Scalability of Ad Hoc Routing Protocols

(wrt mobility, size, load, etc)

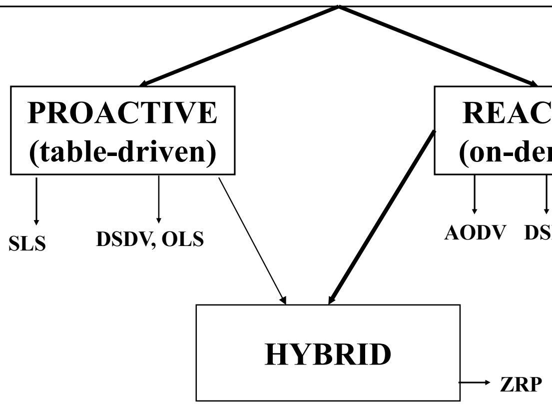
References:

C. Santivanez, R. Ramanathan, I. Stavrakakis, “Making Link-State Routing Scale for Ad Hoc Networks”, MobiHoc 2001, Oct. 4-5, 2001, Long Beach, CA, USA.

C. Santivanez, B. McDonald, I. Stavrakakis, R. Ramanathan, “ On the Scalability of Ad hoc Routing Protocols”, IEEE INFOCOM’02, N. York, USA.

C.Santivarez, R. Ramanathan, "Scalability of Routing in Ad Hoc Networks: Principles and Practise", in AD HOC WIRELESS NETWORKING, X. Chieng, X. Huang, D.-Z. Du (eds.), klywer Academic Publishers, 2003.

ROUTING ALGORITHMS



SLS: Standard Link State
DSDV: Destination Sequenced Distance Vector
OLS: Optimized Link State
AODV: Ad hoc On-Demand Distance Vector
DSR: Dynamic Source Routing
ZRP: Zone Routing Protocol

Proactive/Table-Driven

Maintain routes (paths) for all possible destinations

+ Minimum route discovery delay

– Additional induced control overhead

Meaningless in highly mobile environments

Proactive Algorithms

Routing tables constructed through:

- **Neighborhood Discovery:**
  Each node learns:
  - every neighbor's address
  - cost to communicate with each neighbor
- **Generation and propagation of Link Status control packets**
  - Control packets include one-hop neighbors and the links cost
  - Control packets are propagated over entire network (flooding)
- **Applying an algorithm for routing path construction**
  - Distance vector
  - Link state

Proactive Algorithms: Standard Link State (SLS)

Every node sends Link State Updates (LSUs)

- in case of a link status change
- periodically

Every node stores a copy of the latest LSU in a *topology table* that provides connectivity info for the entire network

The source node may apply Dijkstra's Shortest Path First (SPF) algorithm to find a route

❖Typical representative of proactive routing algorithms
❖ Simple, predictable dynamics, quick convergence

Reactive/On-Demand

Acquisition of routing info only when it is needed

+ Lower bandwidth consumption
- Dramatic increase of delay for specific applications

Generally better than proactive for highly mobile environments, but still rather inefficient

Reactive/On-Demand: Dynamic Source Routing (DSR)

A reactive algorithm - consists of two main processes:

Route Discovery: a source node S wishing to send a packet to a destination node D obtains a source route to D.

➢ It is used only when S attempts to send a packet to D and does not already know a route to it.

Route Maintenance: In case a link error is detected on a route used, source S will attempt to use any other route to D it happens to know, or it can invoke a new *Route Discovery* (to find a new one).

➢ It is used only when S is actually sending packets to D.

Proactive/Table-Driven

Maintain routes (paths) for all possible destinations

\+ Minimum route discovery delay
– Additional induced control overhead

Meaningless in highly mobile environments

Reactive/On-Demand

Acquisition of routing info only when it is needed

\+ Lower bandwidth consumption
\- Dramatic increase of delay for specific applications

Generally better than proactive for highly mobile environments, but still rather inefficient

Development of hybrid algorithms

Hybrid routing algorithms

Trying to take advantage of the efficiencies of proactive and reactive schemes

- **Advantage of proactive algorithms:**
  <u>low delay</u>
- **Advantage of reactive algorithms:**
  <u>high efficiency</u>

Zone Routing Protocol (ZRP)

A typical representative of hybrid routing algorithms

A set of k-hop neighbors constitute the node's *zone*

- The size of a zone is given by a radius of length *r*, where *r* is the number of hops to the perimeter of the zone
- The zones may be overlapping and of variable size

(proactive function) :

A node disseminates event-driven Link State Updates LSUs to its k-hop neighbors (up to k hops)

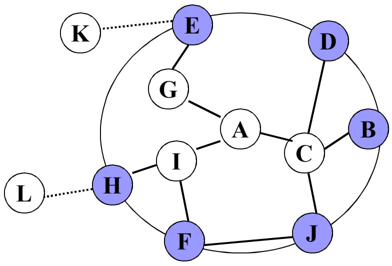
Proactive discovery within a node's local neighborhood (zones)

Zone Routing Protocol (ZRP)

(reactive function) :

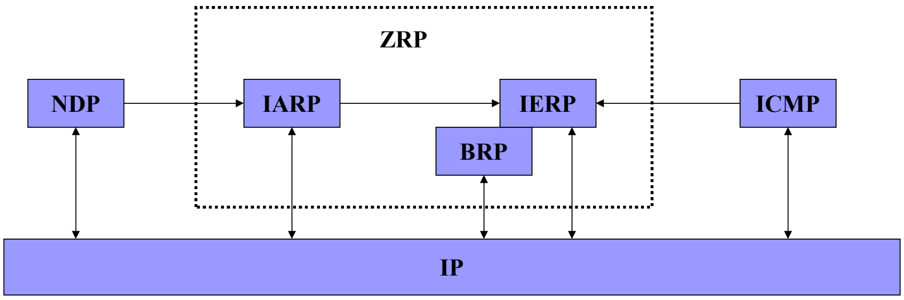
Reactive protocol for communication between the zones

Use of peripheral nodes for efficient flooding:
To forward packets outside its zone, it sends *route request* to a subset of nodes (peripheral nodes), reducing overhead



Routing zone of node A with r = 2

Components of ZRP



NDP - Neighborhood Discovery Protocol: A node needs to know about its neighbors

IARP - Intrazone Routing Protocol: Determine the routes to peripheral nodes (proactive)
IERP - Interzone Routing Protocol: Communication between different zones (reactive)
BRP - Bordercast Resolution Protocol: Efficient use of peripheral nodes

ICMP - Internet Control Message Protocol: Existing protocol to support IP-routing

Plain Flooding (PF)

- The source node broadcasts the packet to be transmitted
- The neighbors of the source node rebroadcast it once to their neighbors, and this is repeated until it reaches the destination
- Nodes must keep track of the packets previously sent, to avoid sending a packet more than once
- PF is not used for data packet delivery due to increased useless retransmissions
- Meaningful in cases of control traffic, small traffic load, small network size and high mobility rate

Scalability of Ad Hoc Routing Protocols

References:

C. Santivanez, R. Ramanathan, I. Stavrakakis, "Making Link-State Routing Scale for Ad Hoc Networks", MobiHoc 2001, Oct. 4-5, 2001, Long Beach, CA, USA.

C. Santivanez, B. McDonald, I. Stavrakakis, R. Ramanathan, " On the Scalability of Ad hoc Routing Protocols", IEEE INFOCOM'02, N. York, USA.

C.Santivarez, R. Ramanathan, "Scalability of Routing in Ad Hoc Networks: Principles and Practise", in AD HOC WIRELESS NETWORKING, X. Chieng, X. Huang, D.-Z. Du (eds.), klywer Academic Publishers, 2003.

Communication overhead

- Traditionally: *overhead* → *control overhead*
  (= amount of bandwidth required to construct and maintain a route)
  - In proactive approaches: number of packets exchanged in order to maintain the node's forwarding tables up-to-date
  - In reactive approaches: bandwidth consumed by the route request/reply messages (global or local)
- control overhead fails to include the impact of suboptimal routes

Suboptimal routes not only increase the end-to-end delay but also result in a greater bandwidth (BW) usage than required.

As the network size increases, keeping route optimality (to reduce aforementioned BW usage increase) imposes an unacceptable control overhead (i.e. corresponding BW consumption).

Communication overhead

- **reactive** protocols introduce suboptimal routes because:
  - they try to maintain the current source-destination path for as long as it is valid, although it may no longer be optimal
  - local repair techniques try to reduce the overhead induced by the protocol at the expense of longer, non optimal paths
- **proactive** protocols introduce suboptimal routes by:
  - limiting the scope of topology information dissemination (e.g. hierarchical routing) and/or
  - limiting the time between successive topology information updates dissemination
    (topology updates are no longer instantaneously event-driven)

Communication overhead

<u>Key Idea:</u>
Revise concept of *overhead*, to include the effect of suboptimal routes

The *minimum traffic load* of a network, is the minimum amount of bandwidth required to forward packets over the shortest distance (in number of hops) paths available, assuming all the nodes have instantaneous *a priori* full topology information.

- routing protocol-independent metric
- assumes that all the nodes are provided *a priori* global information
  - possible in fixed networks
  - in mobile scenarios this is hardly possible

  (even if static routes are provided it is unlikely that these remain optimal)

- facilitates/motivates the definition of *total overhead* of a routing protocol

Minimum traffic load – Total overhead

Total amount of bandwidth	=	*minimum traffic load*	+	*total overhead*
routing protocol dependent		routing protocol independent		routing protocol dependent

The ***<u>total overhead</u>*** induced by a routing protocol is the difference

between

the total amount of bandwidth actually consumed by the network operating under such routing protocol,

and

the *minimum traffic load* that would have been required should the nodes had *a priori* full topology information.

Minimum traffic load – Total overhead

The ***total overhead*** :

- provides an unbiased metric for performance comparison that reflects bandwidth consumption

- may not fully characterize all the performance aspects relevant to specific applications, but
  - bandwidth is likely to remain a limiting factor in terms of scalability
  - bandwidth is proportional to factors including energy consumption, memory and processing requirements

- The different sources of overhead may be expressed in terms of
  - *Reactive routing overhead*
  - *Proactive routing overhead*
  - *suboptimal routing overhead*

Proactive, Reactive & Suboptimal overhead

The ***reactive overhead*** of a protocol is the amount of bandwidth consumed by the specific protocol to build paths from a source to a destination, *after* a traffic flow to that destination has been generated at the source.

- In static networks is a function of the rate of generation of new flows.
- In dynamic (mobile) networks is a function of both the traffic and the rate topology change.

  (paths are (re)built not only due to new flows but also due to link failures in an already active path)

Proactive, Reactive & Suboptimal overhead

The ***proactive overhead*** of a protocol is the amount of bandwidth consumed by the protocol in order to propagate route information *before* it is needed.

The ***suboptimal routing overhead*** of a protocol is the difference between the bandwidth consumed when transmitting data from all the sources to their destinations using the routes determined by the specific protocol, and the bandwidth that would have been consumed should the data have followed the shortest available path(s).

- Example: a source that is 3 hops away from its destination

  a protocol chooses to deliver one packet following a k $(k > 3)$ hop path (for example, due to out-of-date information) $\Rightarrow$ $(k-3) * packet\_length$ bits will be added to the suboptimal RO

Achievable Regions & Operating points

- The 3 different overhead sources are locked in a 3-way trade-off
- in an already efficient algorithm, the reduction of one of them will most likely cause the increase of one of the others
- For example, reducing the ‘zone’ size in ZRP:
  - will reduce ZRP’s proactive overhead, but
  - will increase ZRP’s reactive overhead

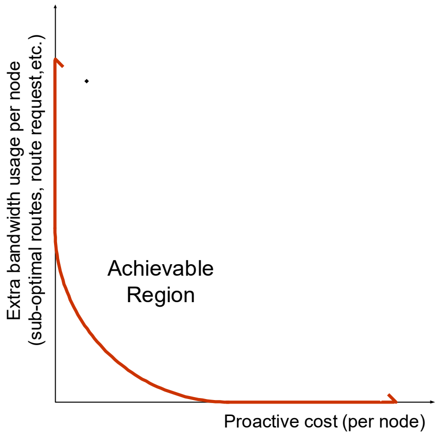
***achievable region* of overhead**:

The three dimensional region formed by all the values of proactive, reactive, and suboptimal routing overheads that can be induced by any protocol under the same scenario (traffic, mobility, etc.)

Achievable Regions & Operating points

Figure: 2-dimensional transformation of the 'achievable region'

- horizontal axis: proactive overhead induced by a protocol
- vertical axis: reactive + suboptimal routing overheads

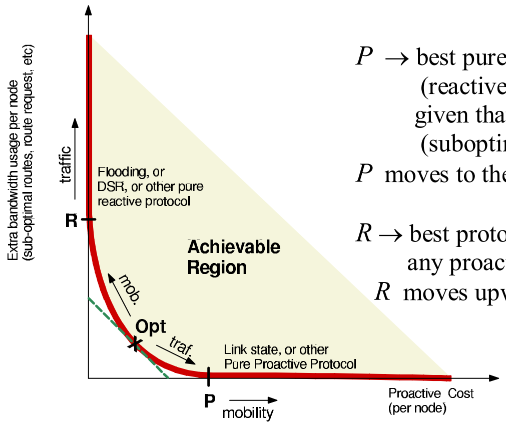


Achievable Regions & Operating points

the achievable region

- is convex
  - consider the points P_1 and P_2 achieved by protocols P_1 and P_2
  - any point $\lambda P_1 + (1-\lambda) P_2$ can be achieved by engaging protocol P_3 that behaves
    - as protocol P_1 a fraction λ of a (long) time
    - as protocol P_2 the remaining of the time
- is lower-bounded by the curve of overhead points achieved by the 'efficient' protocols
  ('efficient' protocols: minimize some source of overhead given a condition imposed on the others)

Achievable Regions & Operating points



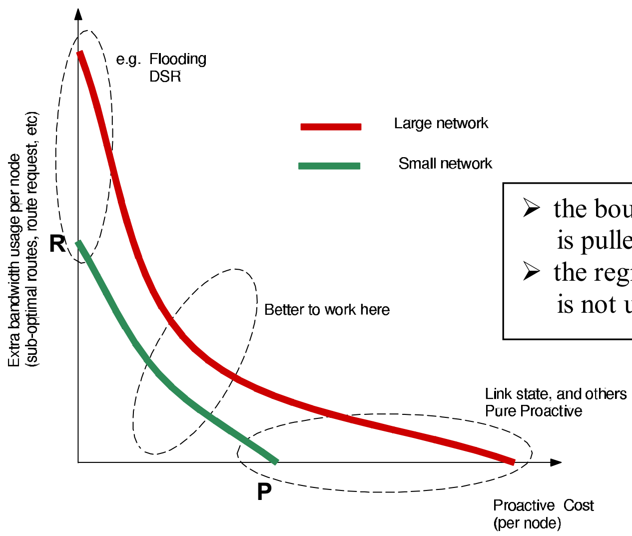
$P \rightarrow$ best pure proactive approach
(reactive overhead = 0)
given that optimal routes are required
(suboptimal overhead= 0)
P moves to the right as mobility increases

$R \rightarrow$ best protocol that does not use
any proactive information
R moves upward as traffic increases or
diversifies

the best protocol (in terms of overhead)	$\rightarrow$	minimizes *total overhead*	$\Leftrightarrow$	achieves point *Opt*

point *Opt* : point tangent to the curve $x + y = constant$

Effect of increasing the network size



- the boundary region is pulled up
- the region displacement is not uniform

Effect of increasing the network size

- points P and R increase proportionally to $\Theta(N^2)$

Pure proactive protocols (SLS)
[Pure reactive algorithms (DSR without the route cache option)]
generate a control message [a route request (RREQ) message]
each time a link changes (worst case) [a new session is initiated]

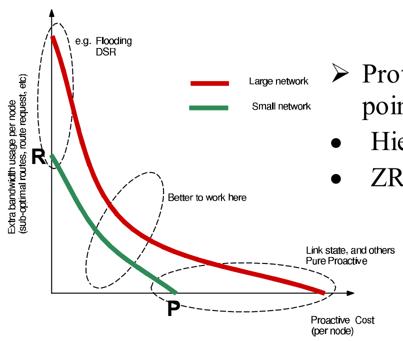
Each control message [RREQ message] is retransmitted by each node

The generation rate of control messages [RREQ messages] increases linearly with network size (N)

The number of message retransmissions increases linearly with N

$\Rightarrow$ The total overhead increases as rapidly as N^2

Effect of increasing the network size



- Protocols inducing ‘intermediate points’:
  - Hierarchical link state (HierLS): $N^{1.5}$
  - ZRP: $N^{1.66}$

As size increases, the best operating point is far from the extreme points ***P*** *and* ***R*** *and in the region where the proactive, reactive, and suboptimal routing overheads are balanced.*

Network's scalability (Qualitatively)

Scalability is the ability of a network to support the increase of its limiting parameters.

Limiting parameters of a network are those parameter whose increase causes the network performance to degrade.

For example:

- mobility rate
- traffic rate
- network size
- network density

Network's scalability (Quantitatively)

Let $Tr(\lambda_1, \lambda_2, \ldots)$ be the ***minimum traffic load*** experienced by a network under parameters $\lambda_1, \lambda_2, \ldots$. Then, the ***network scalability factor*** of such a network, with respect to a parameter λ_i (Ψ_{λ_i}) is:

$$\Psi_{\lambda_i} \overset{\text{def}}{=} \lim_{\lambda_i \to \infty} \frac{\log Tr(\lambda_1, \lambda_2, \ldots)}{\log \lambda_i}$$

- relates the increase in network load to the different network parameters
- may be used to compare the scalability properties of different networks (wireline, mobile ad hoc, etc.)

- For the class of mobile ad hoc networks defined by assumptions a.1-a.8 the *minimum traffic load* $Tr(\lambda_{lc}, \lambda_t, N) = \Theta(\lambda_t N^{1.5})$

$\Rightarrow \quad \Psi_{\lambda_{lc}} = 0$, $\Psi_{\lambda_t} = 1$, and $\Psi_N = 1.5$

Network's scalability (Quantitatively)

- Define the ***network rate*** to assess ***network scalability*** w.r.t. λ_i

The *network rate* $R^{network}$ of a network is the maximum number of bits that can be simultaneously transmitted in a unit of time.

To compute the *network rate* ($R^{network}$), ALL successful link layer transmissions must be counted, regardless of whether the link layer recipient is the final network-layer destination or not.

Network's scalability (Quantitatively)

A network is said to be ***scalable*** with respect to the parameter λ_i if and only if, as the parameter λ_i increases, the network's ***minimum traffic load*** does not increase faster than the ***network rate*** ($R^{network}$) can support. That is, if and only if:

$$\Psi_{\lambda_i} \leq \lim_{\lambda_i \to \infty} \frac{\log R^{network}(\lambda_1, \lambda_2, \ldots)}{\log \lambda_i}$$

Network's scalability (Quantitatively)

Scalability of Mobile Ad Hoc Networks (with respect to N):

It has been shown [Gupta, Kumar 2000] that in mobile ad hoc networks *Θ(N)* successful transmissions can be scheduled simultaneously. I.E., the *network rate* is *Θ(N)* ➔ log R / log N → 1 . Thus, to be regarded as scalable with respect to network size

$$\Psi_N \leq 1 \text{ (* condition for mobile adhoc nets)}$$

For example, for the class of networks considered here (assumptions a.1-a.8)

$$\Psi_N = 1.5$$

⇒ **networks under study are not *scalable* w.r.t. to network size**

Network's scalability (Quantitatively)

Scalability of Mobile Ad Hoc Networks (with respect to N)
Delay Tolerant Networks:

- average path length may be reduced to 2 (*Θ(1)*
  provided that applications can tolerate infinite delivery delays and mobility pattern is random ([Groossglauser&TSE, Infocom 2001])
- thus, network *scalability factor* with respect to network size Ψ_N is equal to 1. (N nodes X traffic rate X constant path length)

Thus, previous condition * is satisfied. Those ad hoc networks (random mobility and capable of accepting infinitely long delays) are the only class of ad hoc networks that are scalable with respect to network size.

Network's scalability (Quantitatively)

<u>**Scalability of Wireline fully connected Networks (with respect to N)**</u>

They may have $\Psi_N = 1$, thus potentially scalable w.r.t. N

- 1 link-load is needed for any transmission; thus, $\Psi_N = 1$.
- Also based on possible simultaneous transmissions, the network rate is also *Θ(N)*)

(However, this scalability requires the nodes' degree to grow without bound, which may be prohibitively expensive)

Network's scalability (Quantitatively)

➢ ***mobile networks:*** the *network rate* does not increase with mobility or traffic load

Recall: [Gupta, Kumar 2000]in mobile ad hoc networks *Θ(N)* successful transmissions can be scheduled simultaneously. I.E., the *network rate* is

Θ(N) --> logR/log* = 0, *=mobility or traffic load)

$\Rightarrow$ the network will be scalable w.r.t. mobility $\Leftrightarrow$ $\Psi_{\lambda_{lc}} = 0$ (=network rate)
w.r.t. traffic $\Leftrightarrow$ $\Psi_{\lambda_t} = 0$ (=network rate)

$\Rightarrow$ the networks considered here ($\Psi_{\lambda_{lc}} = 0$, $\Psi_{\lambda_t} = 1$, and $\Psi_N = 1.5$)
- **are *scalable* w.r.t. mobility**
- **but are not *scalable* w.r.t. traffic**

Network's scalability (Quantitatively)

Similar conclusions for scalability w.r.t. other parameters:

For example, w.r.t. **transmission range**

- as transmission range ℓ increases the spatial reuse decreases as ℓ^2, since area is covered with the square of ℓ (assuming an infinite size network with regular density)
- *network rate* decreases as rapidly as ℓ^2 (reduction of space available for additional, non-interfering transmissions)

$\Rightarrow$ Ψ_ℓ should be lower than -2 for the network to be *scalable* (minimum traffic rate should decrease by 2 orders of magnitude, to compensate for the above network rate reduction)

Network's scalability (Quantitatively)

$\Rightarrow$ Ψ_ℓ should be lower than -2 for the network to be *scalable*

BUT *the minimum traffic load* decreases only linearly w.r.t. $\ell \Rightarrow \Psi_\ell = -1$

(since the length of a path has linear dependence on the transmission range, a transmission range ℓ increase leads to a path reduction that is linear with ℓ)

➔ad hoc networks are not scalable w.r.t. transmission range

Thus, better try not to increase transmission range beyond what is needed to maintain good connectivity (e.g., a lower bound on network degree)

focus on networks with power control
(the transmission range changes so that the network degree is kept bounded

Routing Protocol's scalability (Qualitatively)

- mobile ad hoc *networks* are not *scalable* with respect to size
- How *about routing protocol scalability?*

Routing protocol's scalability is the ability of a routing protocol to support the continuous increase of the network parameters without degrading network performance.

- the network's own scalabilty properties provide the reference level as to what to expect of a routing protocol
- If the **overhead** induced by a routing protocol **grows slower than *the minimum traffic load*** (network property) then the routing protocol is not degrading network performance, even if the overhead happens to grow faster than the *network rate.*

Routing Protocol's scalability (Quantitatively)

Let $X_{ov}(\lambda_1, \lambda_2, ...)$ be the *total overhead* induced by routing protocol X, dependent on parameters $\lambda_1, \lambda_2, ...$ (e.g. network size, mobility rate, data generation rate, etc.).

Protocol X's ***routing protocol scalability factor*** $\rho^X_{\lambda_i}$ with respect to a parameter λ_i , is defined to be :

$$\rho^X_{\lambda_i} \stackrel{\text{def}}{=} \lim_{\lambda_i \to \infty} \frac{\log X_{ov}(\lambda_1, \lambda_2, ...)}{\log \lambda_i}$$

- provides a basis for comparison among different routing protocols

Routing Protocol's scalability (Quantitatively)

(to assess whether a routing protocol is *scalable* the following definition is used)

A routing protocol X is said to be ***scalable* with respect to parameter** λ_i if and only if, as λ_i increases, the *total overhead* induced by such protocol (X_{ov}) does not increase faster than the network's *minimum traffic load.*

That is, if and only if:

$$\rho_{\lambda_i}^{X} \leq \Psi_{\lambda_i}$$

For the class of network under study, a routing protocol X is:

- *scalable* w.r.t. network size $\Leftrightarrow \rho_N^X \leq 1.5$
- *scalable* w.r.t. traffic $\Leftrightarrow \rho_{\lambda_t}^X \leq 1$
- *scalable* w.r.t. mobility rate $\Leftrightarrow \rho_{\lambda_{lc}}^X \leq 0$

Scalability dimensions

- network size is a key parameter, but not the only one
  (mobility, network density, network diameter, traffic diversity, energy etc.)
- Examples :
  diameter ↑ => latency for control ↑ => inconsistent routes, instability↑
  density↑ spatial reuse ↓ => capacity ↓

dimension / layer	Size	Mobility	Density	Diameter
Transport		×		×
Network	⊗	⊗	×	×
Link/MAC		×	×	
Physical		×		

Key scalability dimensions and their effect on the lower four layers (in addition to traffic)

Scalability dimensions

- size, density, diameter, and transmission range are related.
  Given size & density, different transmission power levels ($\downarrow$)
  result in different (node degree ($\downarrow$), network diameter ($\uparrow$))

- in order to increase the overall network performance, the average node degree must remain bounded (provided (bi-)connectivity)

- density dimension can be addressed by means of effective topology (transmission power) control algorithms

- Focus on: network size, mobility & traffic load assuming topology control

Network model: Notations

N : number of nodes in the network

d : average in-degree

L : average path length over all source destination pairs

λ_{lc} : expected number of link status changes that a node detects (/sec)

λ_t : average traffic rate that a node generates in a second (in bps)

λ_s : average number of new sessions generated by a node in a second

Standard asymptotic notations

- $f(n) = \Omega(g(n))$ if $\exists\ c_1,\ n_1 :\ c_1 g(n) \leq f(n)\ ,\ \forall\ n \geq n_1$
- $f(n) = O(g(n))$ if $\exists\ c_2,\ n_2 :\ f(n) \leq c_2 g(n)\ ,\ \forall\ n \geq n_2$
- $f(n) = \Theta(g(n))$ if $\exists\ c_1, c_2$, and $n_0 :\ c_1 g(n) \leq f(n) \leq c_2 g(n)\ ,\ \forall\ n \geq n_0$

c_1, c_2 constants

Assumptions

- **a.1** **As the network size increases, the average in-degree** d **remains constant.**

- imposing a fixed degree in a network is:
  - desirable (density increase jeopardizes the achievable throughput)
  - achievable (through effective power control mechanisms)

- a topology control algorithm should:
  - make the density as small as possible
  - without compromising (bi)connectivity

Assumptions

a.2 : **Let A be the area covered by the N nodes of the network, and $\sigma = N/A$ be the network average density. Then, the expected (average) number of nodes inside an area A_1 is approximately $\sigma * A_1$.**

- on large scales uniformity is expected to increase
  (for example, it is expected that half the area contains approx. ½ of the nodes)
- focus on expected (mean) behavior
  (for a specific network topology this assumption may not hold)

Assumptions

a.2 Let A be the area covered by the N nodes of the network, and $\sigma = N/A$ be the network average density. Then, the expected (average) number of nodes inside an area A_1 is approximately $\sigma * A_1$.

One can talk about the `geographical' and `topological' regions.

- In the `geographical' (large-scale) region, geographical-based reasoning shapes routing decisions.
- In the `topological' region, it is the actual link connectivity (topology) that drives the routing decisions, and geographical insights are less useful.

Assumptions

a.3 : The number of nodes that are at distance of k or less hops away from a source node increases (on average) as $\Theta(d*k^2)$. The number of nodes exactly at k hops away increases as $\Theta(d*k)$.

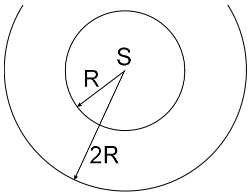
a.4 : The maximum and average path length (in hops) among nodes in a connected subset of n nodes both increase as $\Theta(\sqrt{n})$. In particular, the maximum and the average (L) path length across the network increase as $\Theta(\sqrt{N})$.

- a.3 and a.4 are based on a.2

Assumptions

a.3 , a.4

- a.3 and a.4 are based on a.2



example: a circular area centered at node S of radius R with n nodes

If $R \to 2R$

- covered area, number of nodes inside the area $\to$ 4R*2
- distance (in meters) from S to the farthest nodes $\to$ 2R
- distance (in hops) from S to the farthest nodes $\to$ 2R
  (assuming that the transmission range of the nodes does not change)
- `boundary' area (where the nodes farthest away from S are) $\to$ 2R

Assumptions

a.5 : The traffic that a node generates in a second (λ_t), is independent of the network size N (number of possible destinations). As the network size increases, the total amount of data transmitted/received by a single node will remain constant but the number of destinations will increase (the destinations diversity will increase).

a.6 : For a given source node, all possible destinations ($N-1$ nodes) are equiprobable and – as a consequence of a.5 – the traffic from one node to every destination decreases as $\Theta(1/N)$.

- a.5 & a.6 : As the network size increases the total amount of traffic generated by a single user typically diversifies rather than increases.
- a.5 & a.6 are first order approximations motivated by observed behavior with existing networks (human users)
- some other networks may violate these assumptions

Assumptions

a.5 & a6

Examples:

- low-cost long distance service: a user speaks with more friends (wherever they are), but does not increase the total time the user has to spare for personal phone calls
- increase in size and content of the Internet: a user may find more web pages (destination set diversifies) but he/she will limit the total time (and traffic) spent on the Internet

- Sensor networks **may violate** these assumptions
  - each node may broadcast its information to all other nodes (λ_t increases as $\Theta(\mathrm{N})$)
  - or, may transmit to a central node (causing the destination set to consist of only 1 node, violating a.6)

Assumptions

a.6

➢ **Traffic assumption largely determines the effect of suboptimal routing on performance.**

- limited to the locality of the source ⇒ hierarchical routing, ZRP, and NSLS will benefit
- small set of destinations will favor algorithms such as DSR
- uniform traffic tends to favor proactive approaches such as link state

➢ Equally distributed traffic (a.6) tends to pose the most demanding requirements on a routing protocol.

➢ **A protocol that is scalable (with respect to traffic) under a.6, will also be scalable under any other traffic pattern.**

Assumptions

a.7 : Link status changes are due to mobility. λ_{lc} is directly proportional to the relative node speed.

➢ assumption: short-lived link degradation will not trigger updates

short-term variations in link quality can be offset by link control mechanisms
for example:

- by requiring a high fading margin before declaring a link up
- by waiting for several seconds before declaring a link down

➢ wireless channel is quite unpredictable and long-lived link degradation is possible without mobility

Assumptions

a.8 : Mobility models : time scaling.

Let $f_{1/0}(x,y)$ be the probability distribution function of a node position at time 1 second, given that the node was at the origin $(0,0)$ at time 0. Then, the probability distribution function of a node position at time t given that the node was at the position (x_{t_0}, y_{t_0}) at time t_0, is given by

$$f_{t/t_0}(x,y,x_{t_0},y_{t_0}) = \frac{1}{(t-t_0)^2} f_{1/0}\left(\frac{x-x_{t_0}}{t-t_0}, \frac{y-y_{t_0}}{t-t_0}\right).$$

- motivated by mobility models where the velocity of a mobile over time is highly correlated

  for example, this is the case if the unknown speed and direction are constant
- does not hold for a random walk model

induces smaller node displacements over time ($\Theta(\sqrt{t})$, (t : elapsed time)

- focus on the most demanding scenario (larger displacements)

Routing protocol scalability conditions for the networks subject to assumptions a.1-1.8

- For the class of mobile ad hoc networks defined by assumptions a.1-a.8 the *minimum traffic load* $Tr(\lambda_{lc}, \lambda_t, N) = \Theta(\lambda_t N^{1.5})$ and thus :

$$\Psi_{\lambda_{lc}} = 0,\ \Psi_{\lambda_t} = 1, \text{ and } \Psi_N = 1.5$$

- Consequently, for the class of network considered here, a routing protocol X is:
  - *scalable* w.r.t. mobility rate $\Leftrightarrow$ $\rho^X_{\lambda_{lc}} \le 0$
  - *scalable* w.r.t. traffic $\Leftrightarrow$ $\rho^X_{\lambda_t} \le 1$
  - *scalable* w.r.t. network size $\Leftrightarrow$ $\rho^X_N \le 1.5$

Plain Flooding (PF)

- PF reactive RO = PF proactive RO = 0
- PF suboptimal RO / sec = $\Theta(\lambda_t(N^2 - N^{1.5})) = \Theta(\lambda_t N^2)$
  - $\lambda_t N$ data packets are generated / sec
  - $N-1$ transmissions / packet
    (is (re)transmitted by every node, except the destination)
  - optimal value (on average) = L, $L = \Theta(\sqrt{N})$ (a.4)
    (up to L hops broadcast needed, the rest is OH)
  - $\Rightarrow$ additional bandwidth required = $data\,(N-1-L)\lambda_t N$ bps
- PF total RO /sec = $\Theta(\lambda_t * N^2)$

$\Rightarrow$ $\rho_{\lambda_{lc}}^{PF} = 0$, $\rho_{\lambda_t}^{PF} = 1$, $\rho_N^{PF} = 2$

Standard Link State (SLS)

- SLS reactive RO = SLS suboptimal RO = 0
- SLS proactive RO / sec = $lsu\,\lambda_{lc}N^2$ bps = $\Theta(\lambda_{lc}N^2)$

lsu := size of the LSU (Link State Update) packet

- $N\lambda_{lc}$ LSUs are generated /sec (on average)
  (each node generates an LSU at a rate of λ_{lc} / sec)
- overhead of $lsu\,N$ bits / LSU
  (each LSU is transmitted at least N times (once/node))

- SLS total RO / sec = $lsu\,\lambda_{lc}N^2$ bps = $\Theta(\lambda_{lc}N^2)$

$\Rightarrow$ $\rho_{\lambda_{lc}}^{SLS} = 1$, $\rho_{\lambda_t}^{SLS} = 0$, $\rho_N^{SLS} = 2$.

Dynamic Source Routing (DSR)

- no proactive information is exchanged

- A node (source) reaches a destination by flooding the network with a route request (RREQ) message.
- When an RREQ message reaches the destination (or a node with a cached route towards the destination) a route reply message is sent back to the source, including the newly found route
- The source attaches the new route to the header of all subsequent packets to that destination, and any intermediate node along the route uses this attached information to determine the next hop in the route.

- focus on DSR without the route cache option (DSR-noRC)

DSR without Route Cache (DSR-noRC)

DSR-noRC reactive RO

RREQ (route request) messages are generated by:
- new session requests (at a rate λ_s per second per node)
- failures in links that are part of a path currently in use

only new requests are considered $\Rightarrow$ a *lower bound* is obtained

- $\lambda_s N$ RREQ messages are generated / sec (new session requests)
- overhead of $size\_of\_RREQ(N-1)$ bits / RREQ message
  (each RREQ message is flooded $\Rightarrow$ $N-1$ retransmissions)

$\Rightarrow$ DSR-noRC reactive RO / sec $= \Omega(\lambda_s N^2)$

- *upper bound* $O((\lambda_s + \lambda_{lc})N^2)$

(assumption: each link failure has the same effect as a new session request (local repair fails))

DSR without Route Cache (DSR-noRC)

DSR-noRC suboptimal RO

- only the extra bandwidth required for appending the source-route in each data packet is considered $\Rightarrow$ a lower bound is obtained
- number of bits appended in each packet ~ length L_i of path i

$L_i \geq L_i^{opt}$ (optimal path length), L_i^{opt} instead of L_i $\Rightarrow$ lower bound

- ➢ extra bits for a packet delivered using a path i : $(\log_2 N)(L_i^{opt})^2$
  - ○ at least L_i^{opt} retransmissions of at least $L_i^{opt} \log_2 N$ bits
  - ○ $\log_2 N$ is the minimum length of a node address

DSR without Route Cache (DSR-noRC)

- ➢ extra bits for a packet delivered using a path i: $(\log_2 N)(L_i^{opt})^2$
- ➢ extra bits / packet over all paths (average):
  $E\{(\log_2 N)(L_i^{opt})^2)\} \geq (\log_2 N)E\{L_i^{opt}\}^2 = (\log_2 N)L^2$ bits
- ➢ $\lambda_t N$ packets are transmitted / sec

$\Rightarrow$ suboptimal RO at least $\lambda_t N(\log_2 N)L^2$ bps, $L = \Theta(\sqrt{N})$ (a.4)

$\Rightarrow$ DSR-noRC suboptimal RO / sec = $\Omega(\lambda_t N^2 \log_2 N)$ bps

DSR without Route Cache (DSR-noRC)

DSR-noRC reactive RO / sec $= \Omega(\lambda_s N^2)$

$= O((\lambda_s + \lambda_{lc})N^2)$

DSR-noRC suboptimal RO / sec = $\Omega(\lambda_t N^2 \log_2 N)$ bps

- DSR-noRC total overhead / sec = $\Omega(\lambda_s N^2 + \lambda_t N^2 \log_2 N)$
- $\rho_{\lambda_t}^{DSR-noRC} = 1,\ 0 < \rho_{\lambda_{lc}}^{DSR-noRC} <= 1,\ \ \rho_N^{DSR-noRC} > 2$

Zone Routing Protocol (ZRP)

- lower bound for ZRP' *total overhead* (can be proved)

$$ZRP_{ov} = \Omega(n_k\ \lambda_{lc} N + \lambda_s N^2 / \sqrt{n_k})$$

where: n_k = average number of nodes inside node's 'zone'

- proactive overhead: $\Omega(n_k\ \lambda_{lc} N)$
- reactive overhead: $\Omega(\lambda_s N^2 / \sqrt{n_k})$

(For the sub-optimal routing overhead an upper bound can be proved that shows that it is (asymptotically) dominated by the reactive overhead)

- Minimizing this lower bound by properly choosing the value n_k

$$n_k = \Theta((\tfrac{\lambda_s N}{\lambda_{lc}})^{\frac{2}{3}}) \Rightarrow \Omega(\lambda_{lc}^{\frac{1}{3}} \lambda_s^{\frac{2}{3}} N^{\frac{5}{3}})$$

Can hierarchies enhance scalability?

Hierarchical Routing Protocols

Classical approach to improving routing scalability

Outline

1. Hierarchical Routing Techniques
2. Cluster and Cluster Leader Selection Methods
   - Cluster Radius
   - Cluster Affiliation Method
   - Performance Objective
   - Cluster Leader Selection
3. Topology Abstraction Methods
   - Virtual Node Abstraction
   - Virtual Link Abstraction
4. Location Management Methods
   - LM1, LM2, LM3

Hierarchical Routing Techniques

Characteristics of Hierarchical Routing Techniques

1. Basic idea: organize nodes in groups and assign to nodes different functionalities inside and outside the group [1]
2. Recursive grouping of nodes: nodes are grouped into *clusters (level-1),* clusters into *superclusters (level-2),* and so on…
3. A *cluster-leader* (among the cluster nodes) is elected to coordinate the cluster

Nodes need to have or exchange only partial knowledge of the network

Hierarchical Routing Techniques

Characteristics of Hierarchical Routing Techniques

4. A formal abstraction method is used to form the virtual network topology
5. Routing information about far away nodes is aggregated
   - Bandwidth requirements are lower
   - Memory requirements for storage are lower
   - Processing requirements at the nodes are lower
6. Mobility is constantly changing the physical network topology. Thus, a location management scheme is essential to handle mobility

Hierarchical Routing Techniques

Components of Hierarchical Routing Schemes

- Cluster and Cluster Leader Selection Algorithm
- Topology Abstraction Method
- Location Management Technique

Cluster and Cluster Leader Selection Methods

Cluster Radius Criterion

1. **1-hop clusters**
   - All cluster nodes are 1-hop neighbors of the cluster leader (simple and reliable)
   - Adjacent clusters have cluster leaders **at most** two hops away (via gateway nodes)
2. **More than 1-hop clusters**
   - Cluster radius is below a maximum
   - Cluster radius can be adjusted based on conditions and performance objectives (flexibility)

Cluster and Cluster Leader Selection Methods

Cluster Affiliation Method Criterion

(refers to the way nodes are assigned to clusters)

1. **Cluster nodes decide**
   - Allows for distributed algorithm implementation
     - Lack of centralized control
     - Latency in propagating control information
     - unpredictable dynamics
2. **Cluster leader decides**
   - ‘Grabs’ nodes sequentially: slow convergence
   - ‘Grabs’ nodes in large sets: faster, but requires information from outside the cluster

Cluster and Cluster Leader Selection Methods

Performance Objective Criterion

1. **Clusters with balanced size**
   - Clusters with a minimum or maximum number of nodes
   - E.g. MMWN[6] protocol uses 'join' or 'split' procedures if a cluster population exceeds certain limits
   - Clusters with a fixed number of nodes (almost) [8]

2. **Cluster nodes with k-connectivity degree**
   - Increased robustness against link failures

3. **Minimum level of affiliation**
   - Affiliation: Composite bandwidth between a node and all other nodes in the cluster, or
   - Distance to the cluster leader, or
   - Linear combinations of the above
   - Commonalities in interests of locality [9]

Cluster and Cluster Leader Selection Methods

Cluster Leader Selection Criterion

1. **In homogeneous networks**
   - Nodes with the **lowest ids**
   - Equivalent to *randomized* cluster leader election

2. **In non-homogeneous networks**
   - Election based on extra knowledge about the scenario, e.g. known **mobility patterns** in LANMAR[4]
   - Other extra knowledge about power, memory, etc.

3. **Maximization of some gain function**
   - Gain function example: node degree
   - Other gain functions exist, definitions vary accordingly, e.g. SOAP[2, 7] protocol

Topology Abstraction Methods

- **Objective**:

  To **reduce the topology information** that needs to be propagated inside the network

- **Some existing techniques**:
  - Virtual node abstraction, in HierLS[7]
  - Virtual gateway abstraction, in MMWN[6]

Virtual Node Abstraction

Algorithm

1. Network nodes are level-1 nodes
2. Level-i nodes are grouped into level-i clusters
3. Level-i clusters become level-(i+1) nodes

Example

- i=1: level-1 clusters $\mapsto$ level-2 nodes
- i=2: level-2 clusters $\mapsto$ level-3 nodes
- …
- i=m-1: level-(m-1) clusters $\mapsto$ level-m nodes

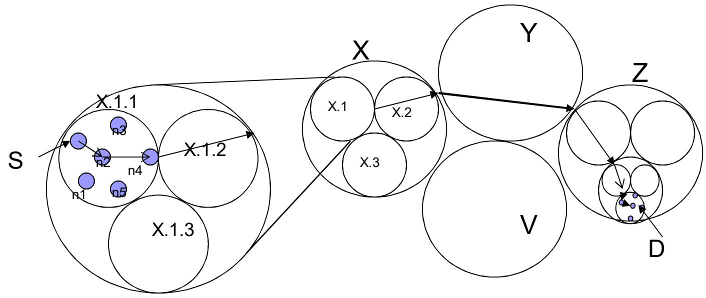
[$\mapsto$ stands for *become*]

Link state information inside a level i cluster is aggregated and transmitted only to other level i nodes belonging in the same level i cluster. **Thus a node link change may not be sent outside the level 1 cluster thus reducing the proactive overhead**.

Virtual Node Abstraction

Example: Routing with the Virtual Node Abstraction

- Routing relies on Location Management service (providing to source S the highest level cluster containing D and not S)
- Node S will only build a high-level route towards D:
  [S – n2 – n4 – X.1.2 – X.2 – Y – Z – D] in the 4-level net hierarchy below

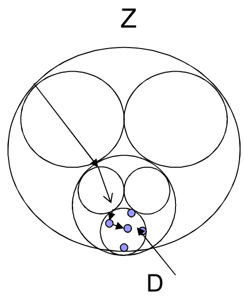


Virtual Node Abstraction

Example: Routing with the Virtual Node Abstraction

- Intermediate nodes will produce the same high-level route
- Intermediate nodes will expand the high-level links that traverse their cluster using lower level information (more detailed links)

This expansion can be seen graphically in the last segment of the route, Z – D



Virtual Node Abstraction

Conclusions on the Virtual node abstraction method

- Intuitive method
- Easy to analyze, implement, debug
- Clusters aren't able to properly estimate the virtual link cost

Note on the Virtual gateway abstraction method

- Provides better aggregation of link information
- Produces routes of better quality
- Extra complexity in maintaining the virtual gateway structure

Location Management (LM) Methods

- **Core job of hierarchical routing**: to efficiently aggregate routing information by using the network hierarchy
- Source nodes need to know the identity of a cluster associated with the destination node
  - LM provides this information
  - LM is needed **only** in mobile networks, not wireline, static networks

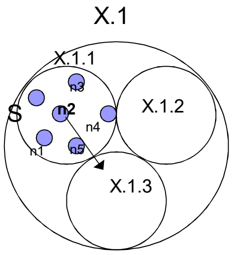
Location Management Methods

- **LM implementation methodologies**
  - Proactive (using Location Update Messages)
  - Reactive (using Paging Techniques)
  - a combination of both

- **Some basic/typical LM techniques**
  1. LM1 (pure proactive)
  2. LM2 (local paging)
  3. LM3 (global paging)

Location Management Method LM1

- Proactive method using Location Update Messages (LUMs)



Algorithm

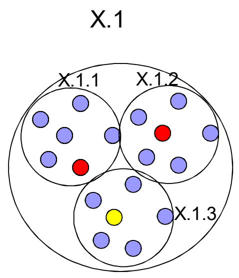
When a node changes its level-i clustering membership, but not its level-(i+1) membership, it sends an LUM to all the nodes inside its level-(i+1) cluster.

Example

- Node n2 moves from cluster X.1.1 to X.1.3
- Node n2 sends updates to all nodes in cluster X.1

Location Management Method LM2

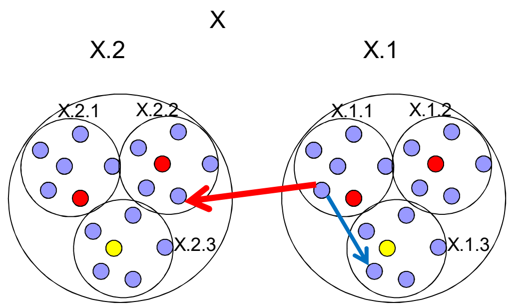
- Each level-i cluster elects one level-i LM server among its' participating nodes, up to the level-m cluster
- LM servers form a **hierarchical tree** of servers
- Location Update Messages are propagated only between LM servers
- Nodes needing location information about other nodes in the network, page their local level-1 LM server **(local paging)**



Level-1 LM server

Level-1 LM server AND level-2 LM server

Location Management Method LM2



Level-1 LM server

Level-1 LM server AND level-2 LM server

BLUE move: X.1.3 receiving, informs X.1
X.1 informs X.1.1, X.1.3
RED move: X.2.2 receiving, informs X.2
X.2 informs X.2.1, X.2.3
X.2 informs X
X informs all X.x except X.2 (here, X.1)
X.1 informs all X.1.1, X.1.2, X.1.3

Location Management Method LM2

LM2 Algorithm

- The level-i LM server in the new cluster of a moving node will send an LUM to its level-(i+1) LM server
- The level-(i+1) server will check if the moving node is new in the level-(i+1) cluster and inform the level-(i+2) LM server accordingly
- The level-(i+1) server will forward the LUM to all level-i LM servers inside this level (i+1) cluster
- Each such level-i LM server will forward LUMs to all of its level-(i-1) LM servers
- The previous step repeats all the way until all such level-1 LU servers are updated

Location Management Method LM3

- Same as LM2, with LM servers exchanging LMUs, <u>with one exception</u>:

  Level-i LM servers receiving LUMs from higher level servers **do not** forward this information to the lower level servers

- A mechanism for removing outdated location information about nodes that left a level-i cluster is needed
- Nodes needing location information about other nodes in the network page their local level-1 LM server
- If it is not updated, then the level-2 LM server is paged, then the level-3 server, and so on **(global paging)**

Comparison of LM1, LM2, and LM3

- LM1 is the **simplest** of the three
  - Consumes **significant bandwidth** for propagating the LUMs
- LM2 reduces the bandwidth consumption in the network
  - Induces **increased complexity** and **latency** in route building
- LM1 and LM2 techniques do not alter the **asymptotic characteristics** of the hierarchical routing protocol used

Comparison of LM1, LM2, and LM3

- LM3 is **the most complex** to implement and analyze
  - It can **significantly reduce** the amount of overhead induced by mobility
  - **Latency is increased**
  - **High paging cost** under high traffic load
  - More susceptible to **single points of failure**

Hierarchical Link State (HierLS)

The network organized in m level clusters, each of equal size k ($N = k^m$), k is predefined while m increases with N.

Location Management (LM) service choices:

- **LM1** Pure proactive.
- **LM2** Local paging.
- **LM3** Global paging.

HierLS-LM1 Total Overhead

- HierLS-LM1 *proactive* RO = $\Omega(s\,N^{1.5} + \lambda_{lc} N)$
- HierLS-LM1 *suboptimal RO* / sec = $\Theta(\lambda_t N^{1.5+\delta})$

$\Rightarrow$ HierLS-LM1 *total overhead* / sec

$$= \Omega(s * N^{1.5} + \lambda_{lc} * N + \lambda_t N^{1.5+\delta})$$

$\Rightarrow \rho_{\lambda_t}^{HierLS-LM1} = 1,\ \rho_{\lambda_{lc}}^{HierLS-LM1} = 1$, and $\rho_N^{HierLS-LM1} = 1.5 + \delta > 1.5$

(HierLS is *almost* scalable w.r.t. network size)

HierLS-LM2 Total Overhead

- HierLS-LM2 *suboptimal RO* / sec =
  = HierLS-LM1 *suboptimal RO* / sec = $\Theta(\lambda_t N^{1.5+\delta})$
- HierLS-LM2 *proactive* RO =
  = HierLS-LM1 *proactive* RO = $\Omega(s\,N^{1.5} + \lambda_{lc} N)$
- HierLS-LM2 *reactive* RO = $O(\lambda_t N)$

$\Rightarrow$ HierLS-LM2 *total overhead* = $\Omega(s\,N^{1.5} + \lambda_{lc} N + \lambda_t N^{1.5+\delta})$
same asymptotic scalability factors as HierLS-LM1

HierLS-LM3 Total Overhead

- HierLS-LM3 *suboptimal RO* / sec =
  = HierLS-LM1 *suboptimal RO* / sec = $\Theta(\lambda_t N^{1.5+\delta})$
- HierLS-LM3 *proactive* RO $=\Theta(s\,N \log N) + \lambda_{lc} N$
- HierLS-LM3 *reactive* RO = $O(\lambda_t N^{1.5+\delta})$

$\Rightarrow$ HierLS-LM3 *total overhead* = $\Omega(s\,N \log N + \lambda_{lc} N + \lambda_t N^{1.5+\delta})$
same asymptotic scalability factors as HierLS-LM1

References

Some references are at: http://cgi.di.uoa.gr/~istavrak/publications.html

1. Xiaoyan Hong, Kaixin Xu, and Mario Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks", IEEE Network Magazine, July/August 2002
2. C. Santivanez, A.B. McDonald, I. Stavrakakis, S. Ramanathan, "On the Scalability of Ad Hoc Routing Protocols", in Proceedings of IEEE Infocom '2002, New York, USA, June 2002.
3. B.A. Iwata, C.C. Chiang, G. Pei, M. Gerla, and T.W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Network", IEEE JSAC on Communications, vol. 17, no. 8, pp. 1369, Aug. 1999
4. G. Pei, M. Gerla and X. Hong, "LANMAR: Landmark Routing for Large Scale Wireless Networks with Group Mobility", in Proceed. Of ACM Workshop on Mobile and Ad Hoc Networking and Computing, MobiHoc' 00, Boston, MA, Aug. 2000
5. A.B. McDonald and T.F. Znati. "A Mobility Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks", IEEE JSAC on Communications, col. 17,no. 8, pp. 1466, Aug. 1999
6. R. Ramanathan and M. Steenstrup, "Hierarchically-organized, multihop mobile wireless network for quality-of-service suppor", BBN Technologies
7. C. Santivanez, S. Ramanathan and I. Stavrakakis, "Making Link State Routing Scale for Ad Hoc Networks", in Proceed. Of MobiHOC' 2001, Long Beach, CA, Oct. 2001

Can we have scalable flat routing protocols?

Flat Routing

- ``flat" ≠ hierarchical
- each node and link in the topology table is an actual node or link
  (no topology abstraction, no ``boundaries", no hierarchical addressing)
- the topology table may grow large as the network size increases
- usually requires much more memory and processing power
  (however, a key challenge is the excessive bandwidth)
- requires careful design
  (else may result in much more bandwidth consumption than hierarchical)

Techniques for bandwidth consumption reduction:
(in isolation or in combination)
- efficient/controlled flooding
- limited LSU generation rate
- limited LSU dissemination depth

Efficient Flooding

- classical flooding is very inefficient
  (each node receives the same packet several times)
- Efficient flooding:
  - reduce the number of times a message is retransmitted
  - each recipient should receive each message at least once
- Example (mechanism):
  - find a tree that covers all the nodes
  - propagate the message across all the nodes in the tree

  (every node in the tree transmits the message only once)

Examples (protocols):
- Optimized Link State Routing (OLSR)
- Topology Broadcast based on Reverse Path Flooding (TBRPF)
- Core Extraction Distributed Ad-Hoc Routing (CEDAR)

Limited Generation (Rate)

➢ limit the amount of control information through update aggregation and reduced generation frequency

Examples

➢ update generation at times which are multiples of a base period t_e (effective for high mobility)

- Global State Routing (GSR)
- Discretized Link State (DLS)

➢ updates only for changes that affect another node's best route

- Source-Tree Adaptive Routing (STAR)

➢ operating on connected subgraphs

- OLSR

Limited Dissemination (Depth)

➢ reduced depth of propagation of routing updates

- most updates sent to a subset of nodes (not the entire network)
- the subset may change over time

➢ promise for scalability improvement

- especially for networks with a large diameter
- challenge: not to overly compromise route optimally

Examples

➢ ZRP, NSLS limit the update propagation to k-neighbors only

➢ Fisheye State Routing

- Network = “in-scope” + “out-of-scope” subsets
- out-of-scope nodes are “informed” with a smaller frequency

➢ family of Fuzzy Sighted Link State (FSLS) algorithms

- LSU generation: at multiples of a base time t_e
- a LSU (in general) travels TTL hops
- TTL depends on the current time

FSLS Algorithms

Key Observation:
Nodes that are far away do not need to have complete topological information in order to make a good next hop decision

➔ thus propagating every link status change over the entire network may not be necessary.
(in hop-by-hop routing, changes experienced by nodes far away tend to have little impact in a node's 'local' next hop decision)

- pure proactive protocols (as SLS) do not scale well with size
  (induced overhead increases as rapidly as N^2)
- a reduction of the proactive overhead may be achieved:
  - in space (by limiting which nodes the link state update is transmitted to)
  - in time (by limiting the time between successive link status updates)
- balance is necessary
  (a decrease in proactive RO will induce an increase in suboptimal RO)

FSLS Algorithms

Family of Fuzzy Sighted Link State (FSLS)

Key Design Idea: the *frequency* of Link State Updates (LSUs) propagated to *distant* nodes is reduced

- a node transmits a Link State Update (LSU)
  - only at particular time instants
    (potentially several link changes are 'collected')

    the *Time To Live* (TTL) field of the LSU
    (specifies how far the LSU is propagated)
    is set to a value that is a function of the current (LSU generation) time

FSLS Algorithms

Under a FSLS protocol

(& after one global LSU transmission (TTL $=\infty$), for example, during initialization)

a node:

- wakes up every $2^{i-1} * t_e$ $(i = 1, 2, 3, \dots)$ seconds
- and transmits a LSU with TTL set to s_i

  (if there has been a link status change in the last $2^{i-1} * t_e$ seconds)

i.e.:

- 'wakes up' every t_e seconds and sends a LSU with TTL set to s_1

  (if there has been a link status change in the last t_e seconds)
- wakes up every $2 * t_e$ seconds and transmits a LSU with TTL set to s_2

  (if there has been a link status change in the last $2 * t_e$ seconds)
- …

FSLS Algorithms

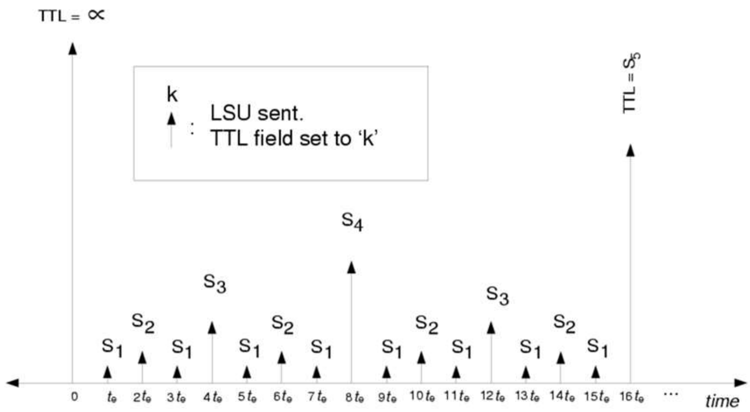
- Strictly speaking, the node will consider link changes since the last time a LSU with TTL greater or equal to s_i was considered (not necessarily transmitted).

- If the value of s_i is greater than the distance from this node to any other node in the network, the TTL field of the LSU is set to infinity (global LSU), and all the counters and timers are reset.

- As a soft state protection on low mobility environments, a periodic timer may be set to ensure that a global LSU is transmitted at least each t_b seconds.

FSLS Algorithms

Example of FSLS's LSU generation process
(mobility is high $\Rightarrow$ LSUs are always generated every t_e seconds)



- the sequence $s_1, s_2, \ldots$ is non-decreasing

FSLS Algorithms

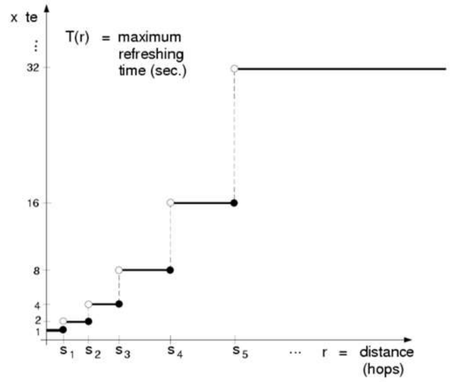
For example, time $4t_e$ is a multiple of t_e (associated with s_1), $2t_e$ (associated with s_2) and $4t_e$ (associated with s_3)

- if there has been a link status change in the past t_e or $2t_e$ seconds, then there has been a link change in the past $4t_e$ seconds
  $\Rightarrow$ if we have to set the TTL field to at least s_1 (or s_2) we also have to increase it to s_3
- if there has not been a link status change in the past $4t_e$ seconds, then there has not been a link change in the past t_e or $2t_e$ seconds
  $\Rightarrow$ if we do not send a LSU with TTL = s_3, we do not send a LSU at all

At times $4*k*t_e$ (k odd) the link state change activity during the past $4t_e$ seconds is checked and, if there is any, then an LSU with TTL set to s_3 is sent. (In the highly mobile scenario assumed on the figure, a LSU with TTL equal to s_3 is sent at times $4t_e$ and $12t_e$.)

FSLS Algorithms

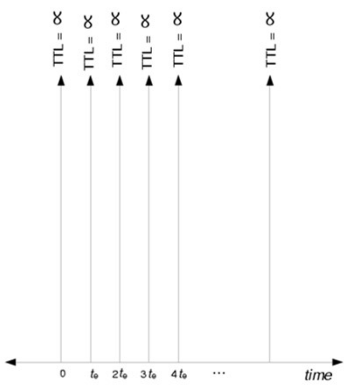
- nodes that are s_i hops away from a tagged node will learn about a link status change at most after $2^{i-1}t_e$ seconds ('refresh' time)
- $T(r)$ determines the latency in the link state information
- Different approaches may be implemented by considering different $\{s_i\}$ sequences.
  - Discretized Link State (DLS)
  - Near Sighted Link State (NSLS)



Maximum refresh time as a function of distance from link event.

FSLS Algorithms: Discretized Link State (DLS)

- DLS is obtained by setting $s_i = \infty$ for all i
- is similar to the Standard Link State (SLS)



difference: under DLS a LSU is not sent immediately after a link status change (only when the current t_e interval is completed)

⇒ several link status changes may be collected in one LSU (modification of SLS that attempts to scale better w.r.t. mobility)

FSLS Algorithms: Discretized Link State (DLS)

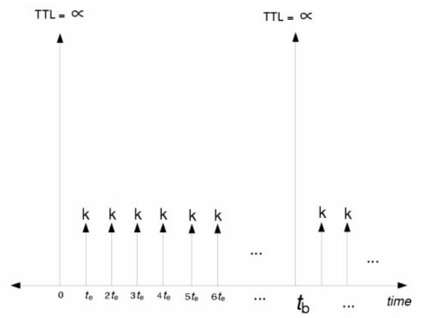
- Under high mobility
  some similarities with Global State Routing (GSR)

- In GSR, a node exchanges its version of the network topology table with its one-hop neighbors each t_{flood} seconds.
(limits the frequency of link state updates to be no greater that $\frac{1}{t_{flood}}$)

- Under high mobility (LSUs are sent every t_e seconds) DLS induces the same proactive overhead as GSR (by setting $t_e = t_{flood}$)

FSLS Algorithms: Near Sighted Link State (NSLS)

- NSLS is obtained by setting
  - $s_i = k$ for $i < p$
  - $s_p = \infty$ (p integer)



- a node receives information only from nodes inside its sight area
  (that are less than 'k' hops away)

FSLS Algorithms: NSLS

Typical problem due to node mobility:

- Suppose that initially a node knows routes to every destination.
- as time evolves nodes move, links go down
  - the node learns that the previous routes will fail
  - the node **does not** learn of new available routes (**out-of-sight information is not being updated**)
- common to every algorithm in the FSLS family (NSLS represents its worst case scenario)

"Solution" using past knowledge

- the node uses the 'memory' of past links
  - sends packets in the direction it 'saw' the destination for the last time
  - **if the packet gets to a node that is on the 'sight' of the destination, this node can forward the packet to the destination**.

FSLS Algorithms: NSLS

NSLS has similarities with:

- the proactive part of the Zone Routing Protocol (ZRP) (without the reactive route search)
- the Distance Routing Effect Algorithm for Mobility (DREAM) difference:
  - NSLS limits the LSU propagation based on the number of hops traversed
  - DREAM limits the position update message's propagation based on the geographical distance to the source

FSLS Algorithms: NSLS

NSLS has similarities with Fisheye State Routing (FSR) :

FSR uses the same topology dissemination mechanism as GSR, but it does not transmit the whole topology information each t_{flood} seconds. (only a short version including only the closest nodes entries is transmitted)

A second, larger timer (t_{large}) is used for out-of-scope nodes

Setting $t_e = t_{flood}$ and $t_b = t_{large}$,
& k : all the nodes in-scope are k or less hops away

$\Rightarrow$ NSLS induces the same control overhead as FSR

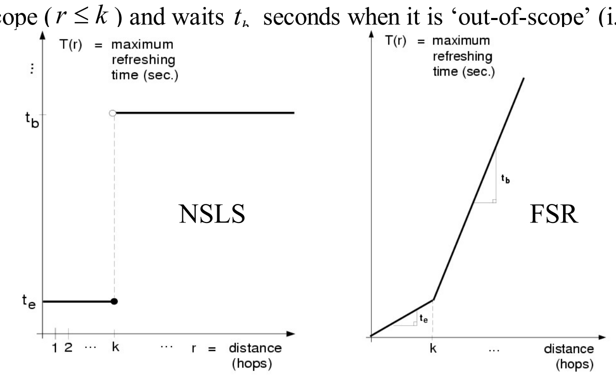
FSLS Algorithms: NSLS

➢ the latency in updating link state information is greater in FSR

NSLS: $T(r) = t_e$ for $r \le k$, and $T(r) = t_b$ for $r > k$

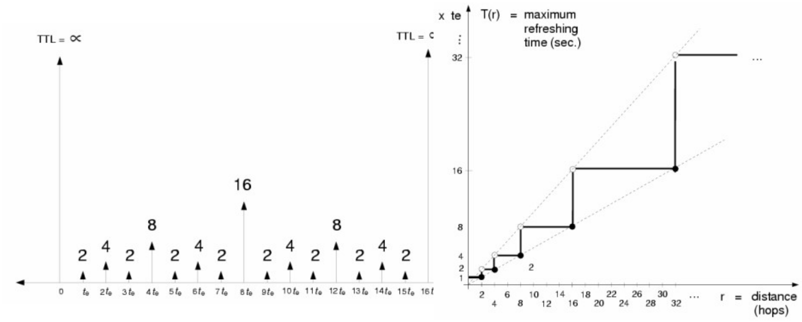
FSR: $T(r) = t_e * r$ for $r \le k$, and $T(r) = k * t_e + (r-k) * t_b$

(In FSR, a LSU waits at most t_e to be propagated one more hop away, if it is in scope ($r \le k$) and waits t_b seconds when it is 'out-of-scope' (i.e. $r > k$)).



FSLS Algorithms: Hazy Sighted Link State (HSLS)

- HSLS is obtained by setting $s_i = 2^i$ for all i
- under a.1-a.8 minimum total overhead



- almost linear relationship between update latency and distance

FSLS Algorithms: Hazy Sighted Link State (HSLS)

latency versus distance curve: optimal performance when linear
(optimal balance between proactive and sub-optimal routing overhead)

- **it turns out that angular uncertainty is roughly constant (independent of the distance)**
  - hop-by-hop routing is based on the next hop decision
  - Prob. of wrong decision depends mainly in the angular uncertainty
  - If constant ⇒ Prob. of bad next hop decision is constant

Intuitively:

- If faster than linear, too many mistakes when forwarding packets to nodes far away
- If slower than linear, fewer mistakes, but the proactive overhead increases

Asymptotic Results

Protocol	Proactive	Reactive	Suboptimal
PF	–	–	$\Theta(\lambda_t N^2)$
SLS	$\Theta(\lambda_{lc} N^2)$	–	–
DSR-noRC	–	$\Omega(\lambda_s N^2)$	$\Omega(\lambda_t N^2 \log_2 N)$
		$O((\lambda_s + \lambda_{lc}) N^2)$	
HierLS	$\Omega(sN^{1.5} + \lambda_{lc} N)$	–	$\Theta(\lambda_t N^{1.5+\delta})$
ZRP	$\Theta(n_k \lambda_{lc} N)$	$\Omega(\lambda_s N^2 / \sqrt{n_k})$	$O(\lambda_t N^2 / \sqrt{n_k})$
HSLS	$\Theta(N^{1.5} / t_e)$	–	$\Theta((e^{\lambda_{lc} t_e K_4} - 1)\lambda_t N^{1.5})$

Asymptotic expressions

Best possible total overhead bounds for mobile ad hoc networks protocols

Protocol	Total overhead (best)	Cases
PF	$\Theta(\lambda_t N^2)$	Always
SLS	$\Theta(\lambda_{lc} N^2)$	Always
DSR-noRC	$\Omega(\lambda_s N^2 + \lambda_t N^2 \log_2 N)$	Always
HierLS	$\Omega(sN^{1.5} + \lambda_{lc} N + \lambda_t N^{1.5+\delta})$	LM1
ZRP	$\Omega(\lambda_{lc} N^2)$	if $\lambda_{lc} = O(\lambda_s / \sqrt{N})$
	$\Omega(\lambda_{lc}^{\frac{1}{3}} \lambda_s^{\frac{2}{3}} N^{\frac{5}{3}})$	if $\lambda_{lc} = \Omega(\lambda_s / \sqrt{N})$ and $\lambda_{lc} = O(\lambda_s N)$
	$\Omega(\lambda_s N^2)$	if $\lambda_{lc} = \Omega(\lambda_s N)$
HSLS	$\Theta(\sqrt{\lambda_{lc} \lambda_t} N^{1.5})$	if $\lambda_{lc} = O(\lambda_t)$
	$\Theta(\lambda_{lc} N^{1.5})$	if $\lambda_{lc} = \Omega(\lambda_t)$

Observing the asymptotic expressions

w.r.t network size

- HierLS and HSLS scale better
- flooding the entire network (link state, route request, or data)

$\Rightarrow$ routing protocol scalability factor w. r.t. network size = 2

- splitting the information dissemination at two different levels (like in 2-level hierarchical routing, NSLS, ZRP, and DREAM)

$\Rightarrow$ routing protocol scalability factor w.r.t. network size = 1.66

- allowing the number of levels grow as the network size increases (as done explicitly by m-level HierLS and implicitly by HSLS)

$\Rightarrow$ routing protocol scalability factor w.r.t. network size = 1.5
(seems to be the limit for routing protocols for networks defined by a.1-a.8)

Observing the asymptotic expressions

w.r.t traffic intensity

- SLS, and ZRP scale better (*total overhead* is independent of λ_t)
- HSLS follows (scales as $\Theta(\sqrt{\lambda_t})$)
- PF, DSR, and HierLS are the last (*total RO* ~ traffic)
- ZRP adapts its zone size ($\rightarrow$ pure proactive)
- HSLS increases its LSU generation rate (reducing update latency and improving the quality of the routes)

Conclusion: as traffic load increases

1. the quality of the routes becomes more and more important
2. more bandwidth should be allocated for the routing process (to improve quality), contradicting the widely held belief that as traffic load is increased, less bandwidth should be allocated to control traffic and let more bandwidth available for user data

Observing the asymptotic expressions

w.r.t. the rate of topological change

- PF *total RO* is independent of the rate of topological change
  if the rate of topological change increases too rapidly may be preferred
  (especially, if size and traffic are small)
- ZRP and DSR are next
  their *lower bounds* are independent of the rate of topological changes
  (their behavior should depend somewhat on the rate of topological change)
- SLS, HierLS, and HSLS
  their *total overhead* increases linearly w.r.t. the rate of topological change

Observing the asymptotic expressions

It is interesting to note that:

- if mobility OR traffic increase
  ZRP achieves almost the best performance
- if mobility AND traffic increase at the same rate
  ($\lambda_{lc} = \Theta(\lambda)$ and $\lambda_t = \Theta(\lambda)$ (for some parameter λ))
  ZRP's scalability w.r.t λ same as HSLS's and HierLS's
  ZRP: $\Omega(\lambda N^{1.66})$, HSLS: $\Theta(\lambda N^{1.5})$, HierLS's: $\Theta(\lambda N^{1.5+\delta})$
- HSLS scales better with traffic intensities than HierLS
  (the only other protocol that scales well with size)

intuitive explanation:
- HierLS never attempts to find optimal routes
  (even under slowly changing conditions)
- HSLS may obtain full topology information =>optimal routes
  (if the rate of topological changes is small w.r.t. $1/t_e$)

Comparing HierLS and HSLS

- both scale well w.r.t. network size
  (both induce a multi-level information dissemination technique)

- HSLS's routes' quality does not degrade with network size
  (angular displacement uncertainty depends mainly on the nodes speed and t_e)

- HierLS's routes' quality suffers small degradation each time the number of hierarchical levels is increased

- HSLS is able to improve the quality of its routes as a response to an increase in traffic load

- HierLS's route quality dependents on the number of hierarchical levels
  (which depend on the cluster size (independent of the traffic load))

$\Rightarrow$ HierLS can not react to an increase in traffic load

Comparing HierLS and HSLS

- theoretical analysis focuses on asymptotically large networks, heavy traffic load, and saturation conditions

- What about the constants involved in the asymptotic expressions?
  (and the effect of other factors (MAC, latency on detecting failures, etc.))

$\Rightarrow$ Simulations: medium size network with moderate loads

- Simulator: OPNET
- Topology: 400 randomly located nodes on a square (320square miles)
- Mobility: Each node chooses a random direction (among 4) and moves at 28.8 mph (at the area boundaries bounces back)
- Traffic: 60 8kbps streams
- radio link capacity = 1.676 Mbps

Comparing HierLS and HSLS

- MAC protocols: unreliable and reliable CSMA
  (reliable CSMA: up to 10 retransmissions if an ACK was not received)
- Simulations time = 350 seconds (initialization: first 50 seconds)
- Performance metric: throughput
  (fraction of packets successfully delivered)
- HierLS-LM1 approach:
  - since the network size is relatively small, only 2 levels were formed during the simulations
  - node affiliation decisions performed by the cluster leaders with the goal of balancing cluster sizes (9 ≤ cluster size ≤ 35)
  - cluster leader selection: the node in the cluster with the largest number of (unassigned) k-hop neighbors

Simulation results

Protocol	UNRELIABLE	RELIABLE
HSLS	0.2454	0.7991
HierLS-LM1	0.0668	0.3445

- in both cases HSLS outperforms HierLS
- unreliable MAC biases performance towards HSLS due to:
  (relative difference is reduced under the reliable MAC)

1. unreliable CSMA=>high rate of collisions=>shorter paths are favored
   - For short paths:
     - HSLS routes are almost optimal
     - HierLS routes may be far from optimal
       (if the destination belongs to a neighboring cluster)
2. latency to detect link up/downs
   - HierLS: information is synchronized among all the nodes in the cluster => some latency is enforced to avoid link flapping
   - HSLS: reacts much faster to link degradation
     (each node may have its own view of the network / may be more aggressive in temporarily taking links down without informing other nodes)

Summary

- ➢ hierarchical routing approaches
- o high implementation complexity
- o hard to analyze
- o overhead for maintaining the hierarchy / location management reduces the savings due to reduction of update dissemination

➢ hierarchical routing does not provide any fundamental advantage over efficient limited dissemination techniques (HSLS)

- ➢ in terms of scalability
  - ▪ HSLS scales no worse w.r.t network size
  - ▪ HSLS scales better w.r.t. traffic rate
- ➢ in terms of performance

Conclusions

Which protocol should be preferred in practice?

it depends on several factors

+ limited dissemination techniques (as HSLS)
- ➢ network size, mobility, and traffic increases
- ➢ implementation complexity is a major concern

+ hierarchical routing
- ➢ storage capacity at each node is limited
- ➢ the topology is sparse
- ➢ many hostile misbehaving nodes
- ➢ non homogeneous networks with underlying structure

  examples: correlated mobility with well defined group patterns

  low power terrestrial nodes and high power/aerial nodes

focus on scalability from a bandwidth point of view
- ➢ Other challenges: routing latency, QoS support, etc

Conclusions

Common misconceptions:

1.As traffic load increases, the bandwidth allocated to routing information dissemination should decrease.
2.As network size increases the best option is to engage a hierarchical routing algorithm.

- flat-routing scalability-improving techniques are good candidates for achieving scalable routing protocols
- imposing an arbitrary hierarchy in homogeneous ad hoc networks provides no scalability advantage
- hierarchical routing would justify its complexity only if the hierarchy built was a response/reflection of an underlying hierarchy/structure in the network