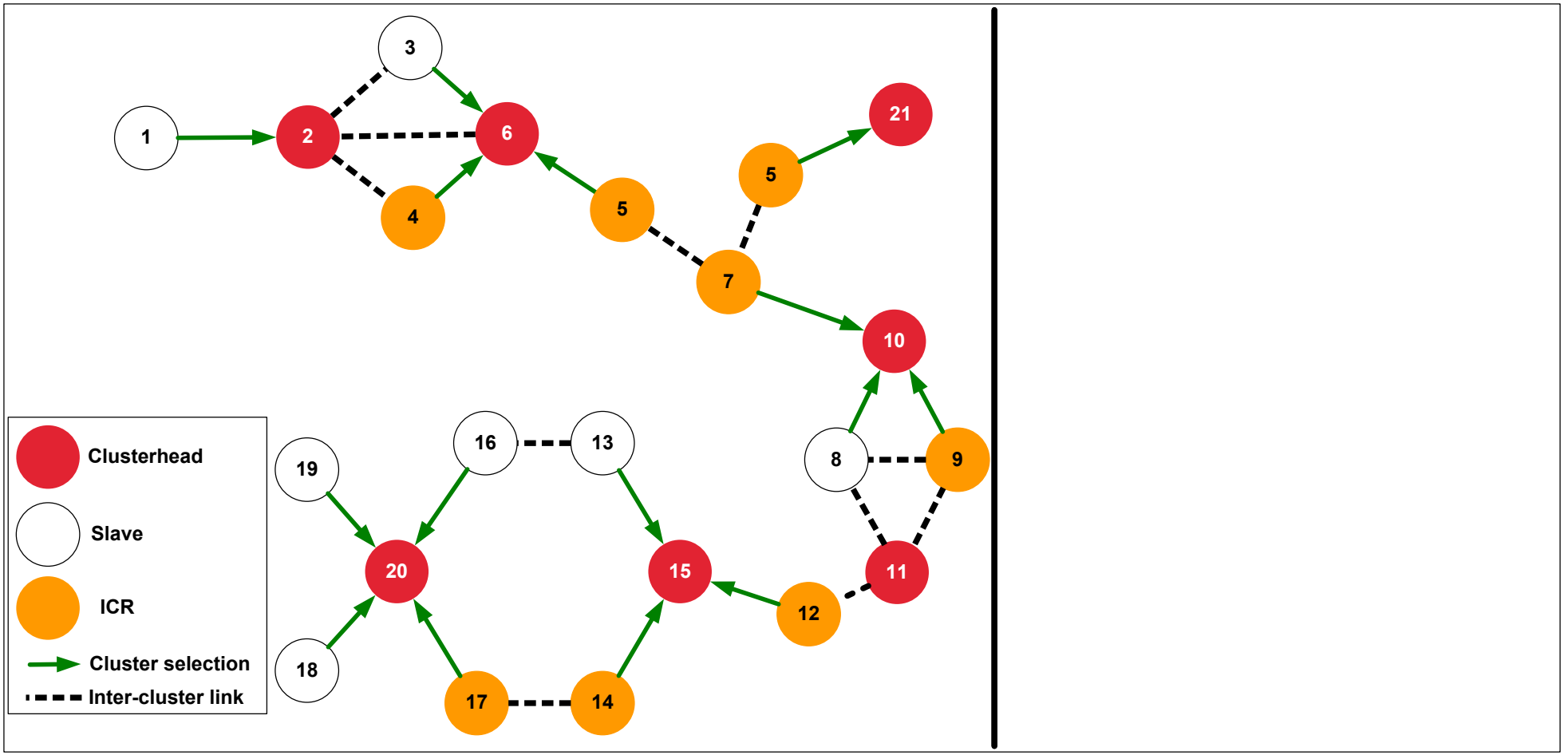
SLSF as dissemination structure

- Reduce the number of announcing nodes
- Aggregate information

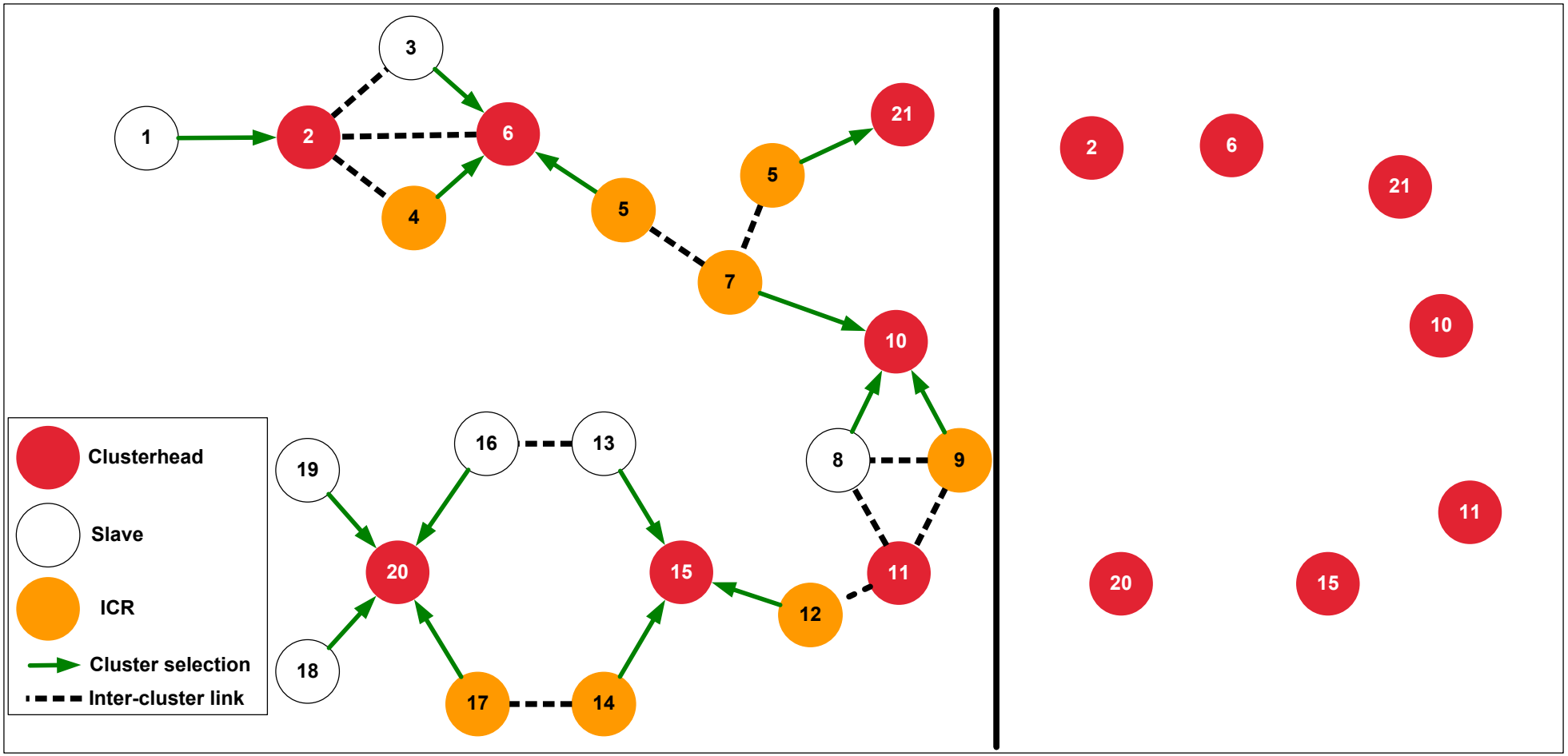
Principles

- Clusterheads :
  - Make routing announces
  - Gather routing table
  - Take all routing decisions
- Slaves :
  - Take no decisions
  - Rely exclusively on their clusterhead
  - Merely forward information

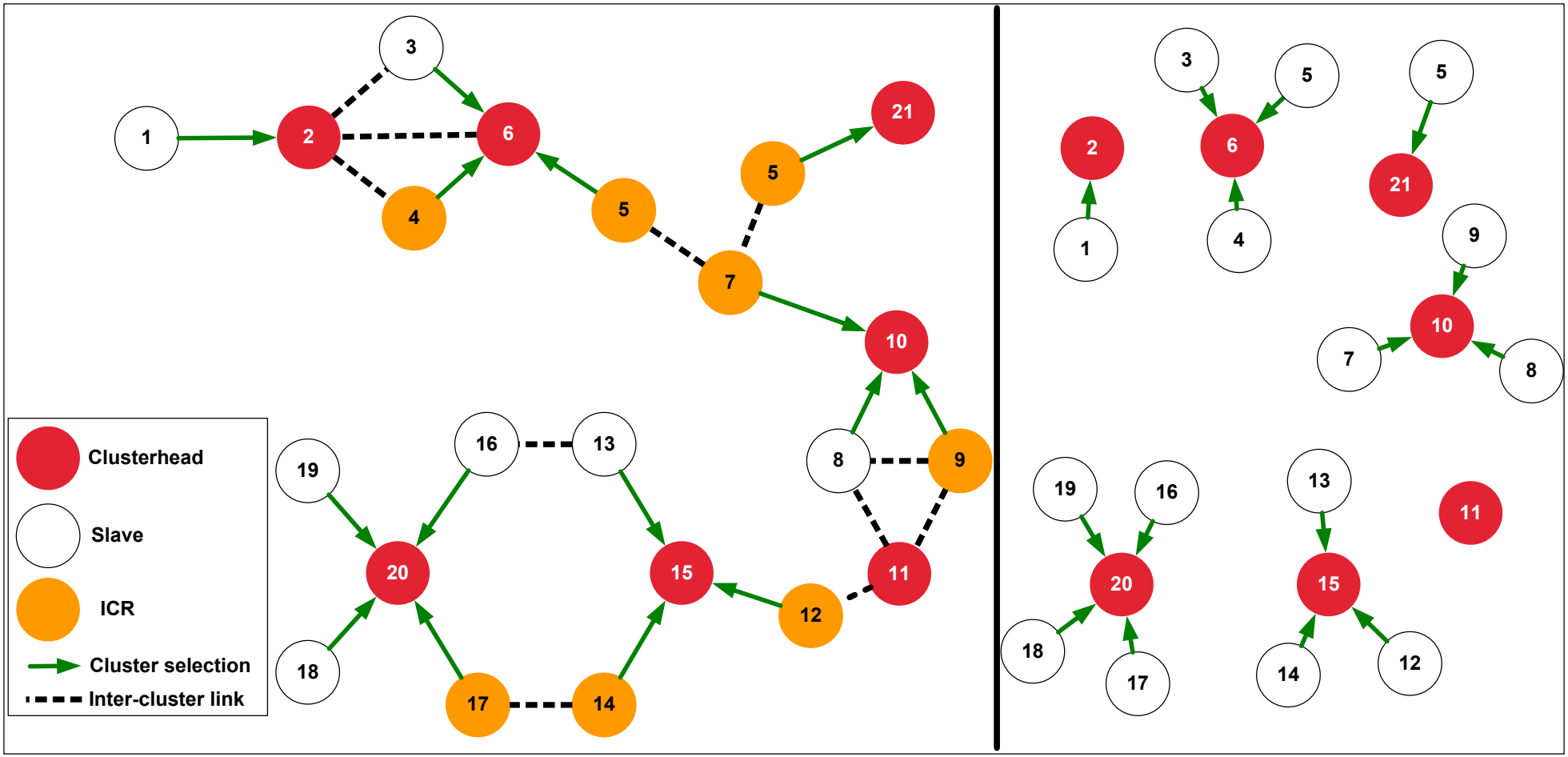
SLSR


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
Clusterhead
Slave
ICR
Cluster selection
Inter-cluster link

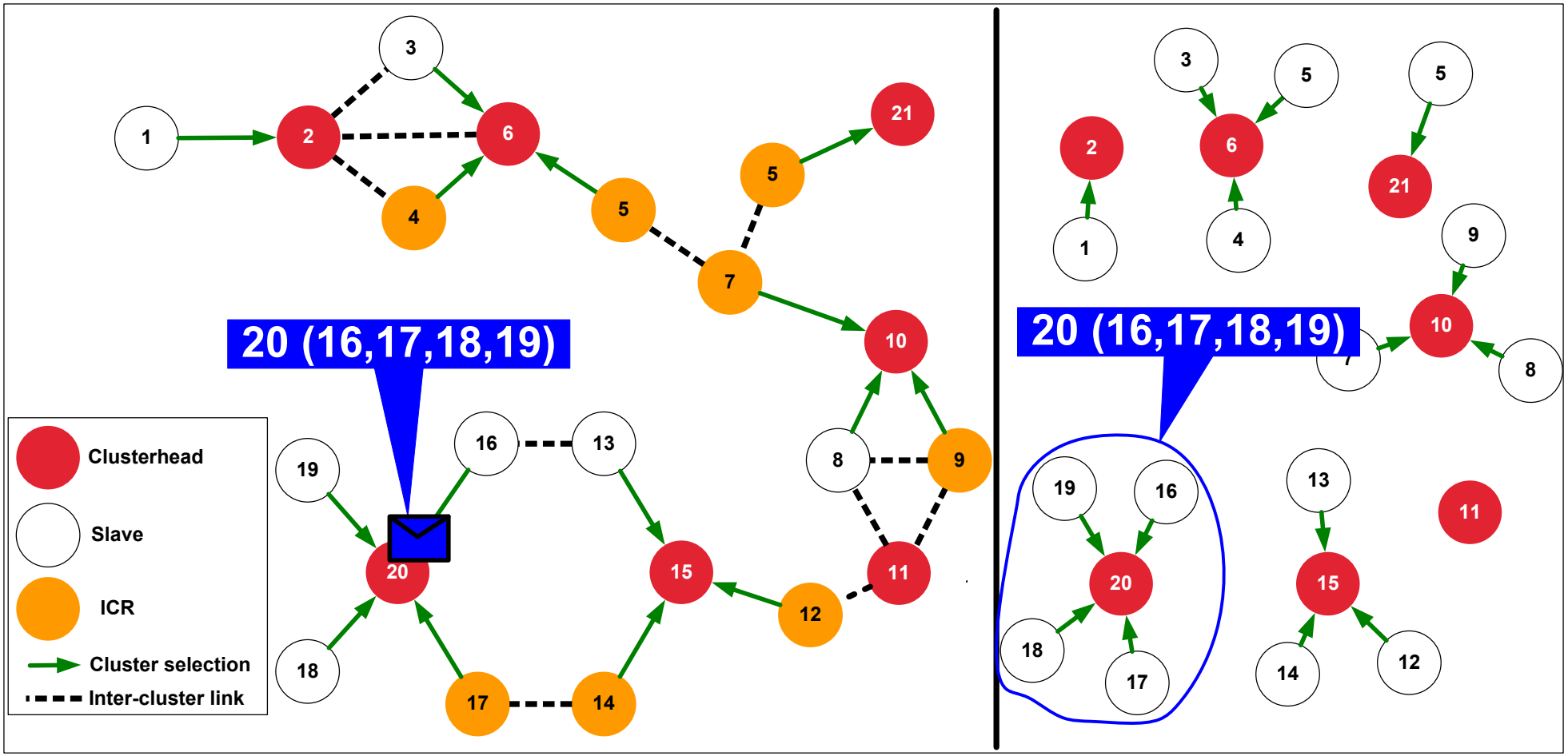
SLSR


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
Clusterhead
Slave
ICR
Cluster selection
Inter-cluster link
2
6
21
10
11
20
15

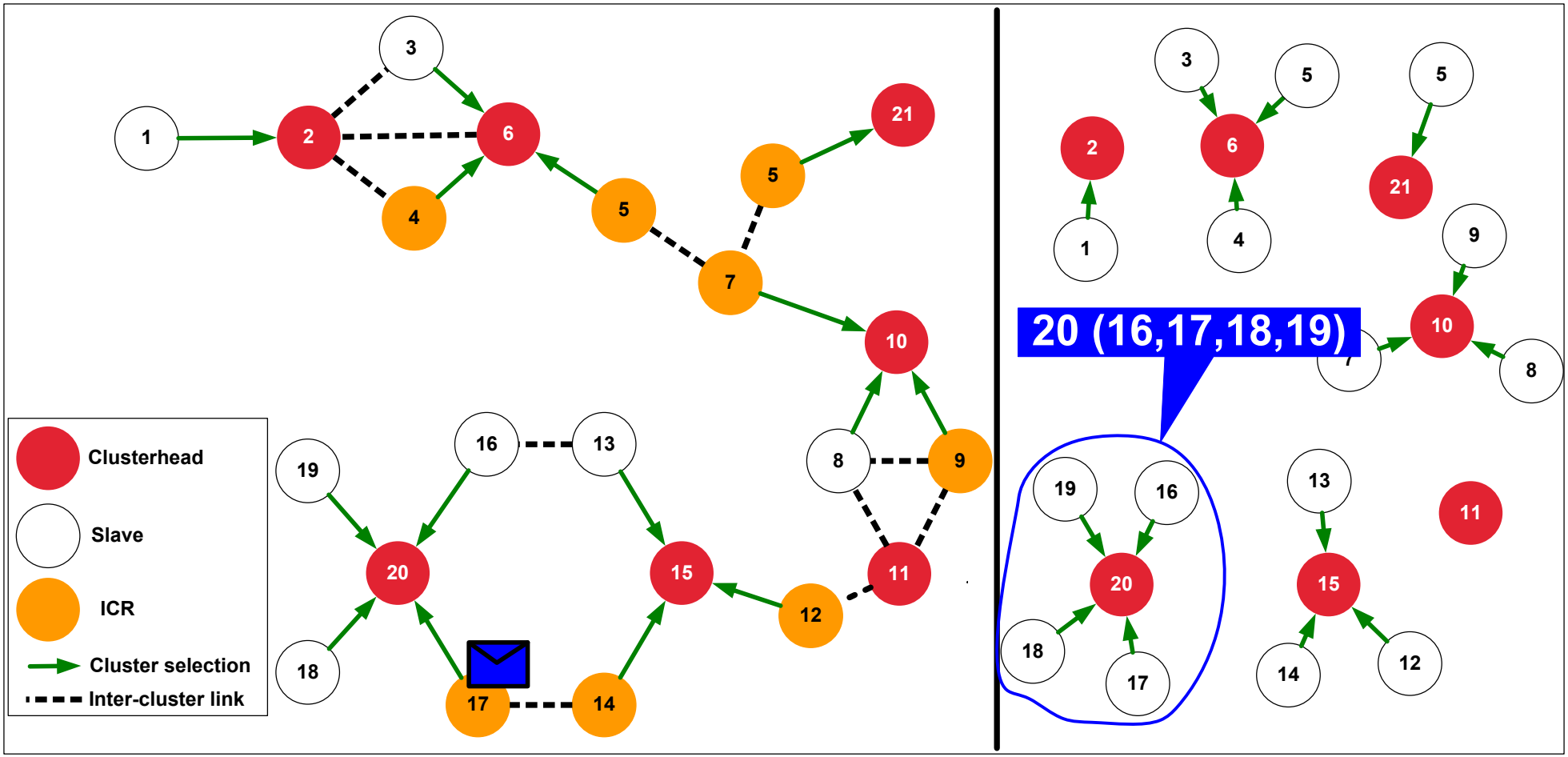
SLSR


1
2
3
4
5
6
5
21
7
10
8
9
11
12
15
20
16
13
19
18
17
14
Clusterhead
Slave
ICR
Cluster selection
Inter-cluster link
3
5
2
6
1
4
5
21
9
10
7
8
19
16
13
11
20
15
18
17
14
12

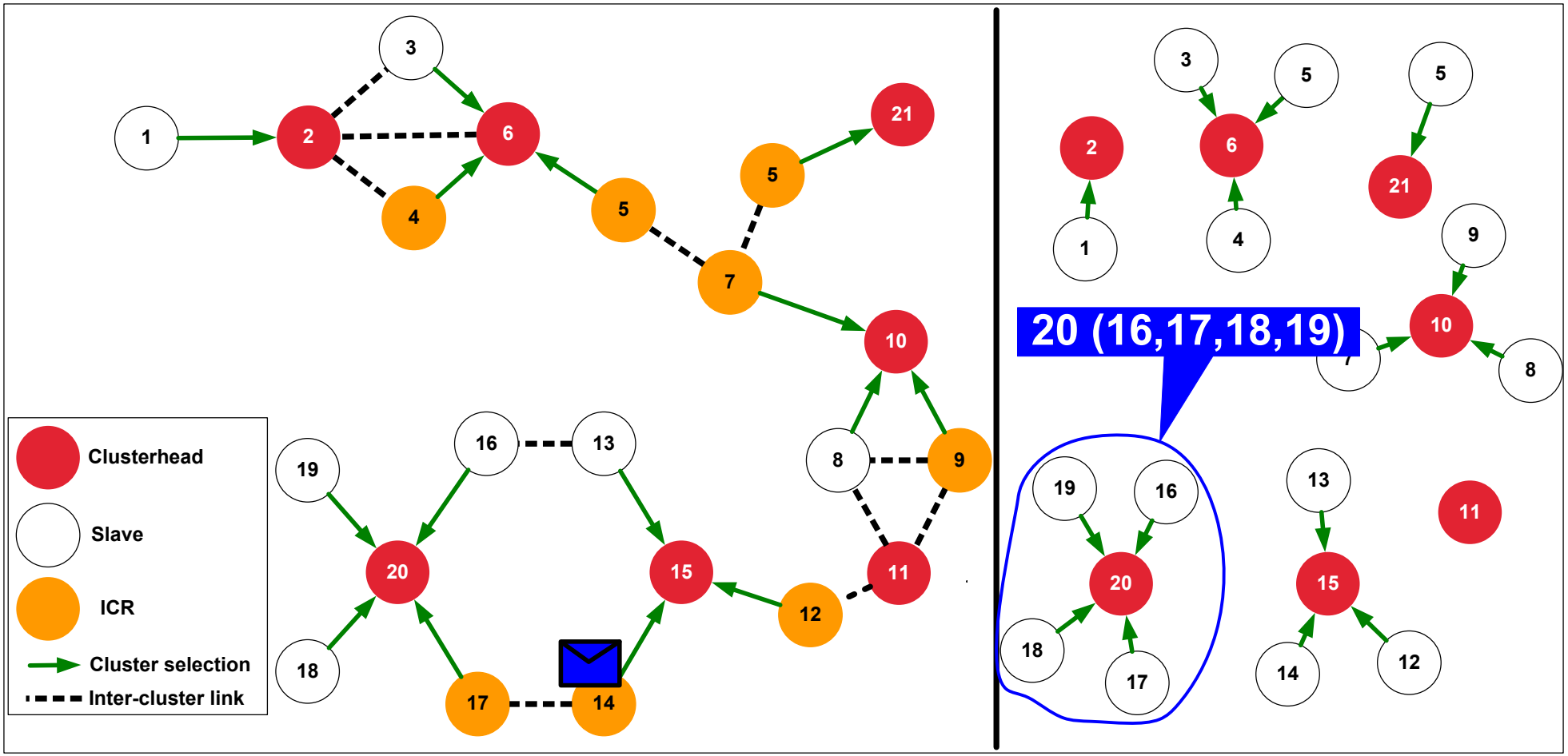
SLSR


20 (16,17,18,19)
Clusterhead
Slave
ICR
Cluster selection
Inter-cluster link
20 (16,17,18,19)

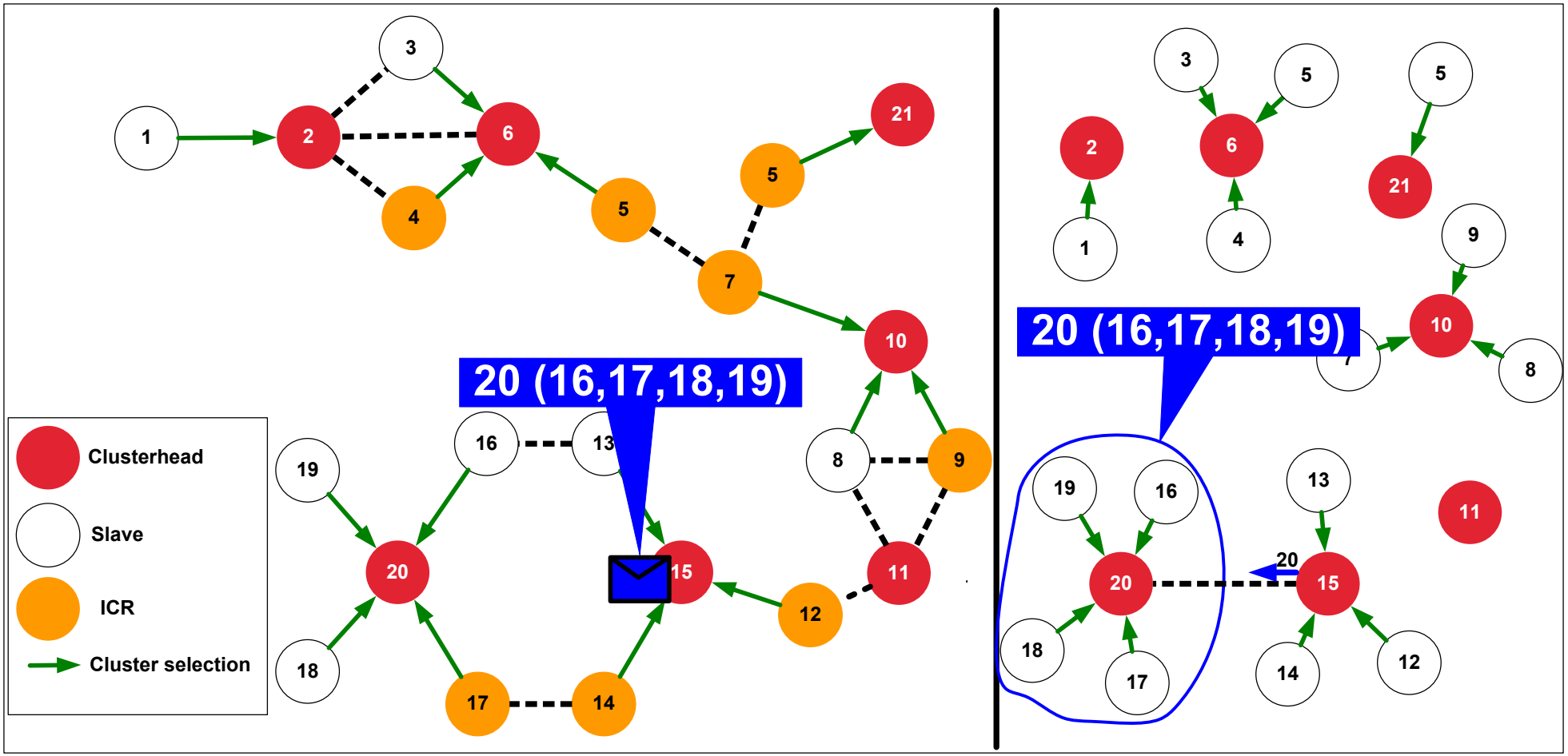
SLSR


Clusterhead
Slave
ICR
Cluster selection
Inter-cluster link
20 (16,17,18,19)

SLSR

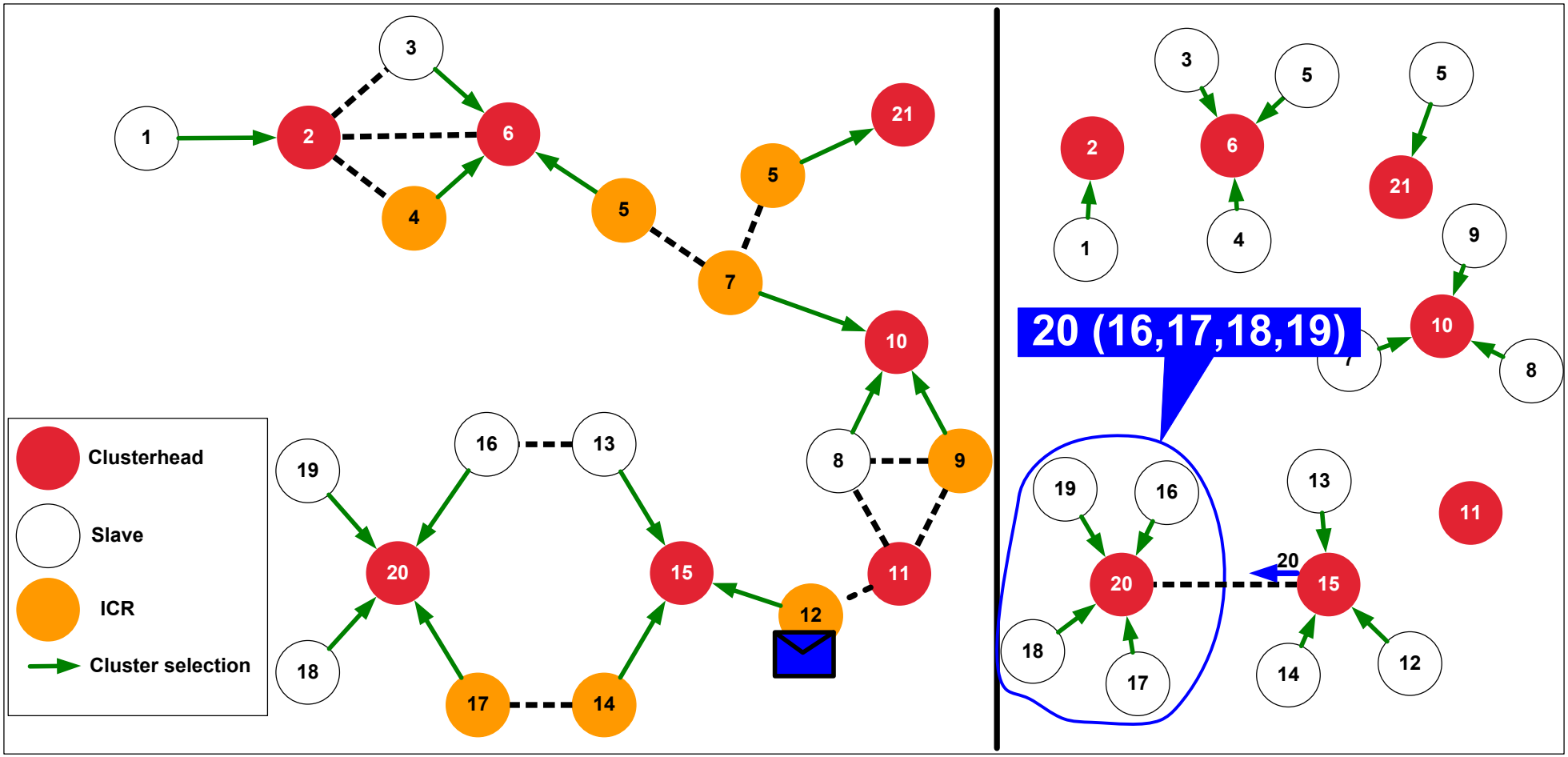

Clusterhead
Slave
ICR
Cluster selection
Inter-cluster link
20 (16,17,18,19)

SLSR


Clusterhead
Slave
ICR
Cluster selection
20 (16,17,18,19)
20 (16,17,18,19)

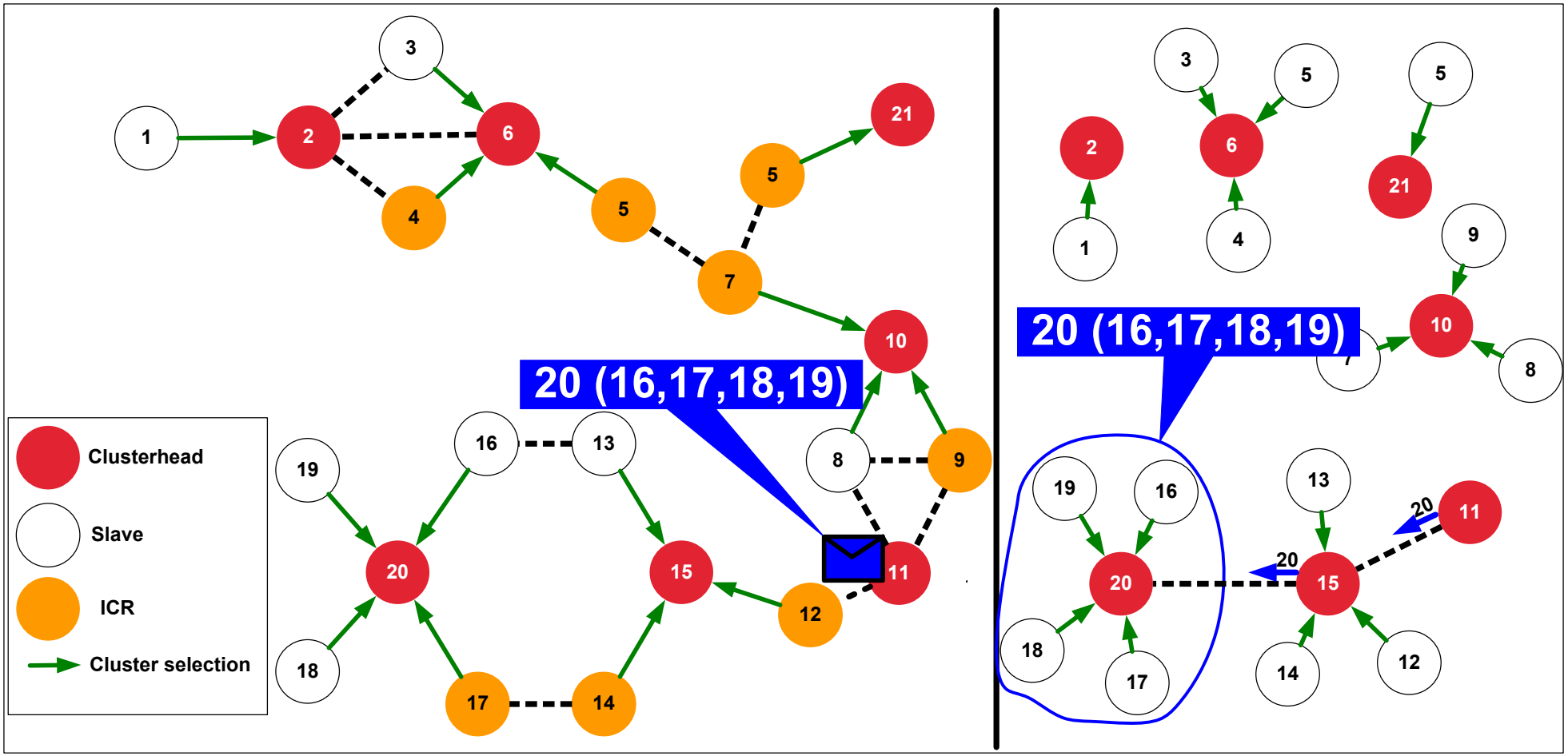
Stable Linked Structure Routing

SLSR



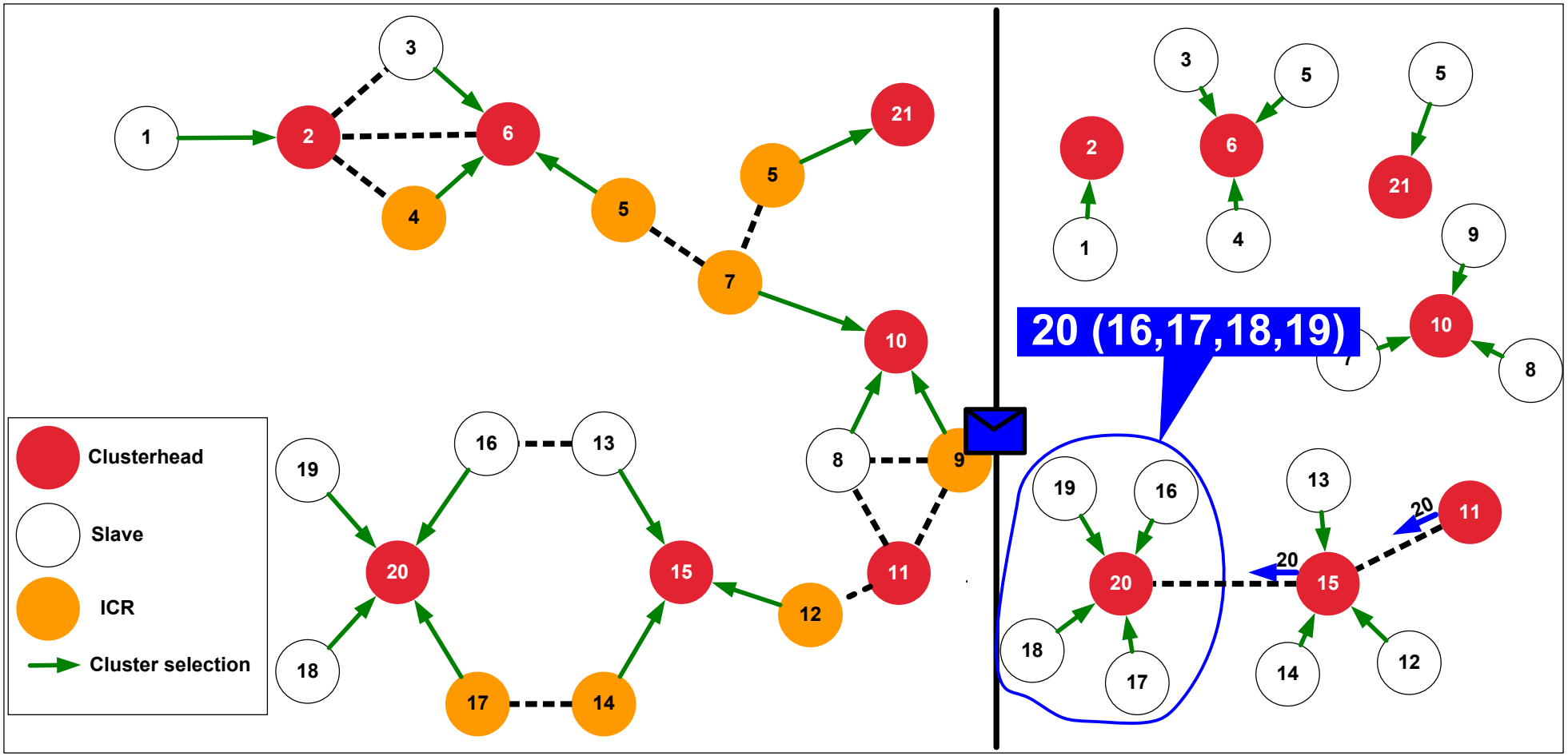
Stable Linked Structure Routing

SLSR

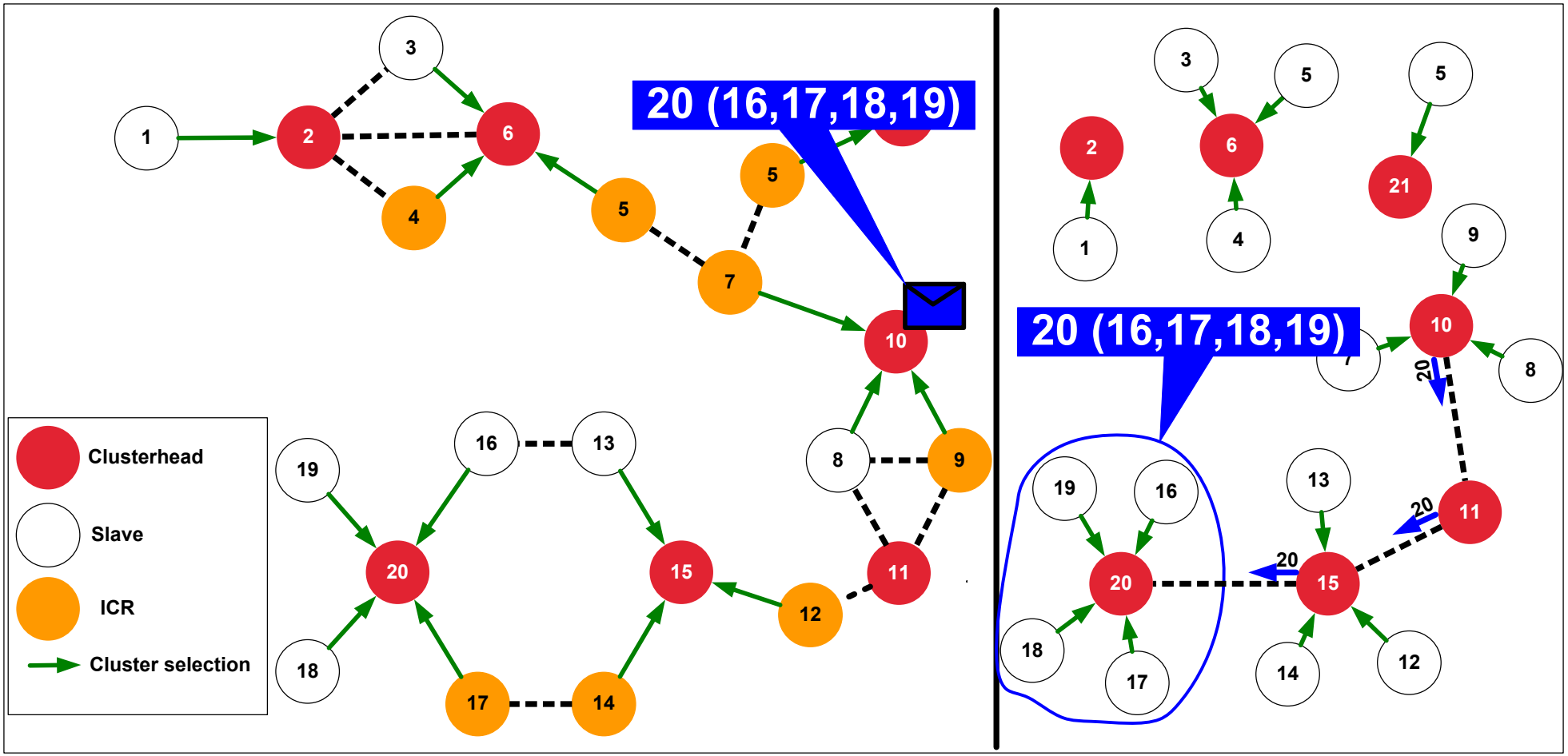


Stable Linked Structure Routing

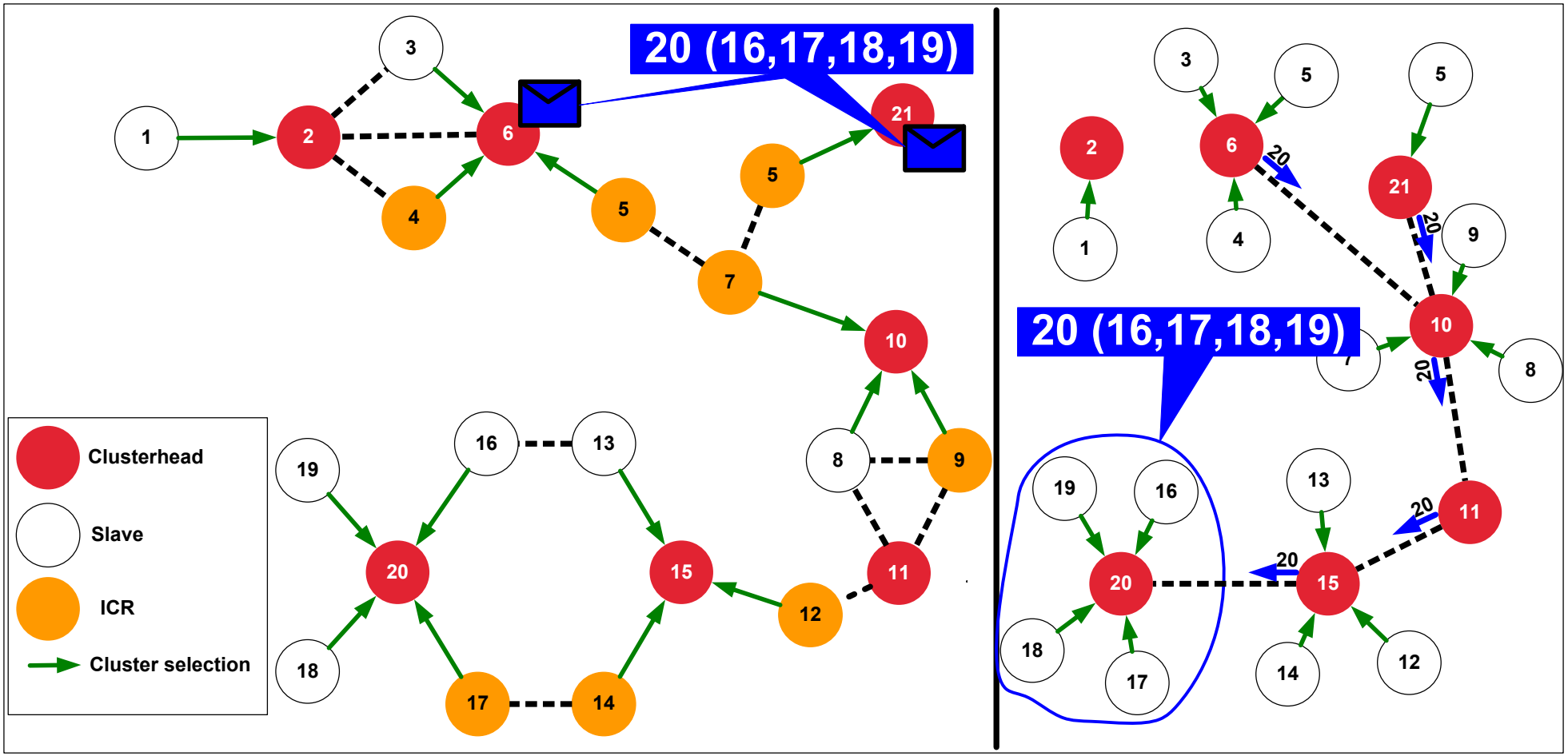
SLSR



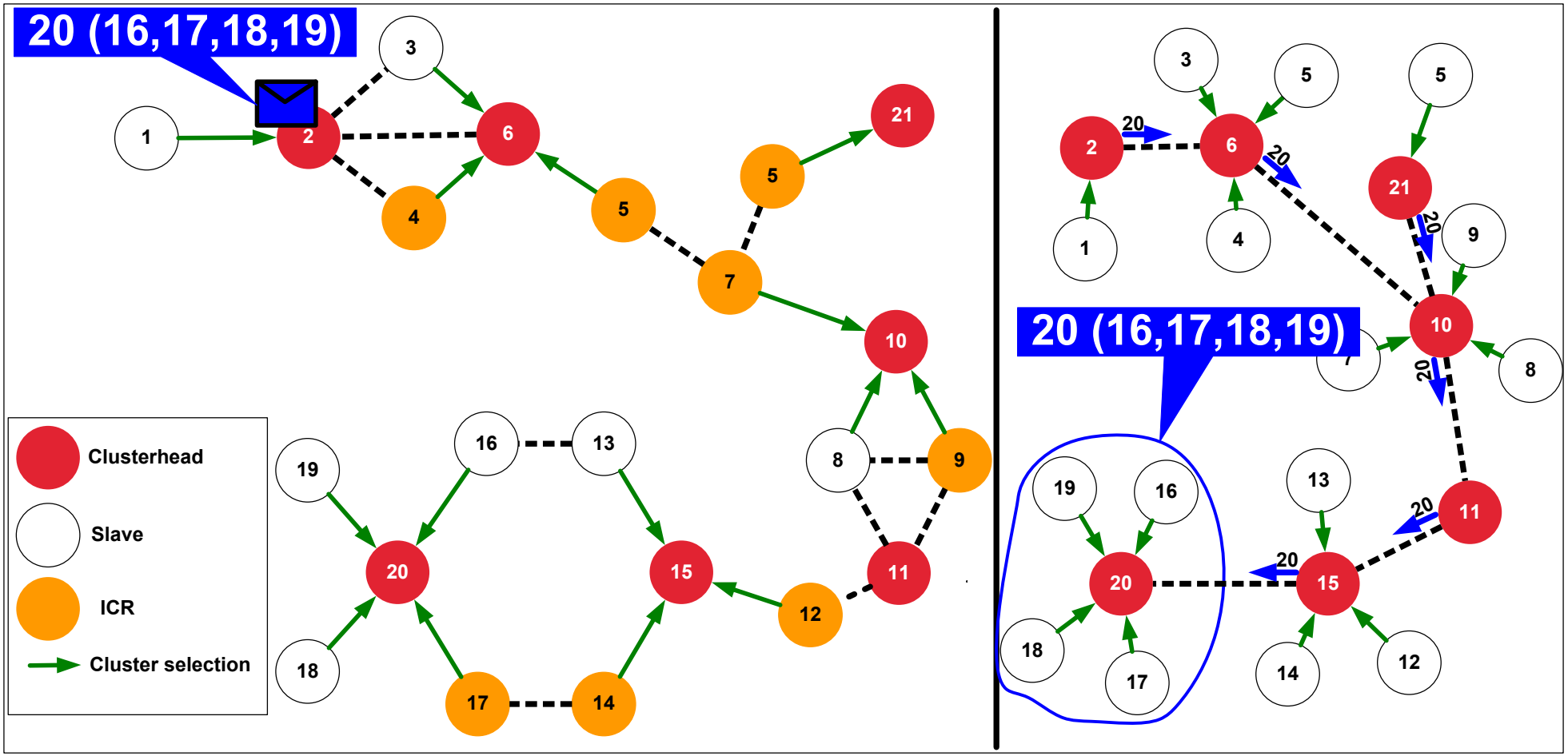
SLSR


20 (16,17,18,19)
Clusterhead
Slave
ICR
Cluster selection
20 (16,17,18,19)

SLSR

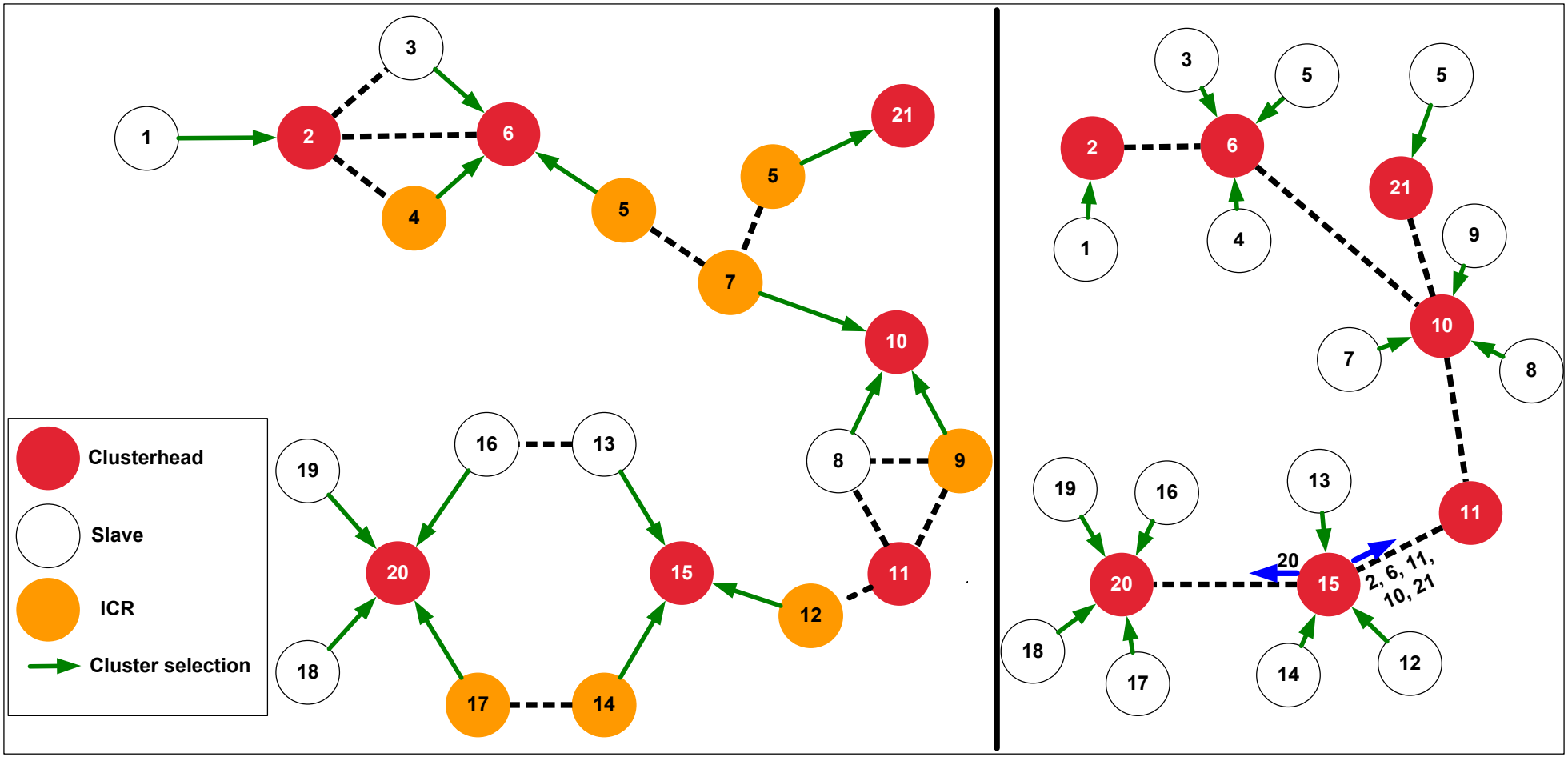

20 (16,17,18,19)
Clusterhead
Slave
ICR
Cluster selection
20 (16,17,18,19)

SLSR

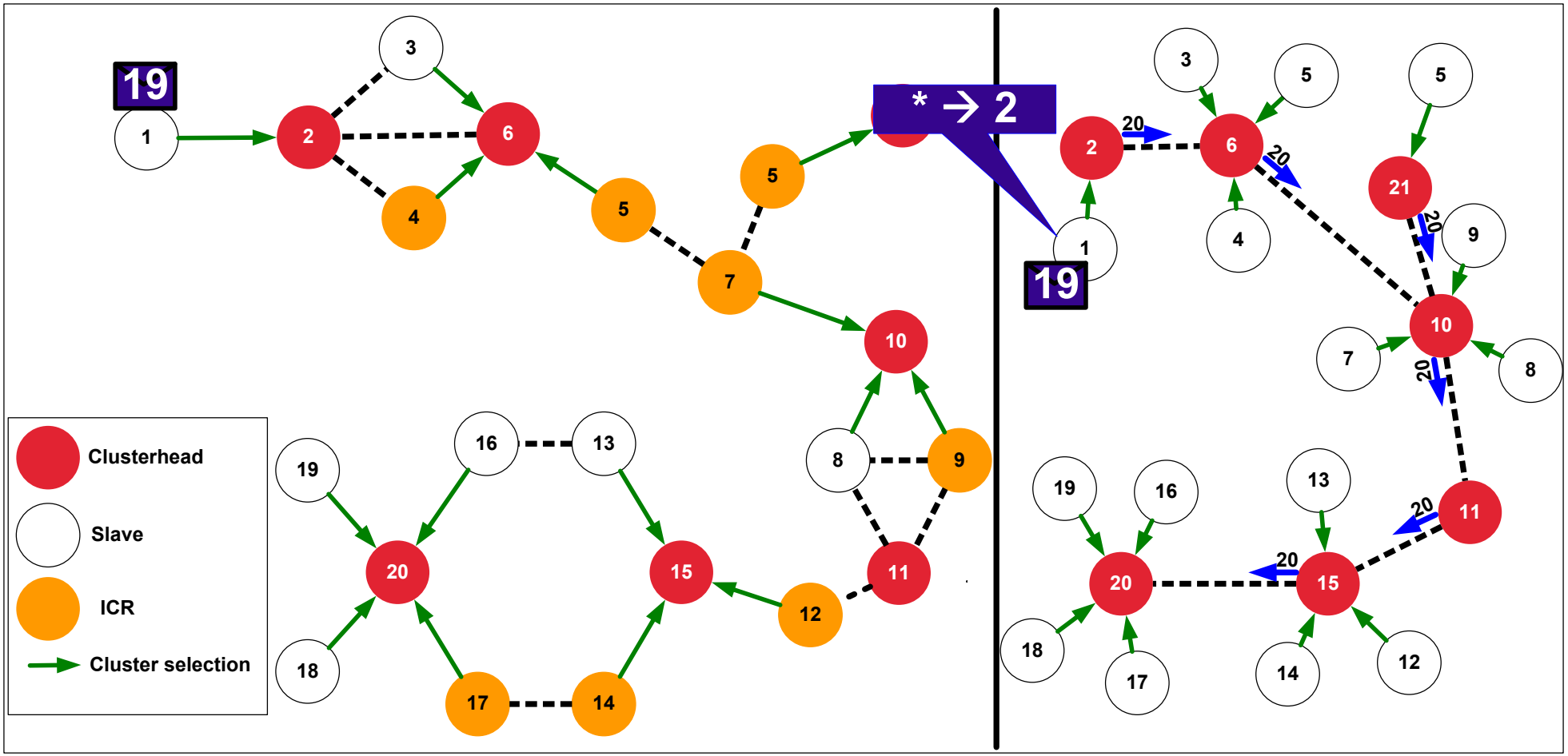

20 (16,17,18,19)
Clusterhead
Slave
ICR
Cluster selection
20 (16,17,18,19)

Stable Linked Structure Routing

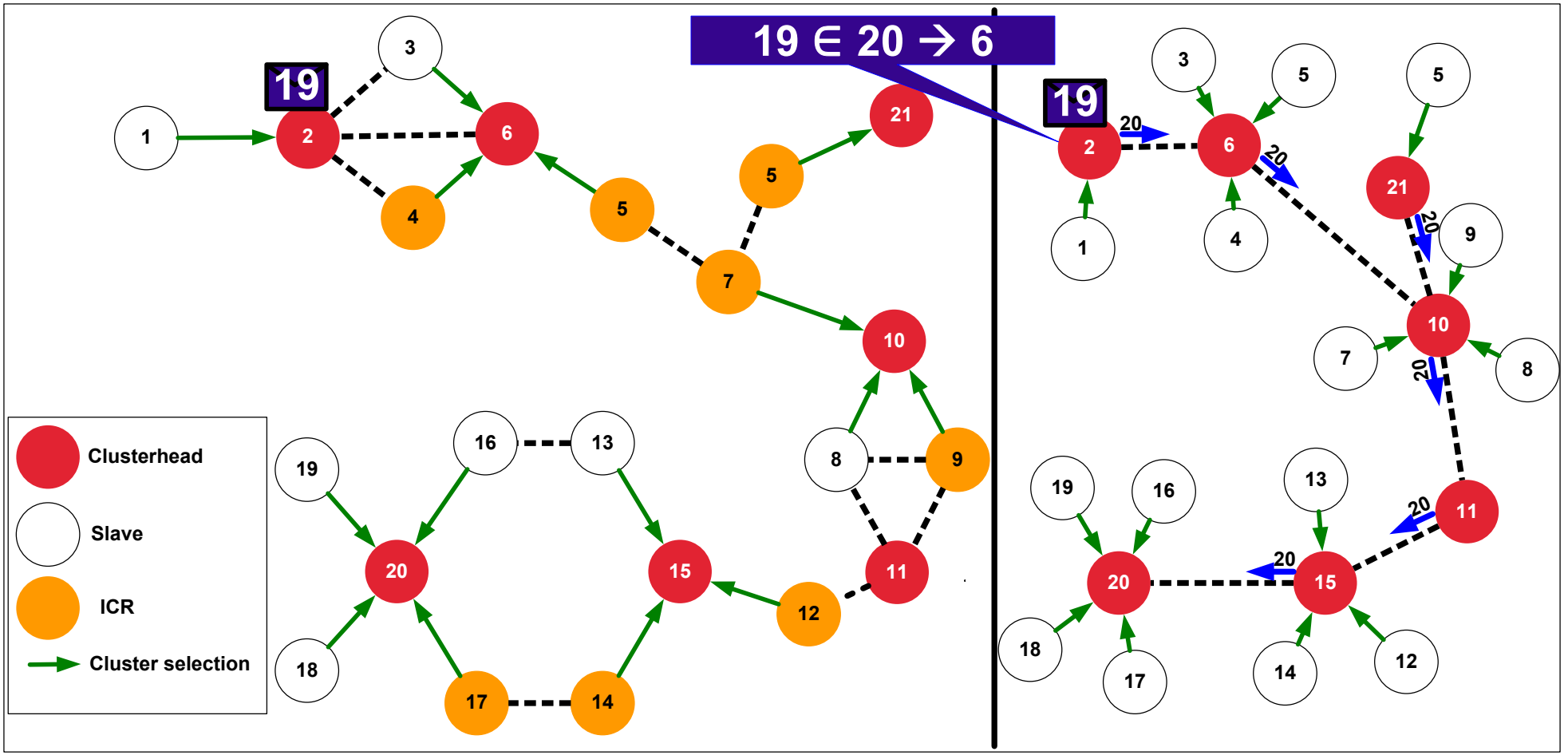
SLSR



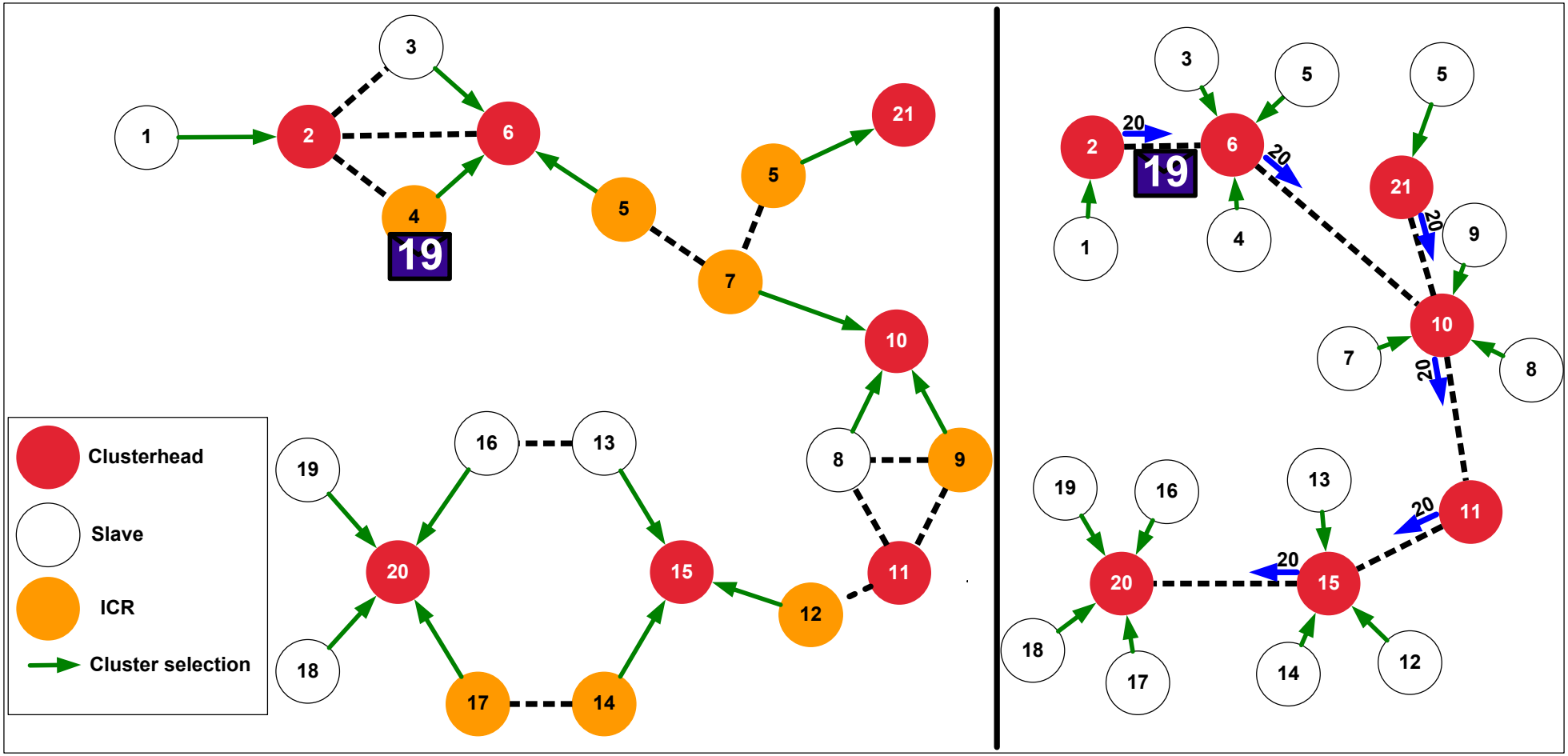
SLSR



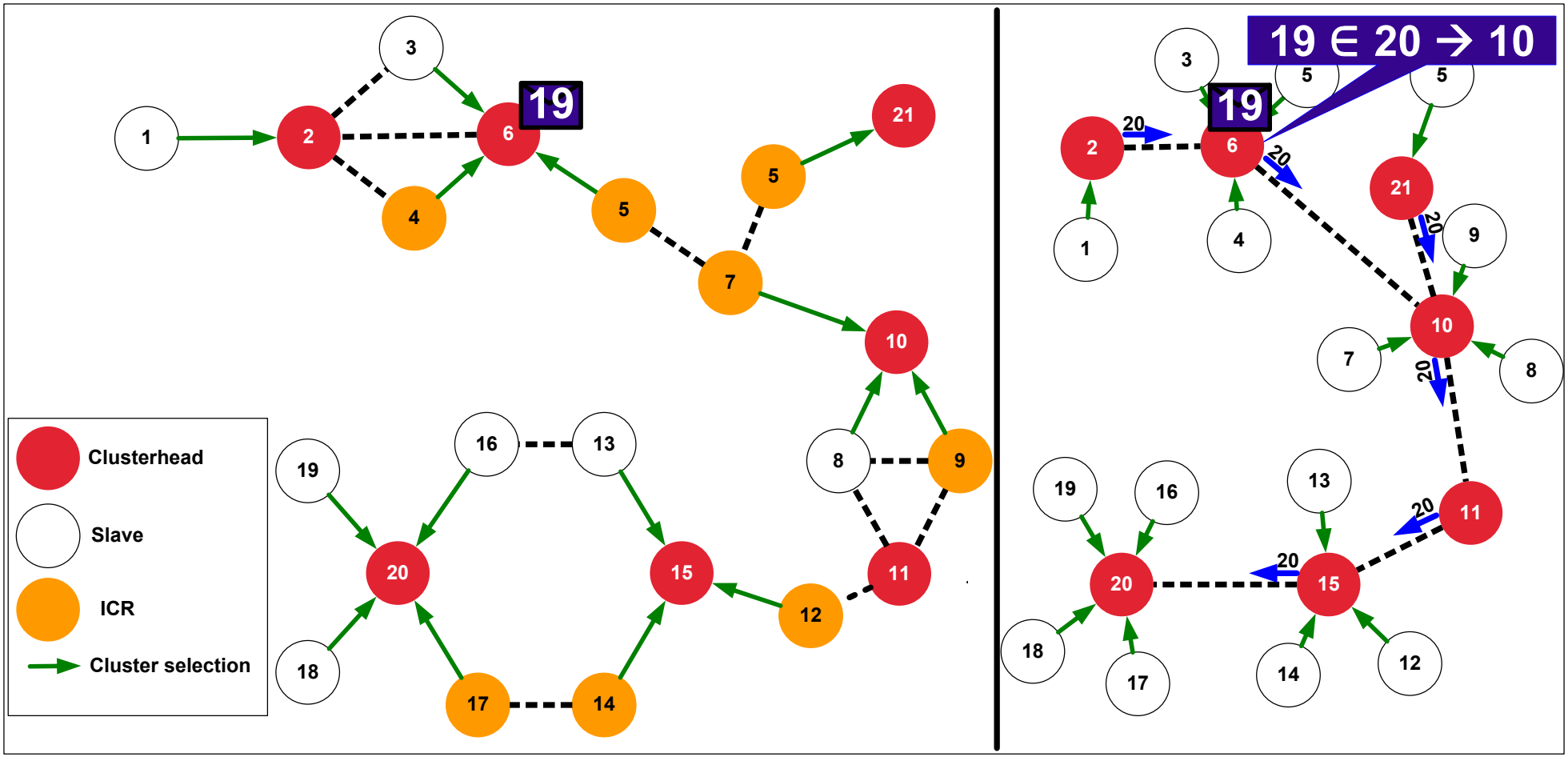
SLSR



SLSR

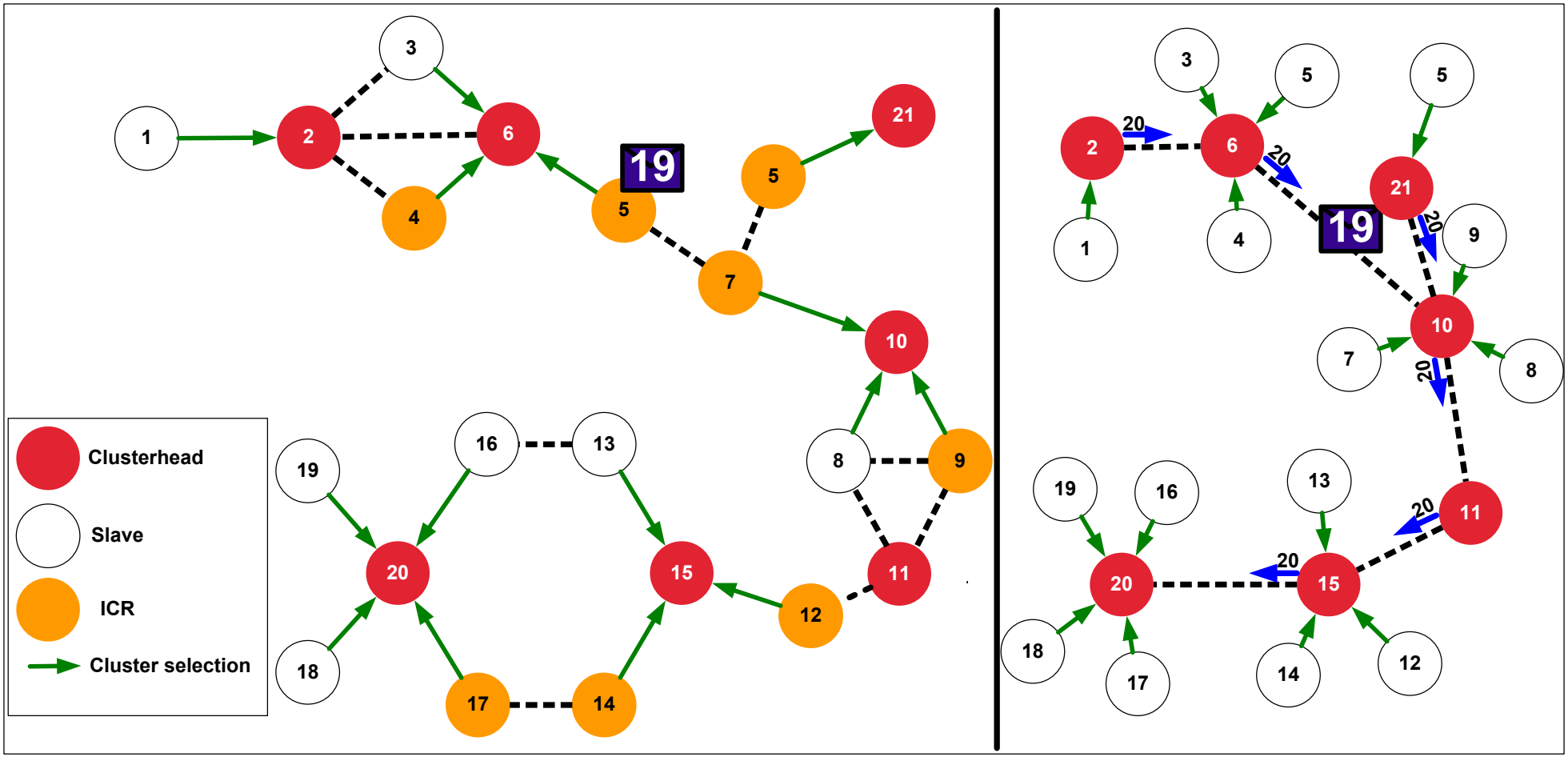


SLSR



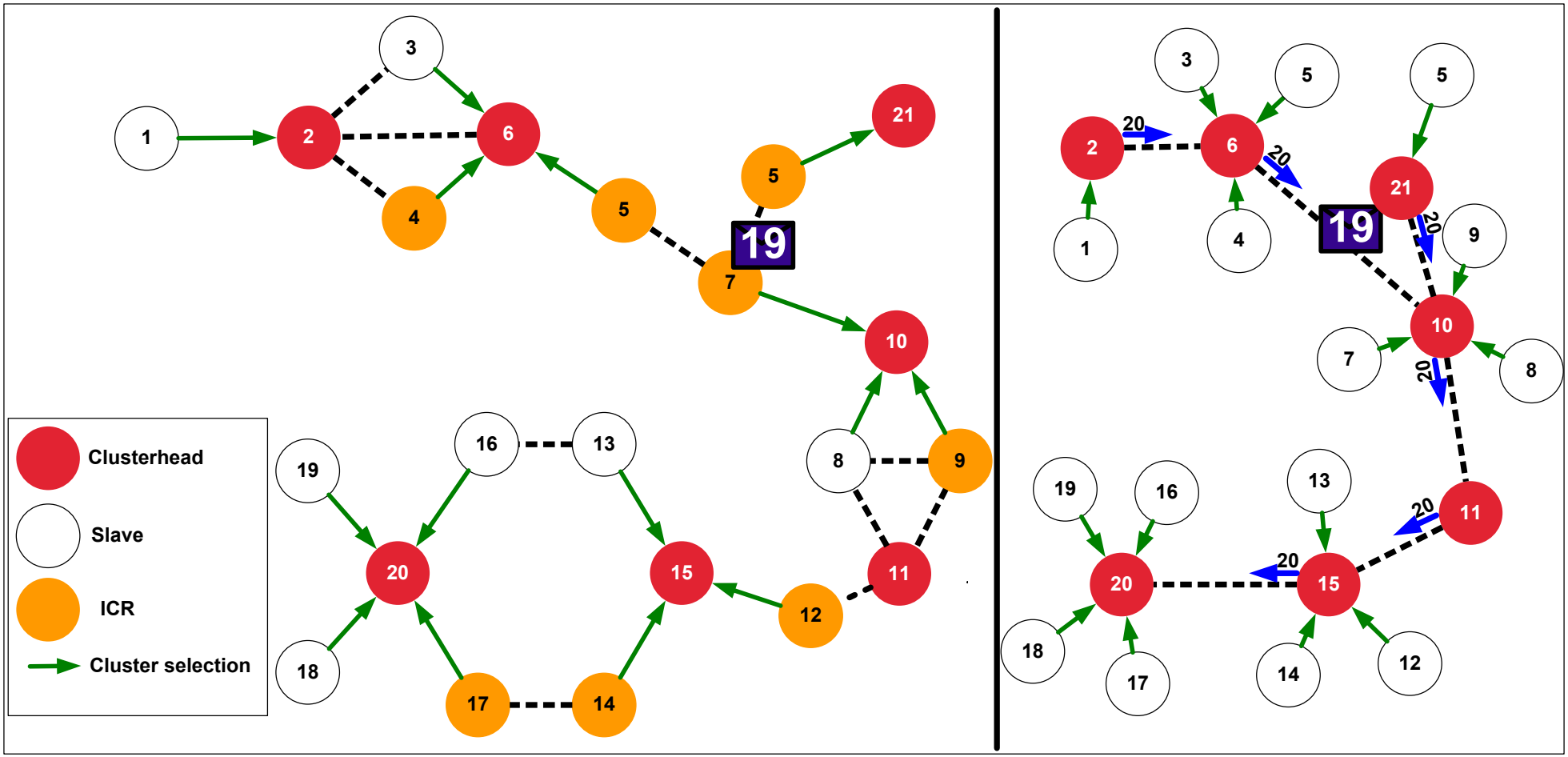
Stable Linked Structure Routing

SLSR

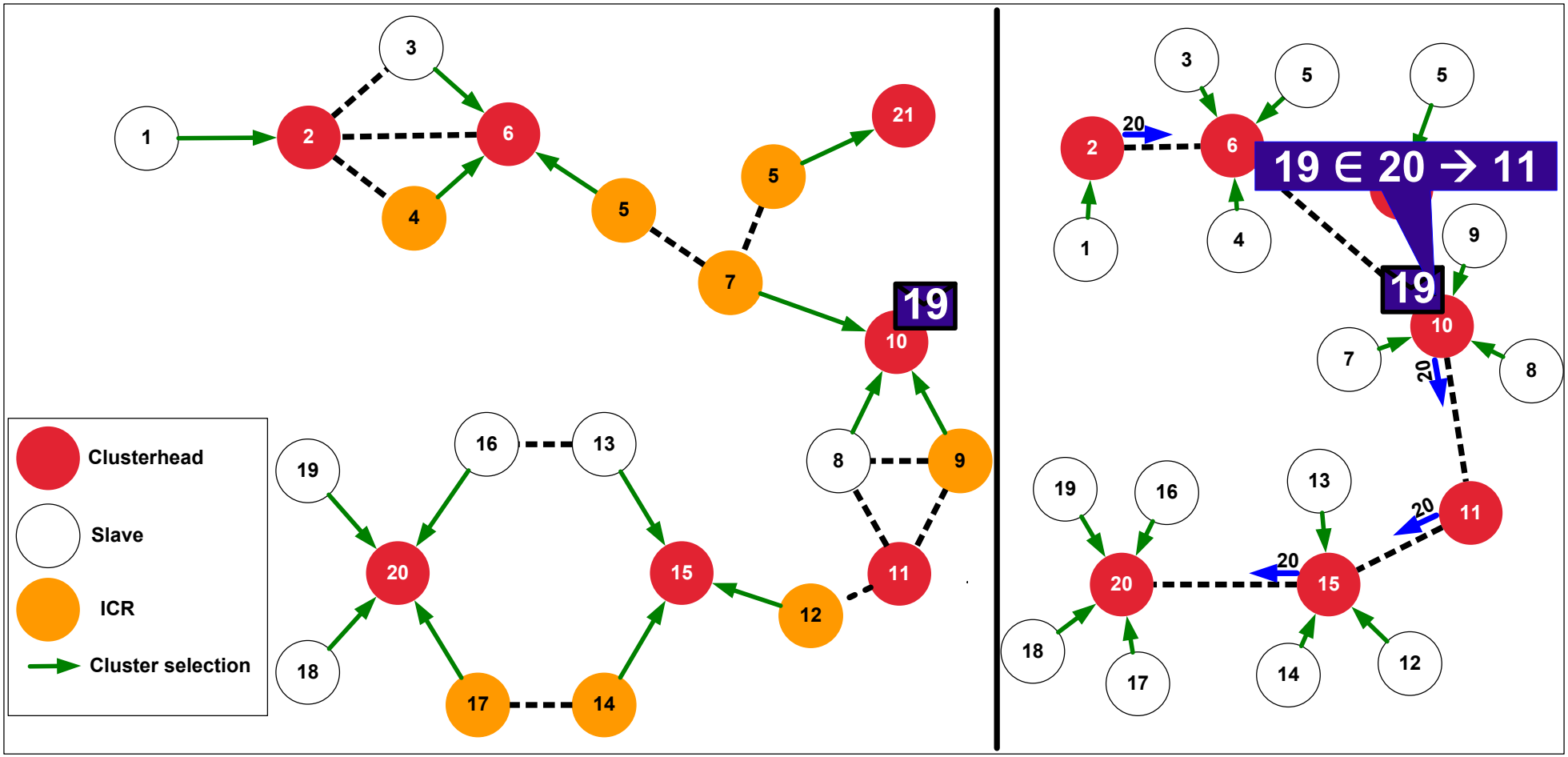


Stable Linked Structure Routing

SLSR

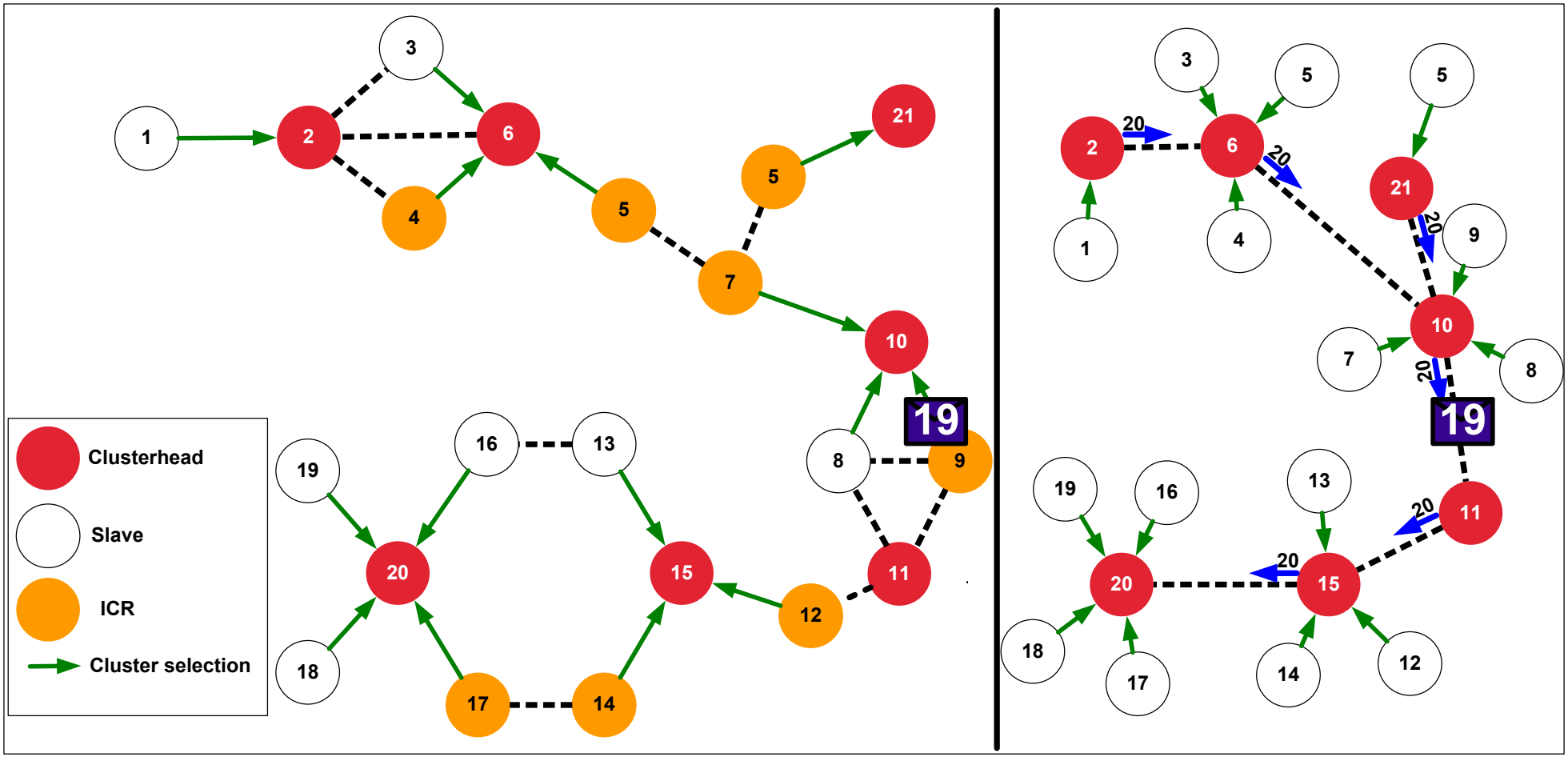


SLSR

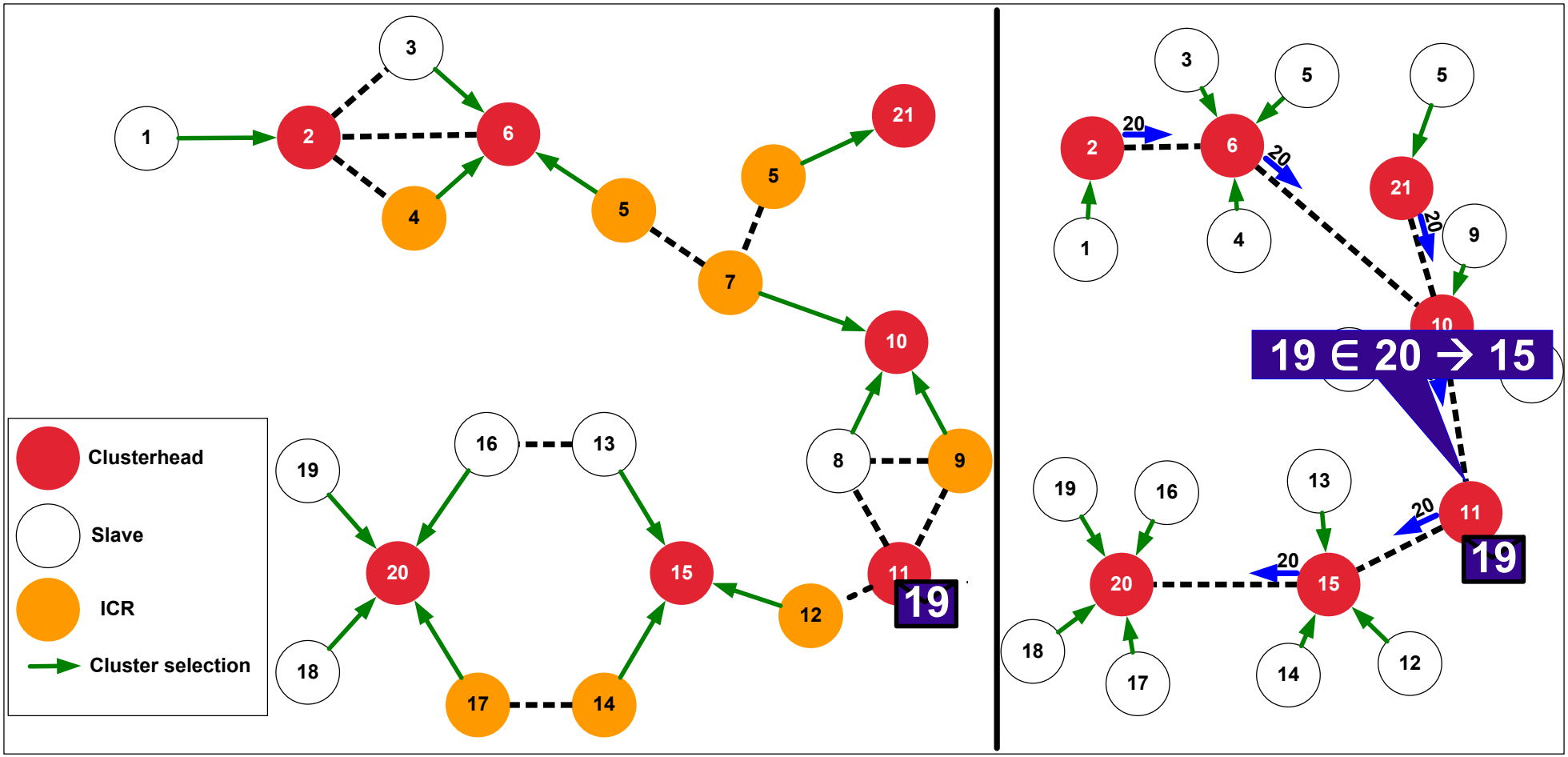


Stable Linked Structure Routing

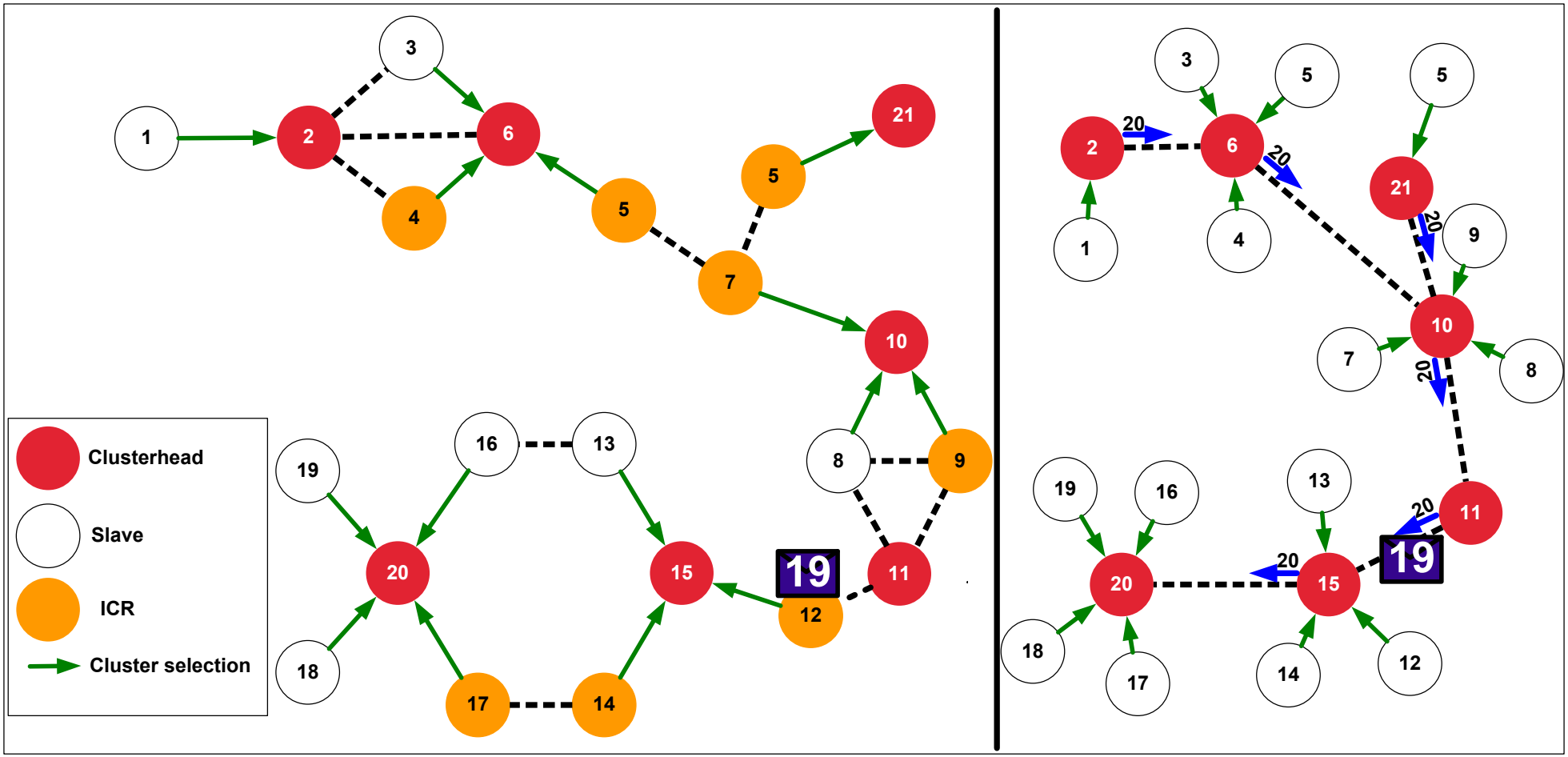
SLSR



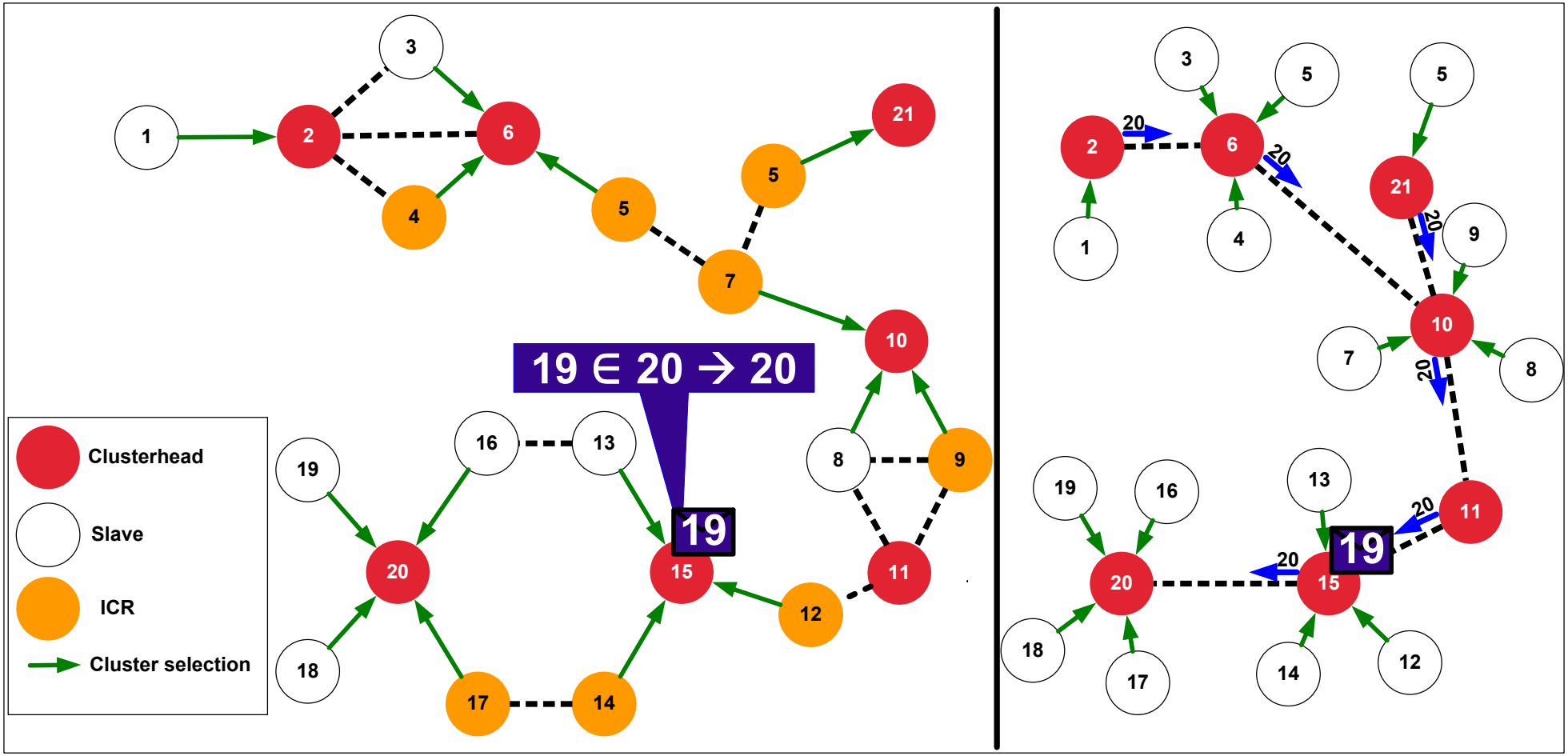
SLSR



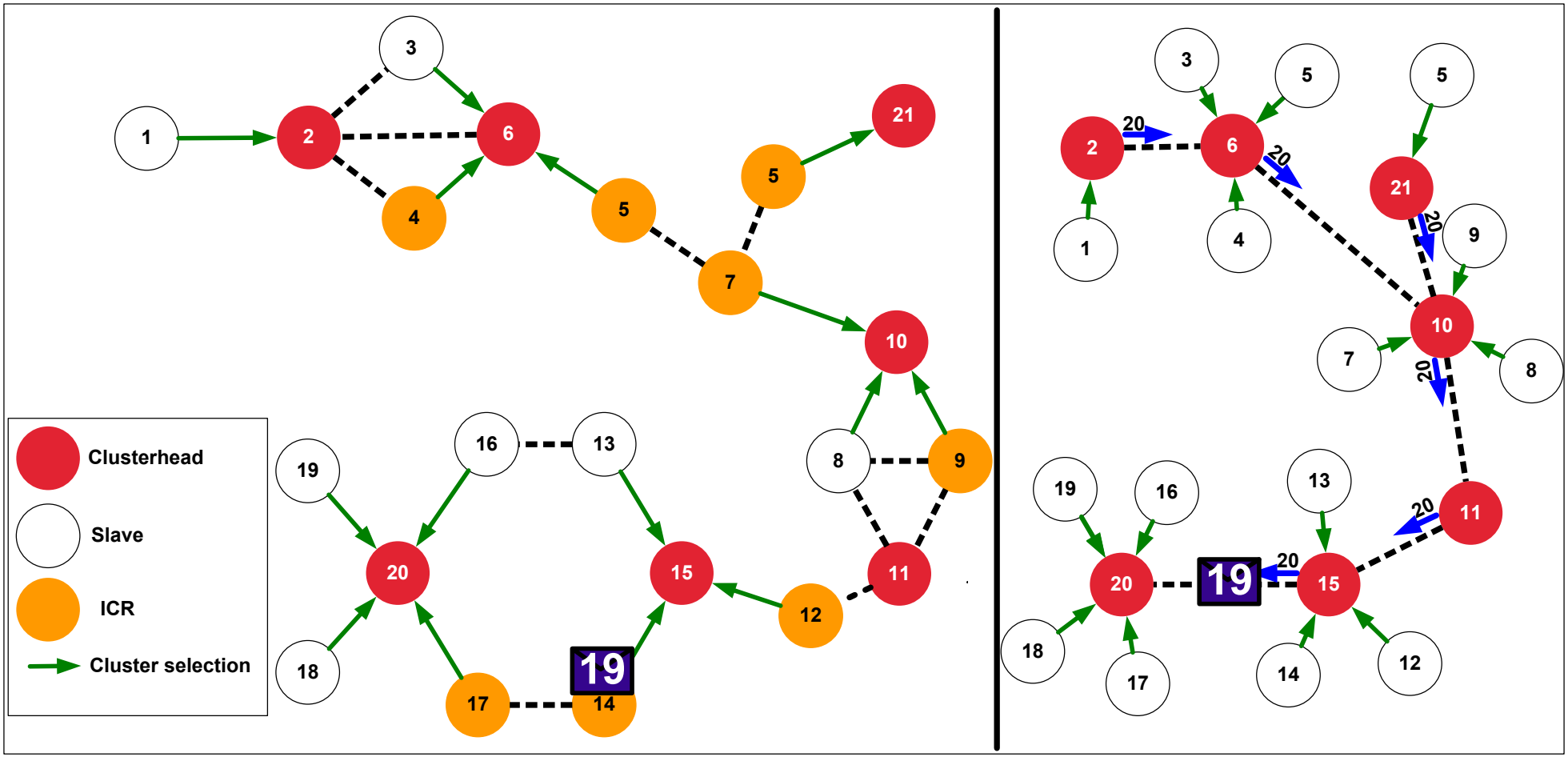
SLSR


Clusterhead
Slave
ICR
Cluster selection
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
20
20
20
20
20

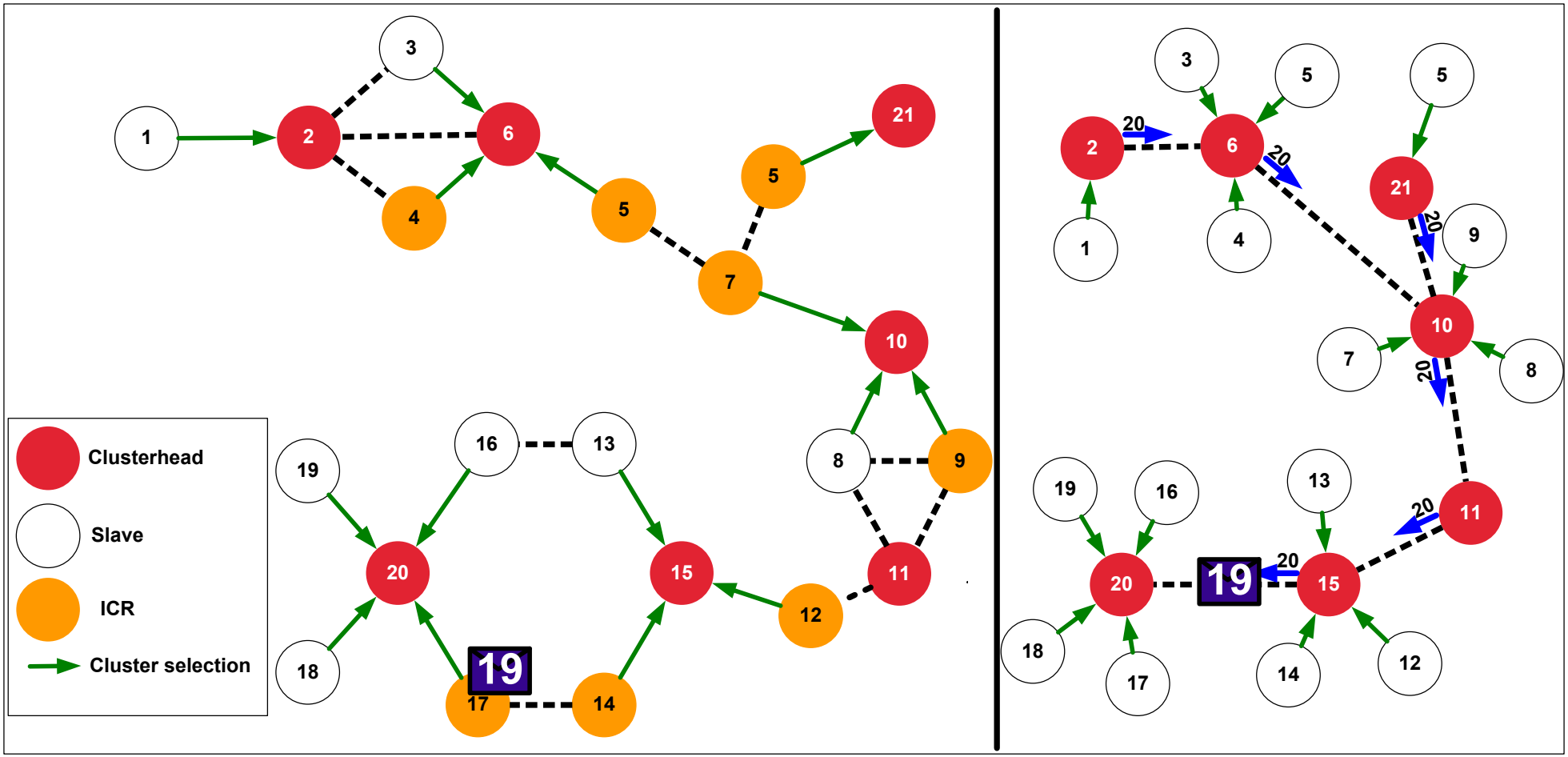
SLSR


Clusterhead
Slave
ICR
Cluster selection
19 ∈ 20 → 20
20
20
20
20
20

SLSR

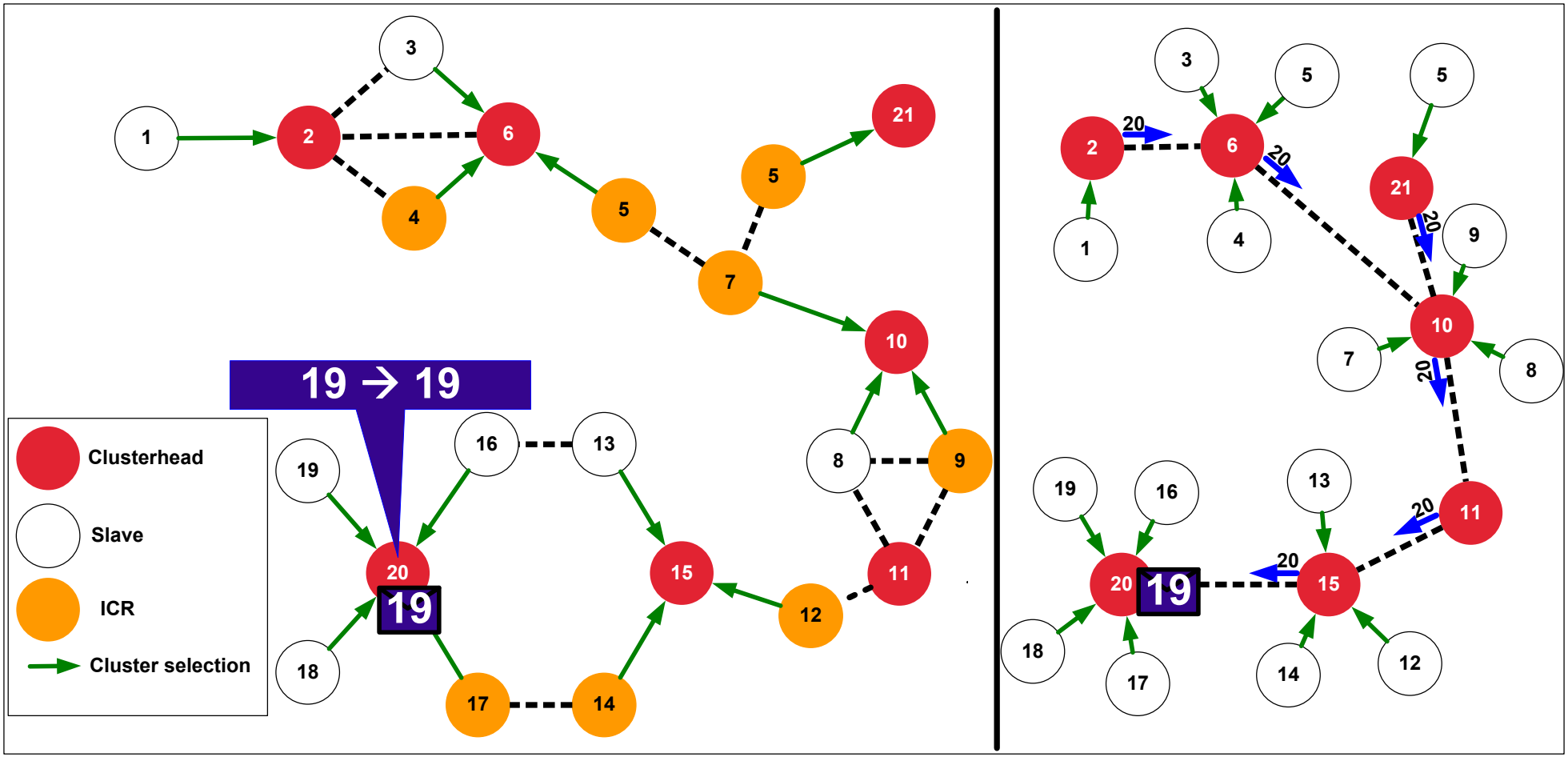

Clusterhead
Slave
ICR
Cluster selection
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
20
20
20
20
20

SLSR

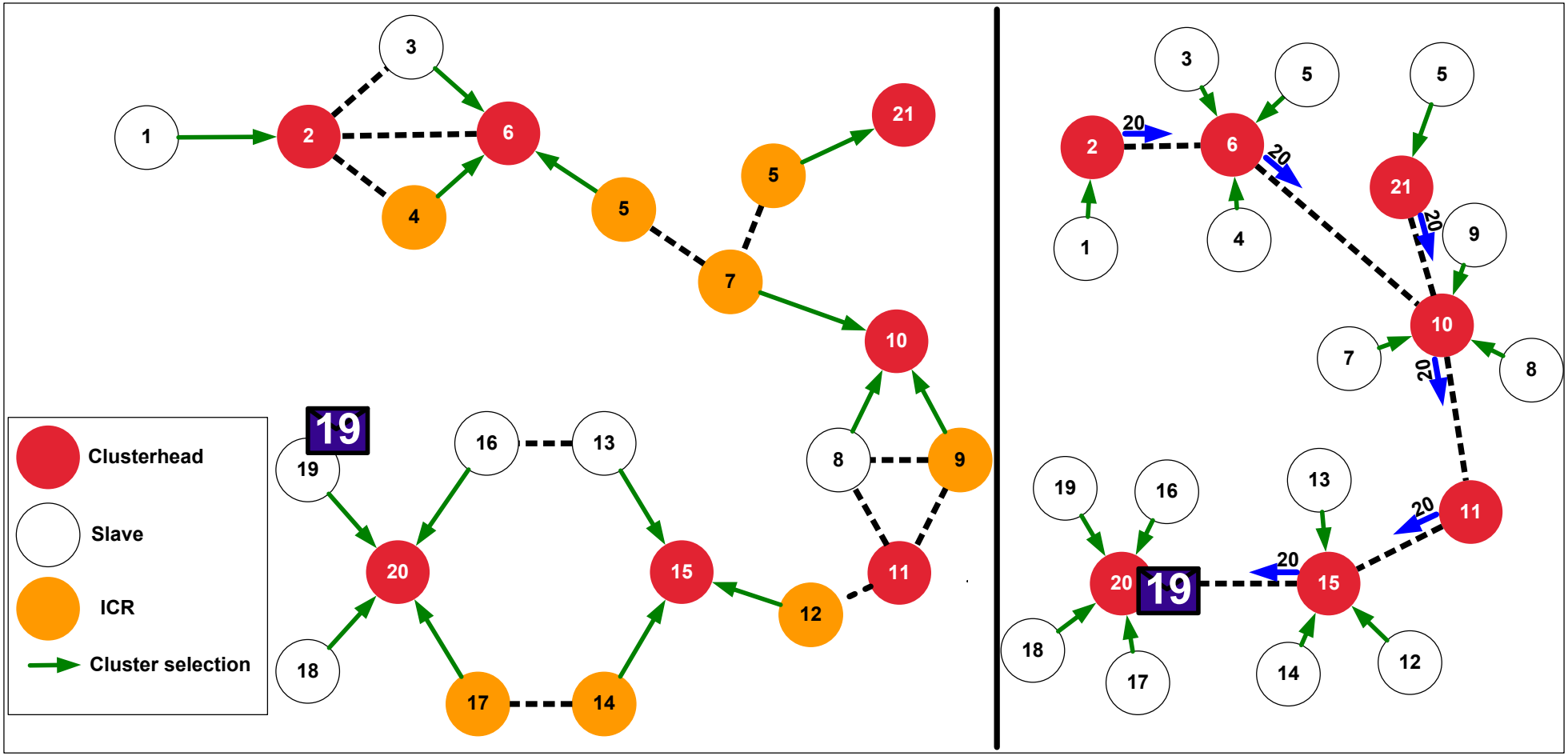

Clusterhead
Slave
ICR
Cluster selection
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
20
20
20
20
20

Stable Linked Structure Routing

SLSR



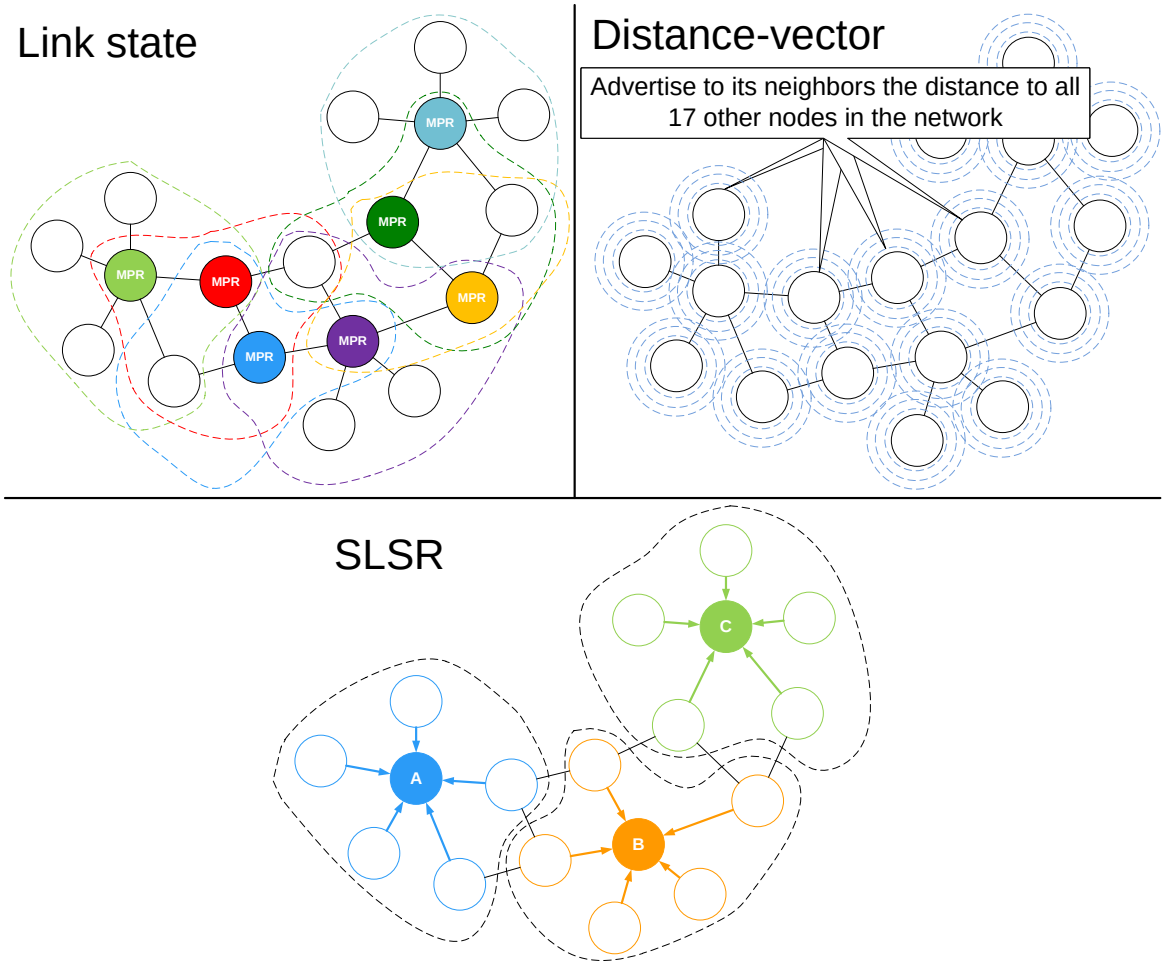
SLSR



SLSR

⇒ Wisely uses information

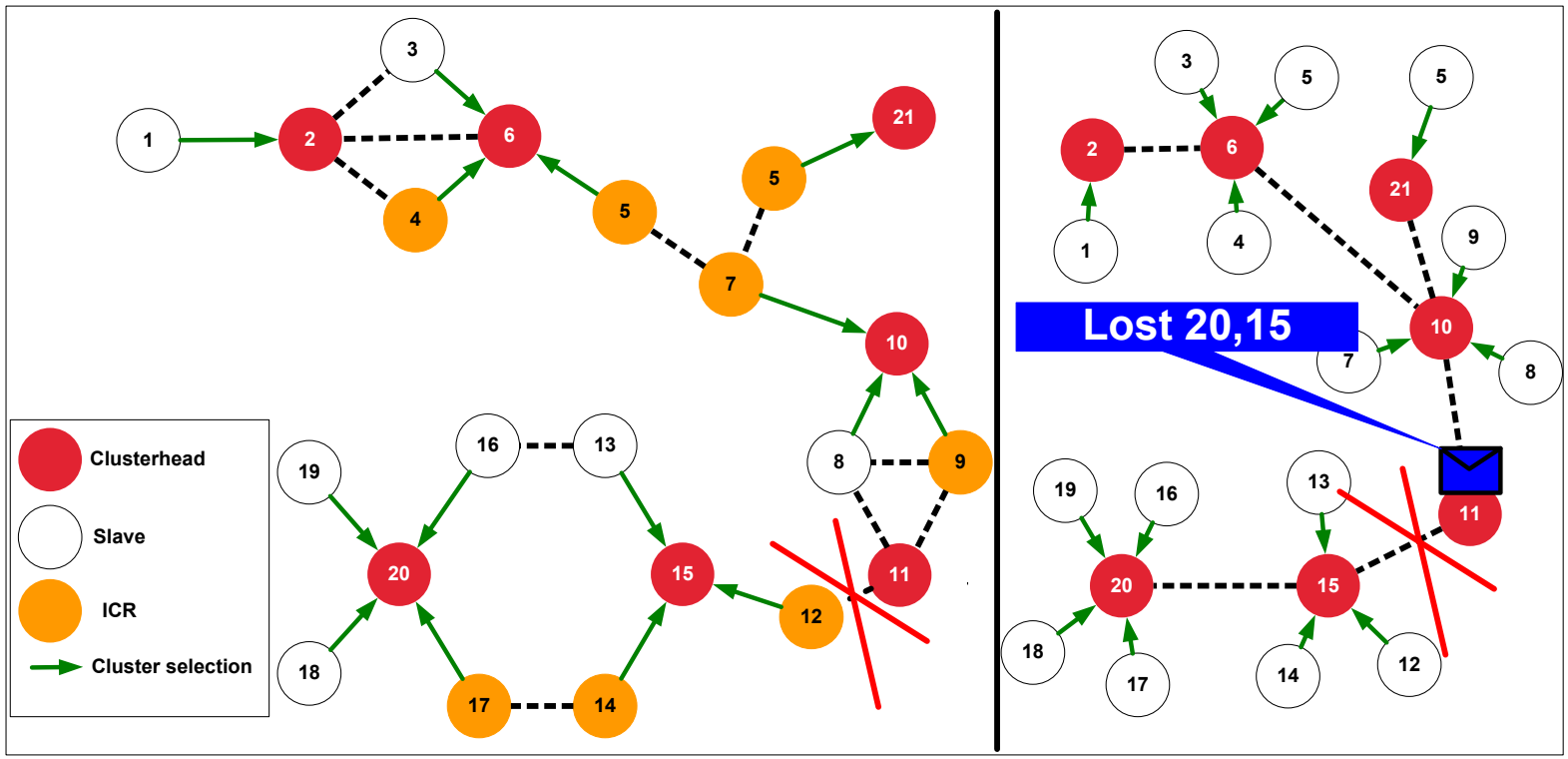
- No overlapping information



SLSR

⇒ Wisely uses information

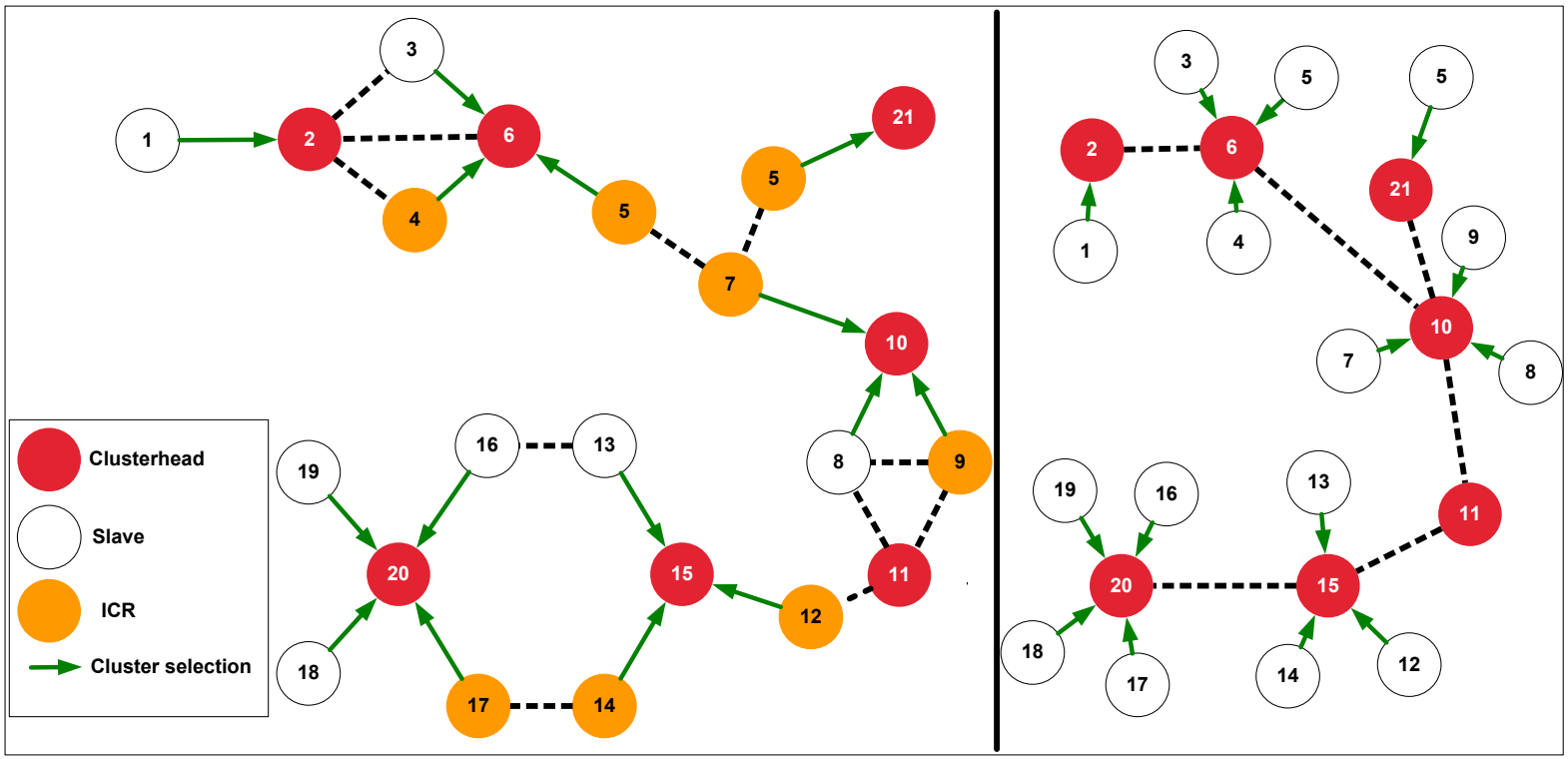
- No overlapping information
- Discarding a route to a clusterhead, also discards all its members



SLSR

⇒ Wisely uses information

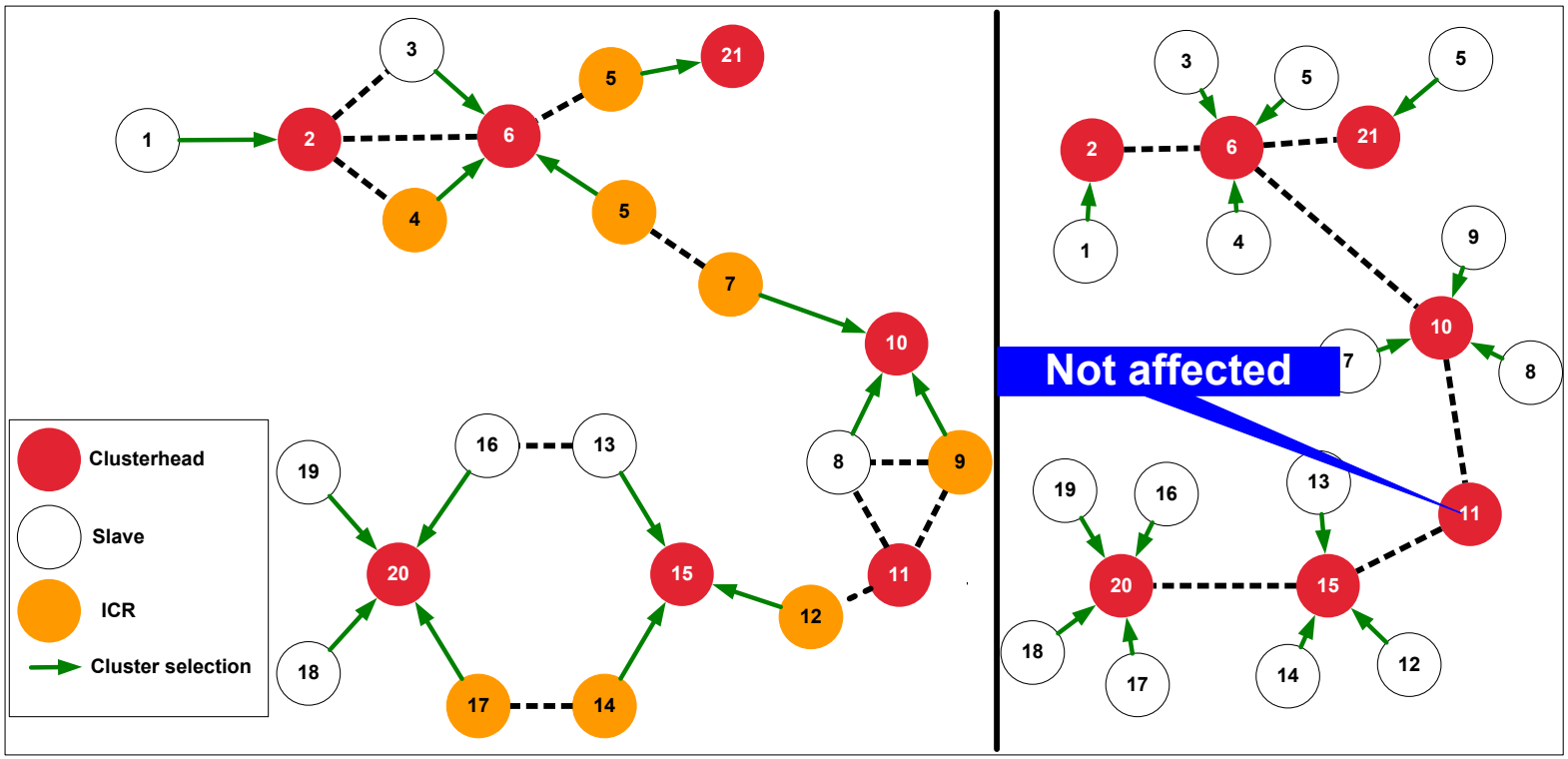
- No overlapping information
- Discarding a route to a clusterhead, also discards all its members
- If an update does not affect the route : do not forward it



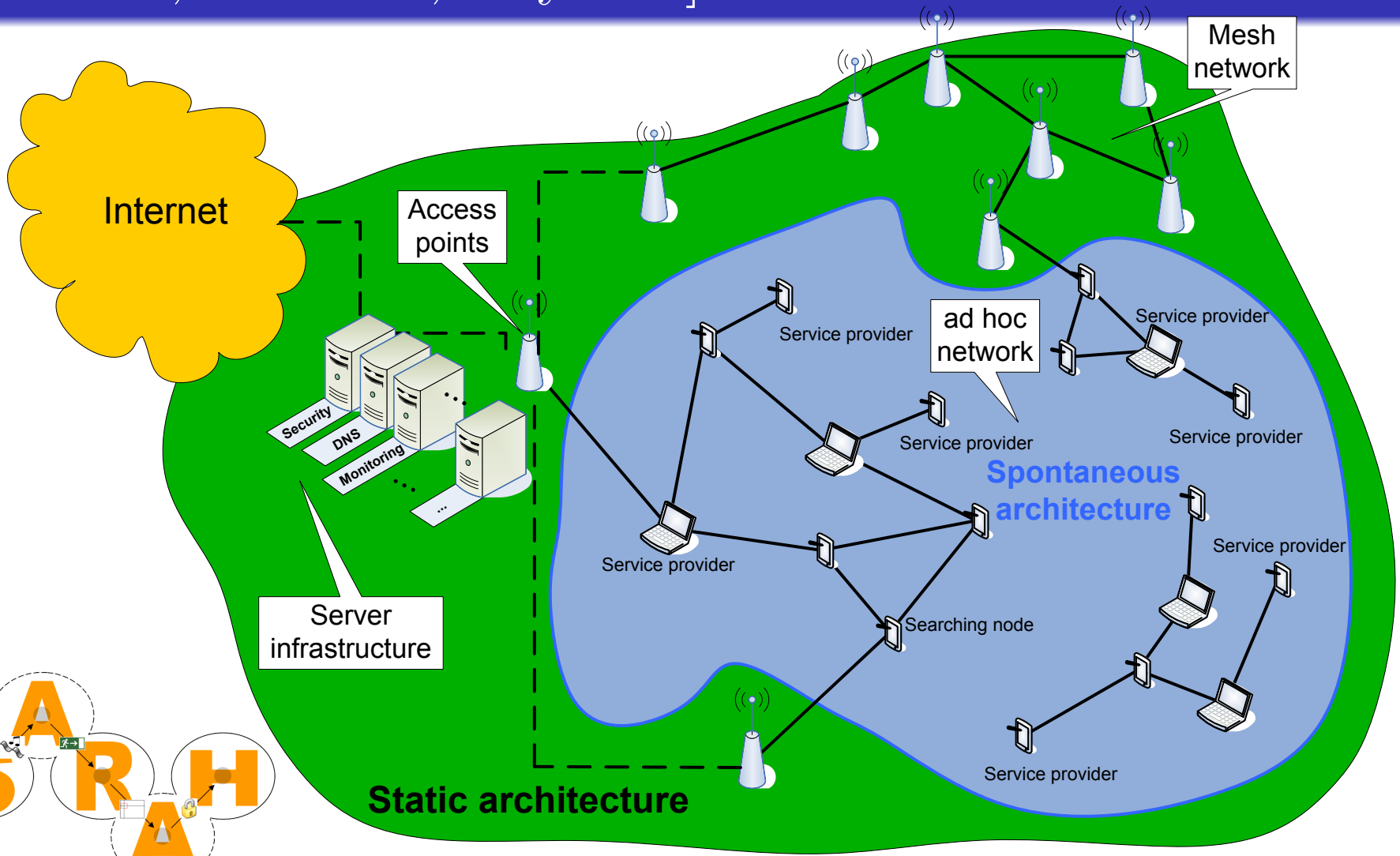
SLSR

⇒ Wisely uses information

- No overlapping information
- Discarding a route to a clusterhead, also discards all its members
- If an update does not affect the route : do not forward it



Service Discovery Architecture [Ciar 09, Lecl 10a, Reyn 10]



Find an adapted service discovery protocol

Requirements :

- Without central server
- Handles dynamicity
- Can aggregate information
- Lightweight
- Can include context awareness

Bonus option :

- Exists, well known
- Can take advantage of infrastructure

Zeroconf[Ches 05]

- Uses the DNS standard
- Without infrastructure
- Handles conflicts
- Uses Multicast IP for dissemination

3 parts :

- Auto-IP
  - Local IP address assignment mechanism
- mDNS
  - DNS name resolution
- DNS-SD
  - DNS service discovery
  - naming/DNS entry structure guidelines

Find an adapted service discovery protocol

Zeroconf

- Works without central server✓
- Handles dynamicity✓
- Aggregates information✓
- Is lightweight✓
- Can include context data✓

Bonus option :

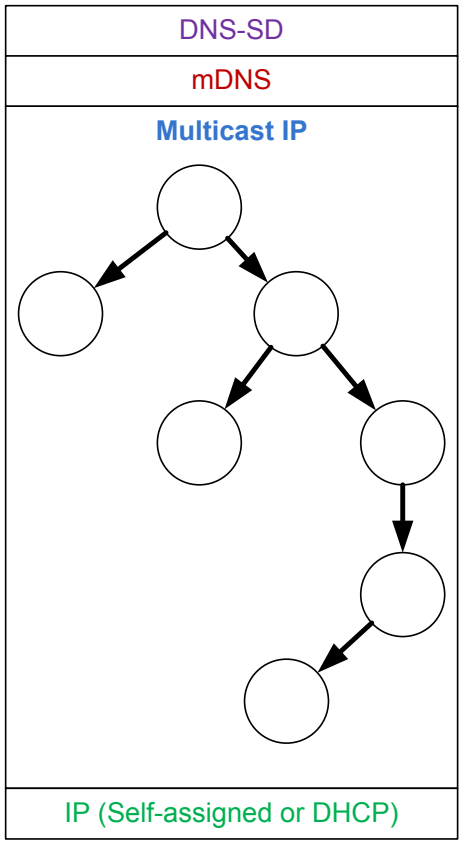
- It is a standard and very popular✓
- Can use DNS servers when available✓

Zeroconf for ad hoc networks

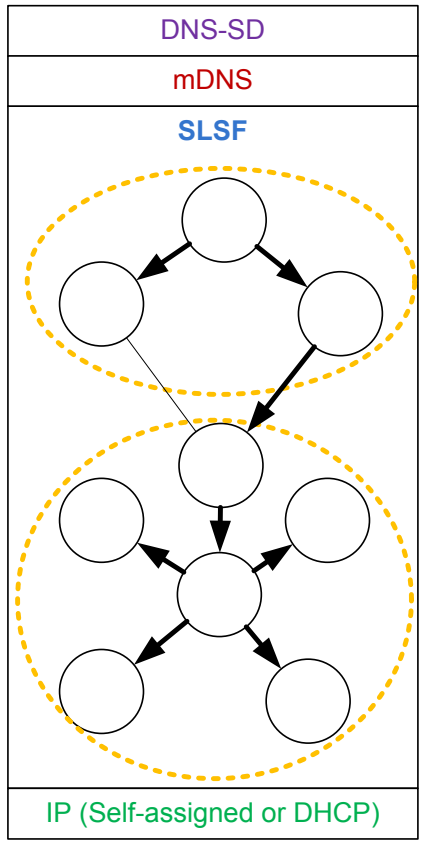
Adapt Zeroconf for ad hoc networks by :

- Replacing the multicast structure

Original Zeroconf stack



Modified Zeroconf stack



Software development and experiments on real devices

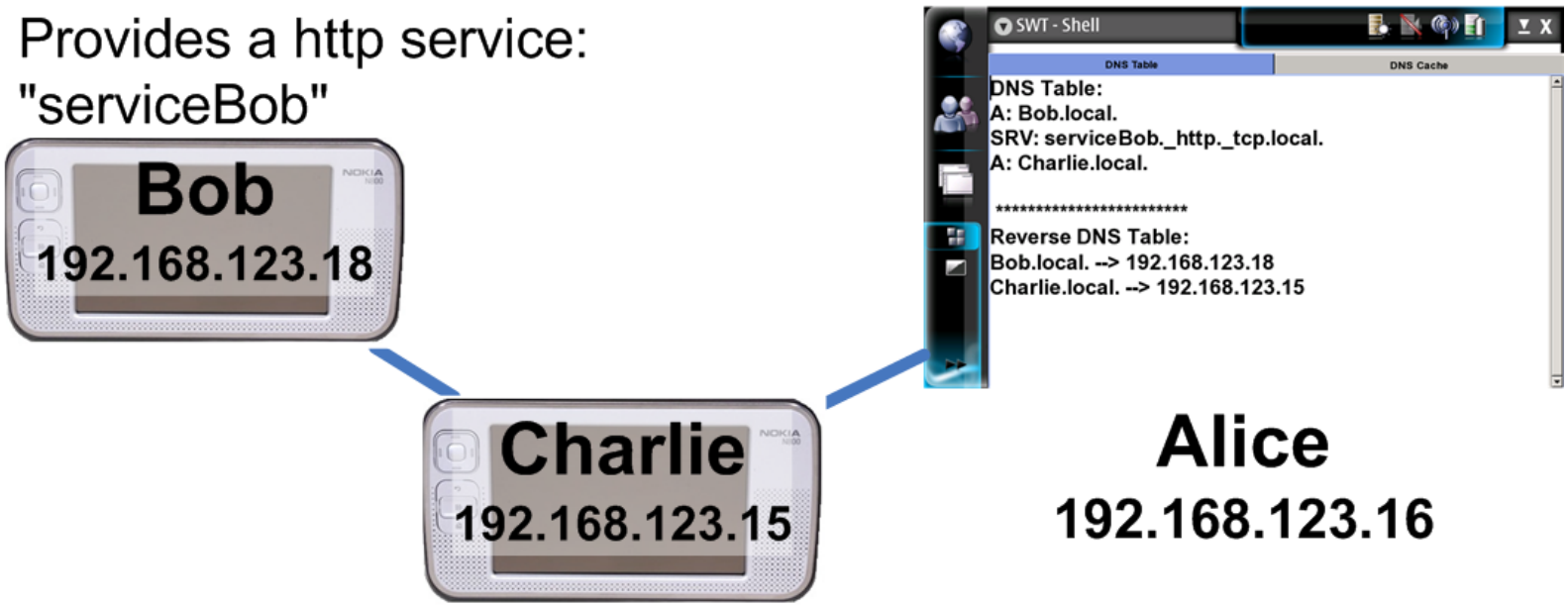
Experiments :

- N800 tablets, Windows PC, Linux
- JANE platform mode

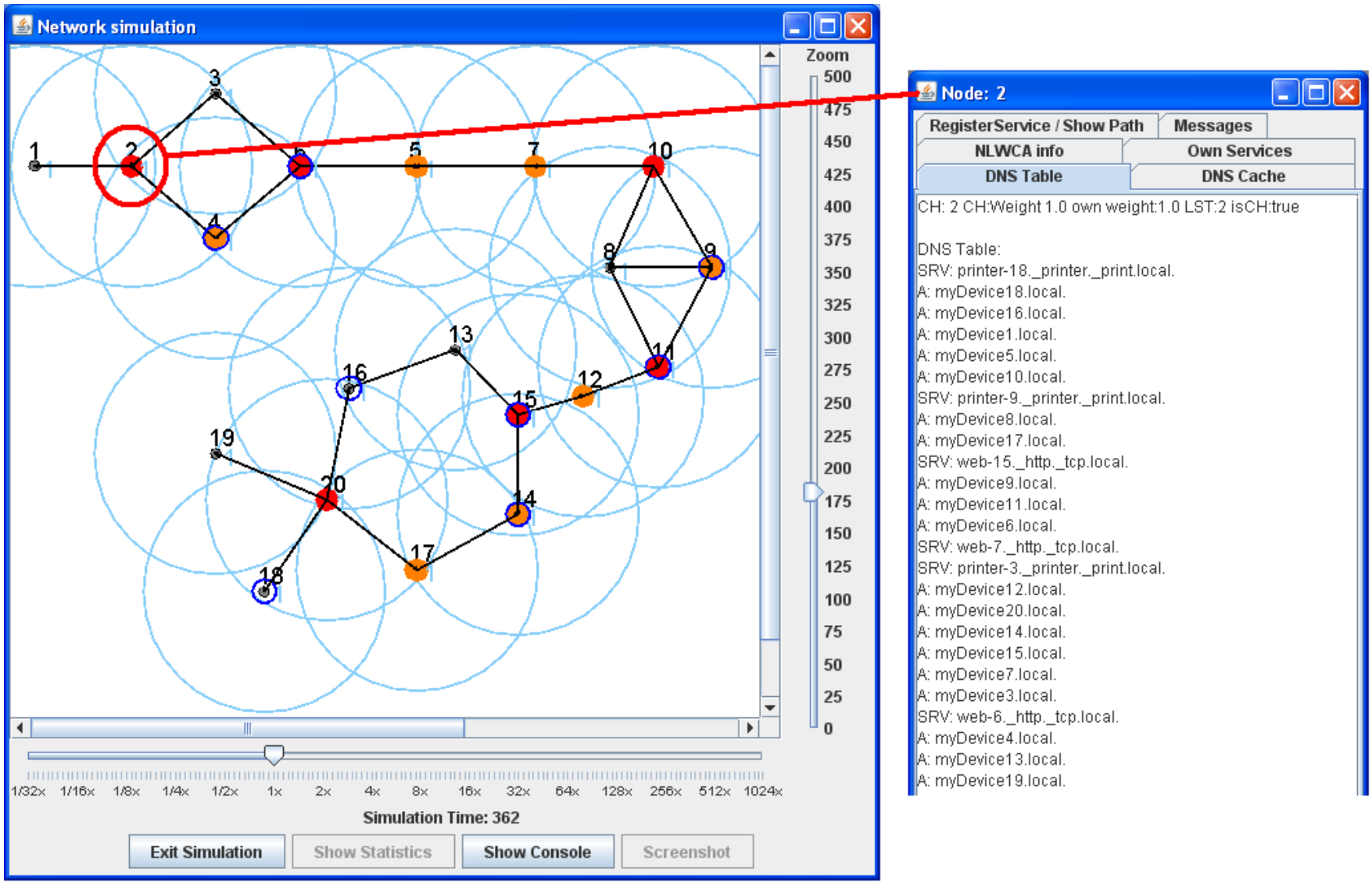
We implemented in Java :

- OLSRv2
- NLWCA modifications
- SLSF
- SLSR
- Adaptation of Zeroconf (jmDNS code)
- Example application : Chat/Http on top of SLSF

Zeroconf on ad hoc devices



Zeroconf on ad hoc devices

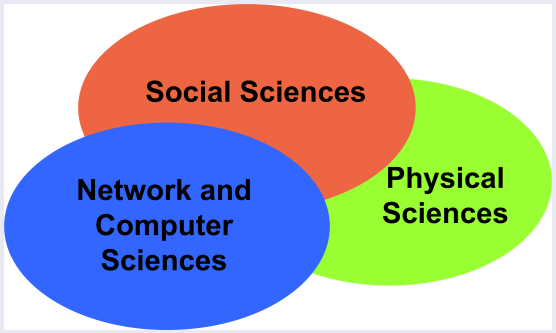


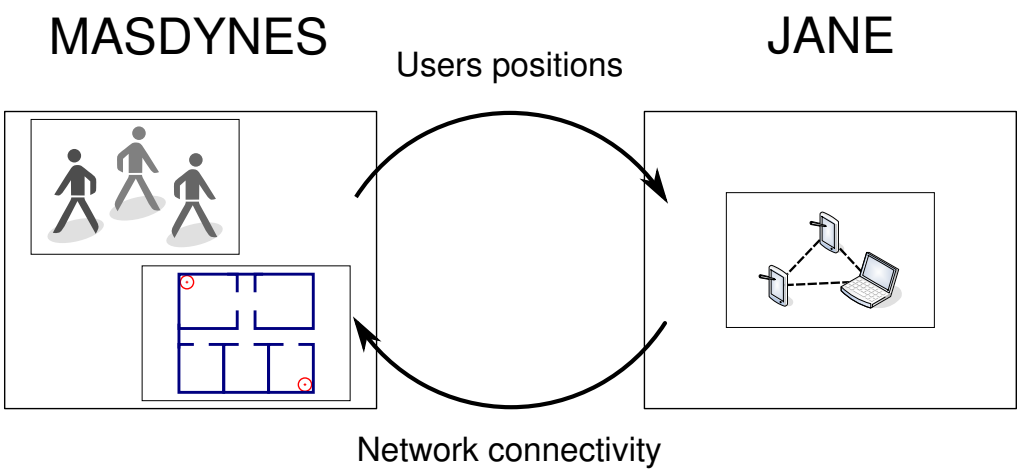
New mobility models

Couple existing simulators and models

- In a joint work with Julien Siebert
- Couple existing simulators
- From different domains

⇛ AA4MM[Sieb 10]





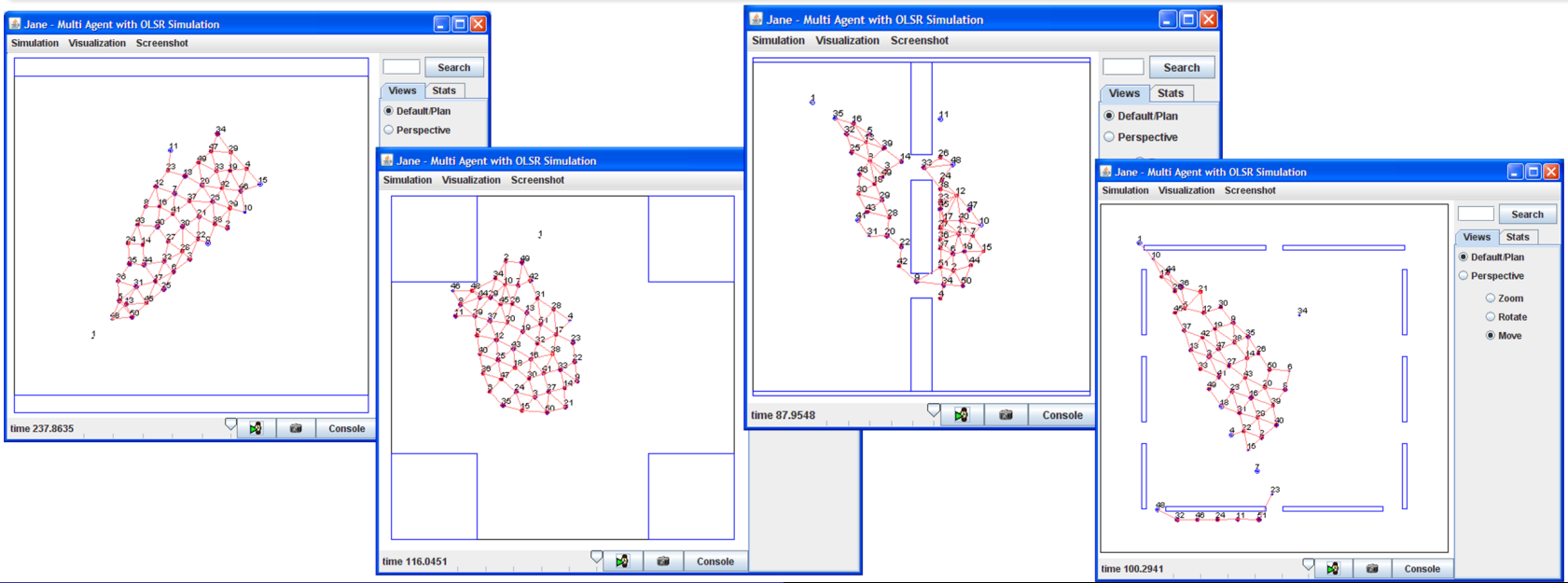
New kind of experiments[Lecl 10d]

Go further than "classic" simulations :

Evaluation, prototyping of network protocols and applications

What if :

- users react to network (QoS) and environment (obstacles) ?
- behaviors evolve (over time and space) ?



Conclusion

⇛ Propose a thorough solution for service discovery in ad hoc networks

Contribution stack :

- Dissemination
- Routing
- Research tools
- Context
  ⇛ Service discovery

That works in a symbiotic way

- Take advantage when available
- But also, work when not available

Conclusion

Service Discovery Architecture

- Involves all contributions
- Provides a solution to use ad hoc networks within a sparse infrastructure

Simulation and Experiments

- Small experiments to validate the approach and feasibility
- Simulation to measure performances

Technical aspects

- Protocols implementations
- Simulators and research tools

Perspectives

Short term

- Adapt protocols to context using metrics
- Advanced mobility models and scenarios
- Large scale experimentation on real devices

Long term

- New kind of protocols and design approaches :
  - User behavior aware
  - Scenario oriented optimistic protocols
  - Context-based protocol switching
- Other usages of the proposed structure
  - Ad hoc multicast
  - Content distribution

Thank you for your attention


A. Andronache and S. Rothkugel.
"HyTrace Backbone-Assisted Path Discovery in Hybrid Networks".
Communication Theory, Reliability, and Quality of Service, 2008. CTRQ '08. International Conference on Communication Theory, Reliability, and Quality of Service, pp. 34–40, 29 2008-July 5 2008.


A. Andronache and S. Rothkugel.
"NLWCA Node and Link Weighted Clustering Algorithm for Backbone-Assisted Mobile Ad Hoc Networks".
Networking, 2008. ICN 2008. Seventh International Conference on Networking, pp. 460–467, April 2008.


J. Blum, M. Ding, A. Thaeler, and X. Cheng.
"Connected Dominating Set in Sensor Networks and MANETs".

In : *Handbook of Combinatorial Optimization*, pp. 329–369, D.-Z. Du and P. Pardalos, Kluwer Academic Publisher, 2004.


S. Cheshire and D. Steinberg.
Zero Configuration Networking : The Definitive Guide. O'Reilly Media, Inc., 2005.


L. Ciarletta, T. Leclerc, and L. Reynaud.
"Architecture pour la découverte de services avancée". 2009.


"Optimized Link State Routing Protocol (OLSR), rfc3626. `http://www.ietf.org/rfc/rfc3626.txt`".
2003.


D. Gorgen, H. Frey, and C. Hiedels.
"JANE-The Java Ad Hoc Network Development Environment".

Simulation Symposium, 2007. ANSS '07. 40th Annual, pp. 163–176, March 2007.


T. Leclerc, L. Ciarletta, A. Andronache, and S. Rothkugel. "OLSR and WCPD as Basis for Service Discovery in MANETs".
Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM '08., pp. 184–190, 29 2008-Oct. 4 2008.


T. Leclerc, L. Ciarletta, L. Reynaud, and A. Schaff. "Prototype d'architecture de découverte de services avancée".
Research Report, 2010.


T. Leclerc, L. Ciarletta, and A. Schaff. "SLSF : Stable Linked Structure Flooding For Mobile Ad Hoc Networks".
In : *IEEE ISWPC*, Palazzo Ducale, Modena, Italy, 2010.


T. Leclerc, L. Ciarletta, and A. Schaff.
"A Stable Linked Structure Flooding for Mobile Ad Hoc Networks with Fault Recovery".
In : *WWIC*, pp. 204–215, 2010.


T. Leclerc, J. Siebert, V. Chevrier, L. Ciarletta, and O. Festor.
"Multi-modeling and co-simulation-based mobile ubiquitous protocols and services development and assessment".
In : *7th International ICST Conference on Mobile and Ubiquitous Systems - Mobiquitous 2010*, Sydney Australia, 12 2010.


L. Reynaud, F. Jan, L. Ciarletta, T. Leclerc, and G. Shahab.
"Intégration des services à la plateforme pour démonstrateurs".
Tech. Rep., 2010.

Livrable L5.2, Sous-Projet 5 (SP 5) : Expérimentations.

J. Siebert, L. Ciarletta, and V. Chevrier.
"Agents and Artefacts for Multiple Models coordination. Objective and decentralized coordination of simulators.".
In : *SAC 2010 25th Symposium on Applied Computing*, ACM, Lausanne Suisse, 2010.