

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.

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 usage than required.

As the network size increases keeping route optimality imposes an unacceptable cost.

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

- revise the concept of *overhead* so that it includes 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)
- motivates the definition of the *total overhead* of a routing protocol

Minimum traffic load – Total overhead

$$\begin{array}{ccccc} \textit{Total amount} & = & \textit{minimum traffic load} & + & \textit{total overhead} \\ \textit{of bandwidth} & & & & \\ \text{routing protocol} & & \text{routing protocol} & & \text{routing protocol} \\ \text{dependent} & & \text{independent} & & \text{dependent} \end{array}$$

The *total overhead* induced by a routing protocol is the difference between the total amount of bandwidth actually consumed by the network running such routing protocol minus 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

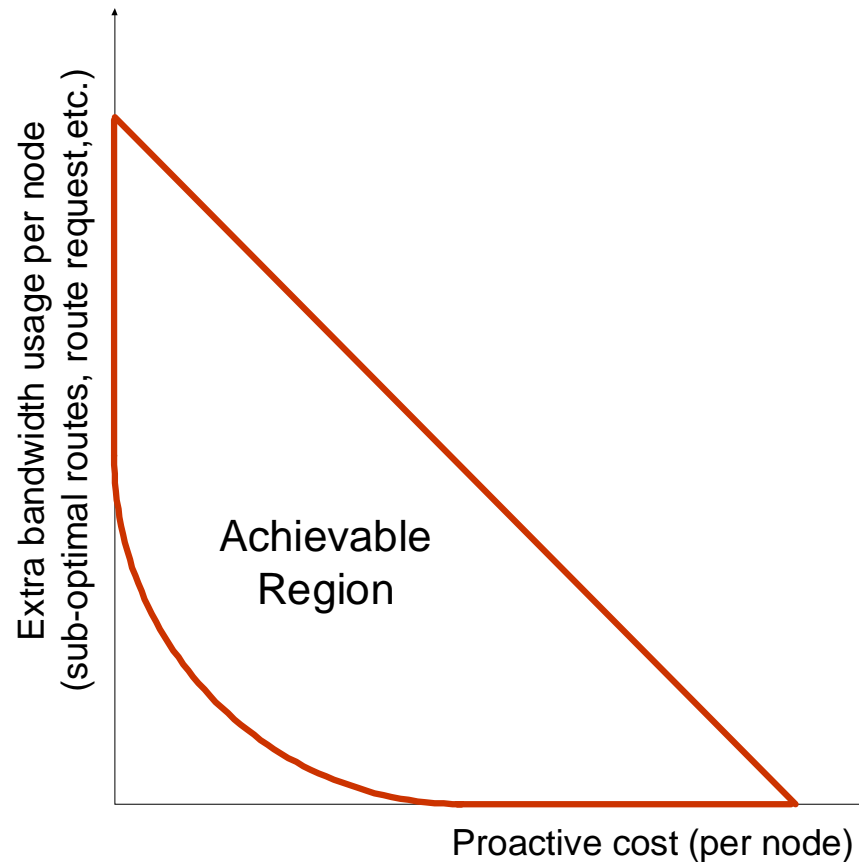
achievable region of overhead:

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



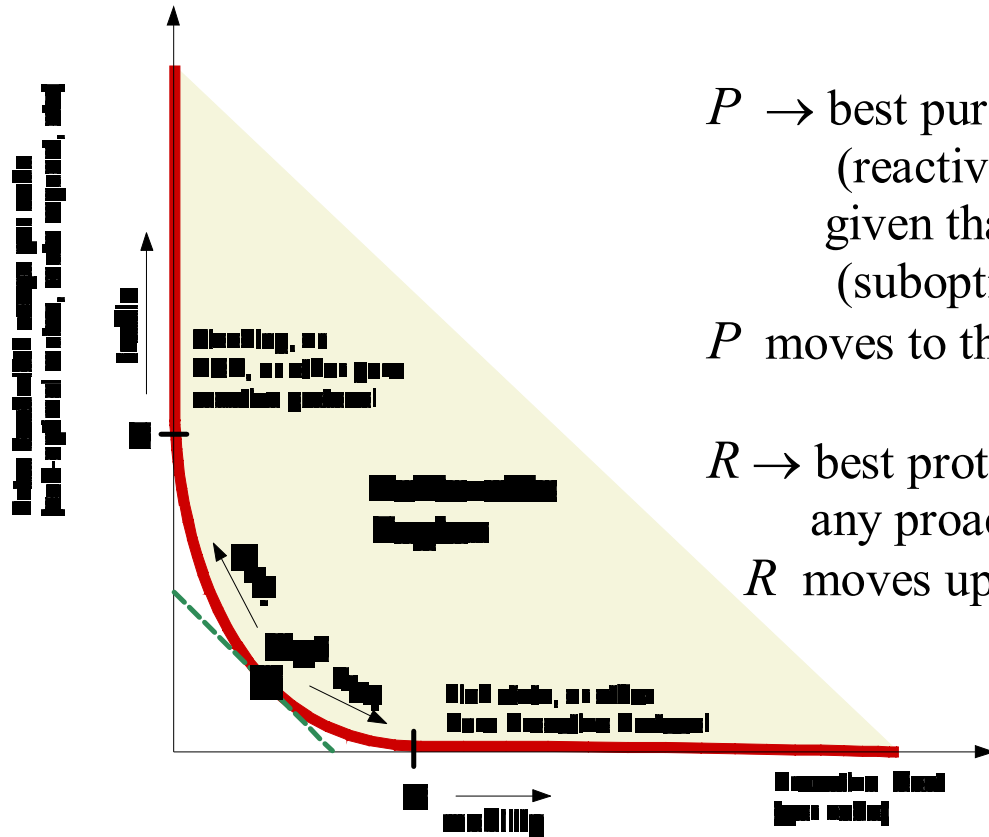
Achievable Regions & Operating points

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 → 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 → best protocol that does not use
 any proactive information

R moves upward as traffic increases or
 diversifies

the best protocol
 (in terms of
 overhead)

→

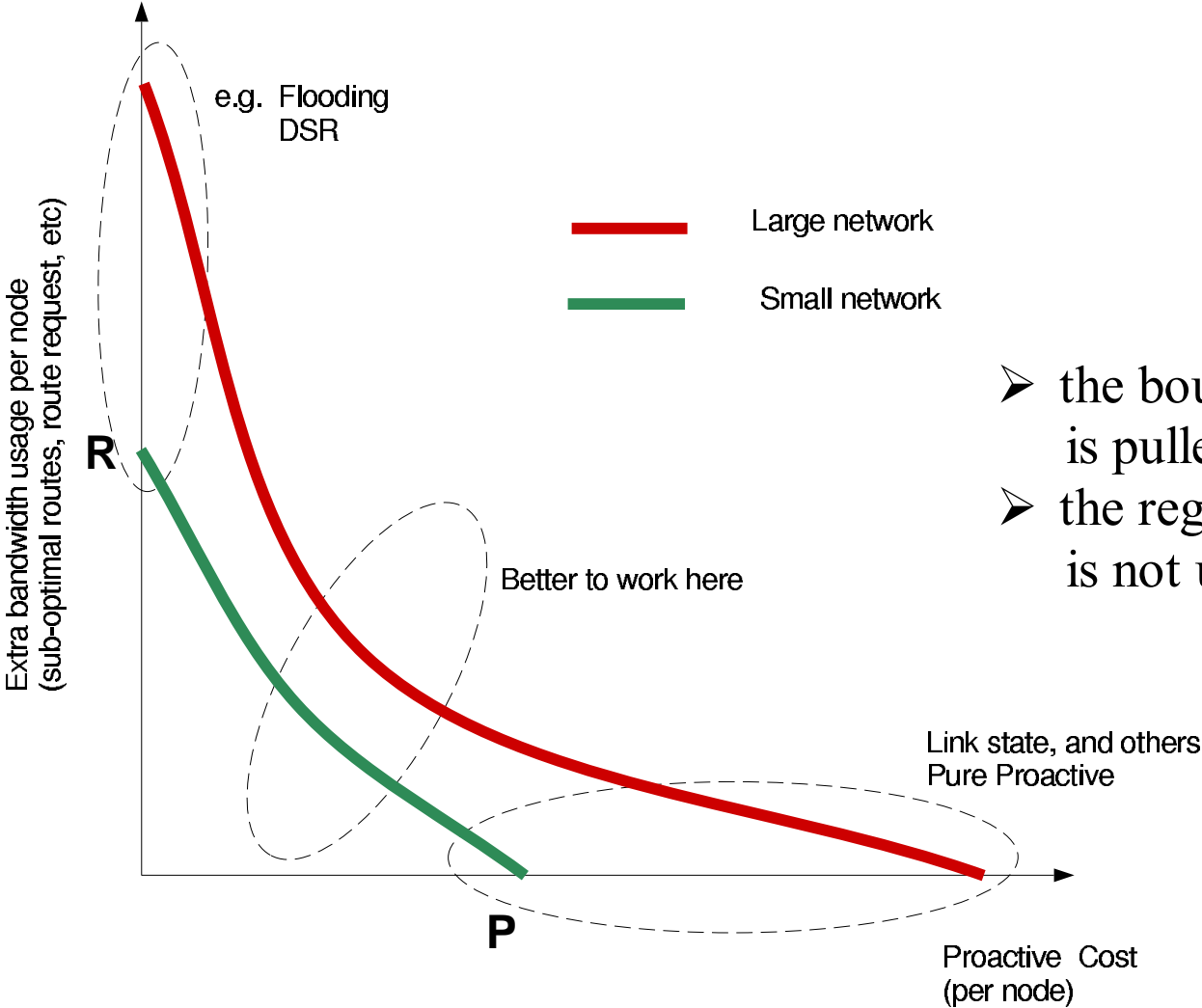
minimizes
total overhead

⇔

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]

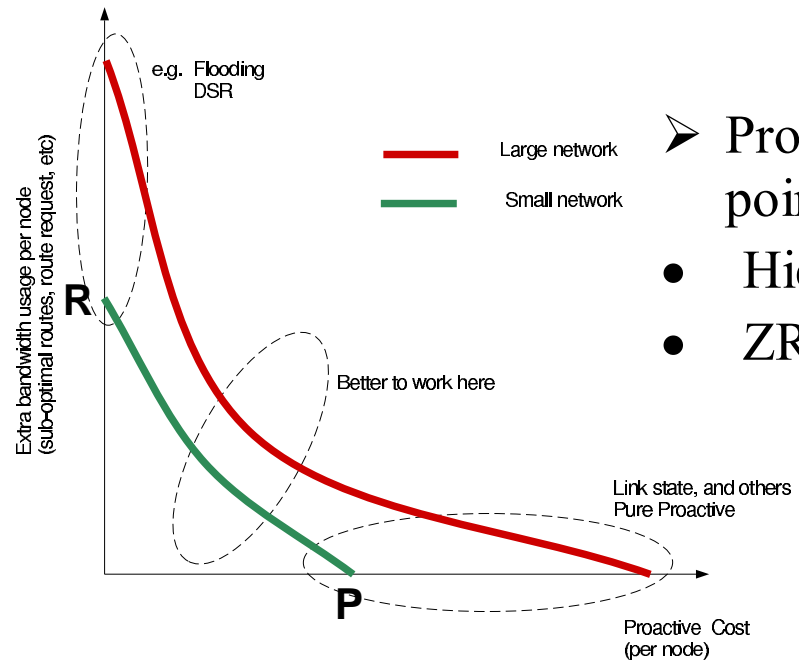
Each control message [RREQ message] is retransmitted by each node

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

The number of message retransmissions increases linearly with N

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

$$\Psi_{\lambda_i} \stackrel{\text{def}}{=} \lim_{\lambda_i \rightarrow \infty} \frac{\log Tr(\lambda_1, \lambda_2, \dots)}{\log \lambda_i}$$

- 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* $Tr(\lambda_{lc}, \lambda_t, N) = \Theta(\lambda_t N^{1.5})$
 $\Rightarrow \Psi_{\lambda_{lc}} = 0, \Psi_{\lambda_t} = 1, \text{ and } \Psi_N = 1.5$

Network's scalability (Quantitatively)

- to assess whether a network is *scalable* w.r.t. to λ_i
the *network rate* dependency on λ_i must be considered

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

For the *network rate* ($R^{network}$) computation 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:

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

Network's scalability (Quantitatively)

For example, class of networks under study (assumptions a.1-a.8)
(resulting from applying power control techniques)

$$\Rightarrow \Psi_N = 1.5$$

to be regarded as scalable with respect to network size $\Psi_N \leq 1$

(It has been shown [Gupta, Kumar 2000] that in mobile ad hoc networks $\Theta(N)$ successful transmissions can be scheduled simultaneously. I.E., the *network rate* is $\Theta(N)$)

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

Network's scalability (Quantitatively)

Note: It has been shown in [Groossglauser&TSE, Infocom 2001] that if the network applications can support infinitely long delays and the mobility pattern is completely random, then the average path length may be reduced to 2 ($\Theta(1)$) regardless of network size and, as a consequence, that network *scalability factor* with respect to network size Ψ_N is equal to 1. Thus, 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. This work does not consider that class of networks since they have no practical relevance.

Network's scalability (Quantitatively)

- Wireline networks: if fully connected may have $\Psi_N = 1$
⇒ potentially scalable w.r.t. network size (in the bandwidth sense)

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

Similarly,

- the *network rate* does not increase with mobility or traffic load
(Gupta – Gunar result: $\Theta(N)$)
⇒ a network will be scalable w.r.t. mobility $\Leftrightarrow \Psi_{\lambda_c} = 0$
w.r.t. traffic $\Leftrightarrow \Psi_{\lambda_t} = 0$

⇒ the **networks under study**

- **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. additional parameters

For example,

- as transmission range increases the spatial reuse decreases (assuming a infinite size network with regular density)
- *network rate* decreases as rapidly as ℓ^2

⇒ Ψ_ℓ should be lower than -2 for the network to be *scalable*

⇒ **ad hoc networks are not scalable w.r.t. transmission range**
(the *minimum traffic load* decreases only linearly w.r.t. $\ell \Rightarrow \Psi_\ell = -1$)

⇒ **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 scalability 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 then the routing protocol is not degrading network performance (determined by the *minimum traffic load*), even if the overhead happens to grow faster than the *network rate*.

Routing Protocol's scalability (Quantitatively)

Let $X_{ov}(\lambda_1, \lambda_2, \dots)$ be the *total overhead* induced by routing protocol X , dependent on parameters $\lambda_1, \lambda_2, \dots$ (e.g. network size, mobility rate, data generation rate, etc.). Then, the Protocol X 's *routing protocol scalability factor* with respect to a parameter λ_i ($\rho_{\lambda_i}^X$) is defined to be :

$$\rho_{\lambda_i}^X \stackrel{\text{def}}{=} \lim_{\lambda_i \rightarrow \infty} \frac{\log X_{ov}(\lambda_1, \lambda_2, \dots)}{\log \lambda_i}$$

- 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 the parameter** λ_i if and only if, as the parameter 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. mobility rate $\Leftrightarrow \rho_{\lambda_{ic}}^X \leq 0$
 - *scalable* w.r.t. traffic $\Leftrightarrow \rho_{\lambda_i}^X \leq 1$

Scalability dimensions

- network size is a key parameter, but not the only one (mobility, network density, network diameter, traffic diversity, energy etc.)
- Examples :
 - diameter $\uparrow \Rightarrow$ latency for control $\uparrow \Rightarrow$ inconsistent routes, instability \uparrow
 - density \uparrow spatial reuse $\downarrow \Rightarrow$ capacity \downarrow

dimension	Size	Mobility	Density	Diameter
layer				
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
(diameter and size are mutually dependent, density is not a limiting factor)

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 $\exists c_1, n_1: c_1g(n) \leq f(n), \forall n \geq n_1$

➤ $f(n) = O(g(n))$ if $\exists c_2, n_2: f(n) \leq c_2g(n), \forall n \geq n_2$

➤ $f(n) = \Theta(g(n))$ if $\exists c_1, c_2, \text{ and } n_0: c_1g(n) \leq f(n) \leq c_2g(n), \forall n \geq n_0$

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. $\frac{1}{2}$ of the nodes)
- focus on expected (mean) behavior
(for a specific network topology this assumption may not hold)
- geographical reasoning may not define one hop connectivity
(where multipath fading, obstacles, etc. are more important)
however, it strongly influences connectivity
(as observed according to larger scales)

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 $\Theta(d * k)$.

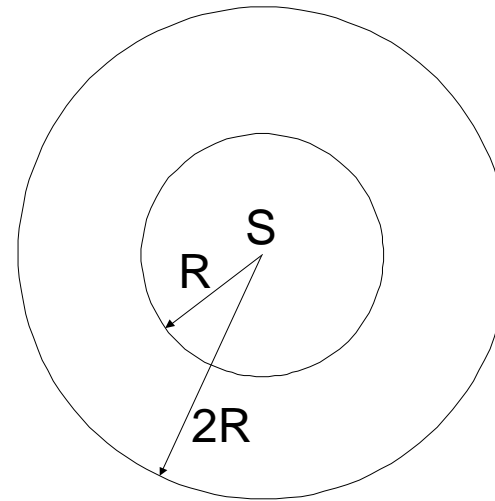
a.4 : The maximum and average path length (in hops) among nodes in a connected subset of n nodes both increase as $\Theta(\sqrt{n})$. In particular, the maximum path length across the whole network and the average path length across the network (L) increases as $\Theta(\sqrt{N})$.

➤ 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 $R \rightarrow 2R$

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

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 $\Theta(N)$)
 - 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 \Rightarrow 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 under study, a routing protocol X is:

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

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)
 - ⇒ additional bandwidth required = $data(N - 1 - L)\lambda_t N$ bps
- PF total RO /sec = $\Theta(\lambda_t * N^2)$
 - ⇒ $\rho_{\lambda_c}^{PF} = 0$, $\rho_{\lambda_t}^{PF} = 1$, $\rho_N^{PF} = 2$

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 (in 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 = $\Theta(\lambda_{lc} N^2)$

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

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 \sim 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):

$$E\{(\log_2 N)(L_i^{opt})^2\} \geq (\log_2 N)E\{L_i^{opt}\}^2 = (\log_2 N)L^2 \text{ bits}$$

➤ $\lambda_i N$ packets are transmitted / sec

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

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

DSR without Route Cache (DSR-noRC)

$$\begin{aligned} \text{DSR-noRC reactive RO / sec} &= \Omega(\lambda_s N^2) \\ &= O((\lambda_s + \lambda_{lc}) N^2) \end{aligned}$$

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

➤ DSR-noRC total overhead / sec = $\Omega(\lambda_s N^2 + \lambda_t N^2 \log_2 N)$

➤ $\rho_{\lambda_t}^{DSR-noRC} = 1, 0 < \rho_{\lambda_{lc}}^{DSR-noRC} \leq 1, \rho_N^{DSR-noRC} > 2$

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 = $\Theta(\lambda_t N^{1.5+\delta})$

$$\begin{aligned} \Rightarrow \text{HierLS-LM1 } \textit{total overhead} / \textit{sec} \\ = \Omega(s * N^{1.5} + \lambda_{lc} * N + \lambda_t N^{1.5+\delta}) \end{aligned}$$

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

(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* = $O(\lambda_t N)$

⇒ 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

Zone Routing Protocol (ZRP)

- lower bound for 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

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

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 require much more memory and processing power (however, the main challenge is the excessive bandwidth)
- require careful design (else may result in much more bandwidth consumption than hierarchical)

Techniques for bandwidth consumption reduction:

(in isolation or in combination)

- efficient flooding
- limited generation
- limited dissemination

Efficient Flooding

- classical flooding is a 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

- limit the amount of control information

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

- 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

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 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 protocol (as SLS) do not scale well with size (the 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 reduction on proactive RO will induce an increase in suboptimal RO)

FSLs Algorithms

Family of Fuzzy Sighted Link State (FSLs)

- 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 is set to a value
(specifies how far the LSU is propagated)
that is a function of the current time

FSLs Algorithms

Under a FSLs protocol

& after one global LSU transmission (TTL = ∞)
(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)

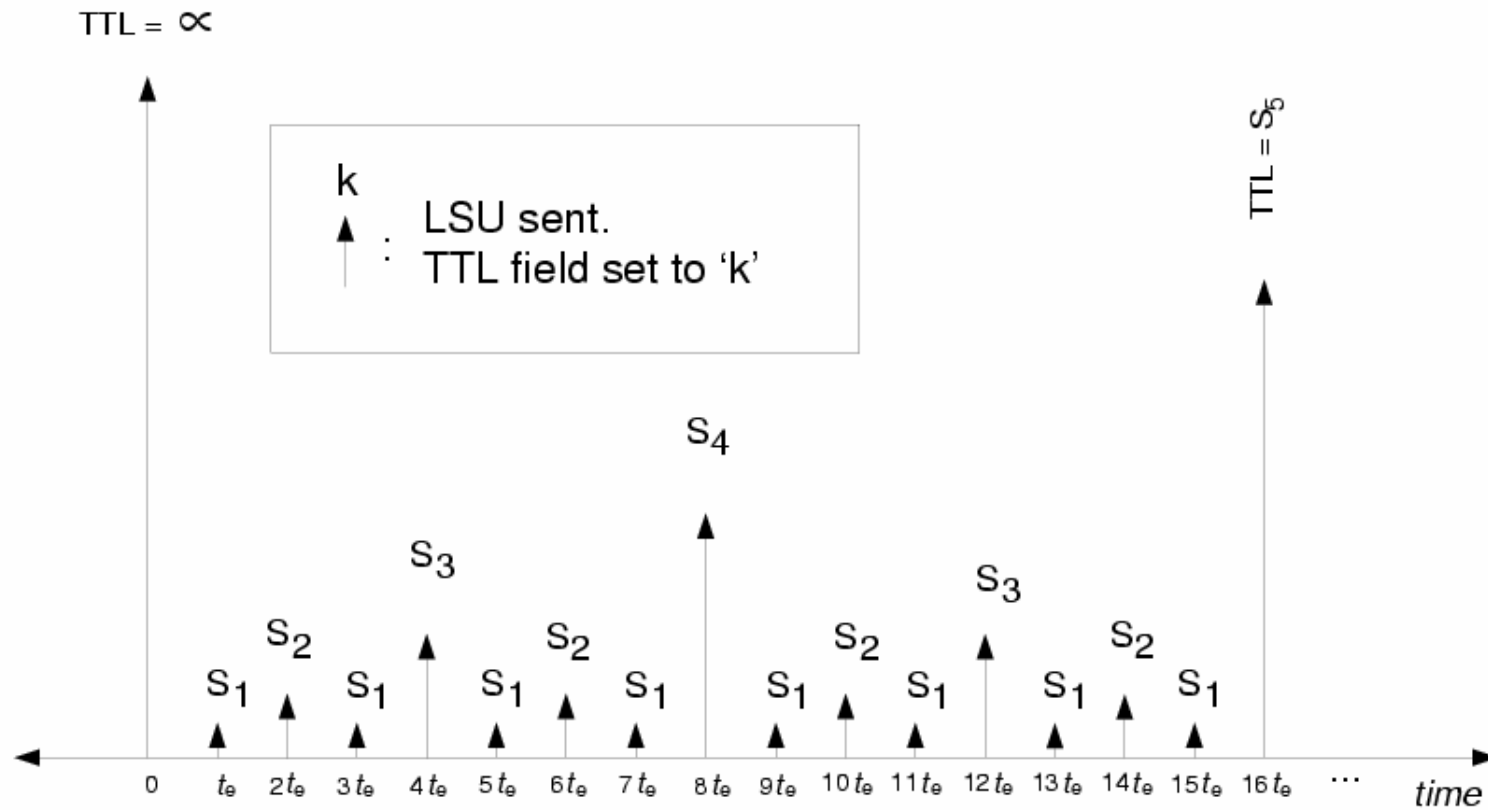
...

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, \dots 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

⇒ 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

⇒ 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 $12t_e$.)

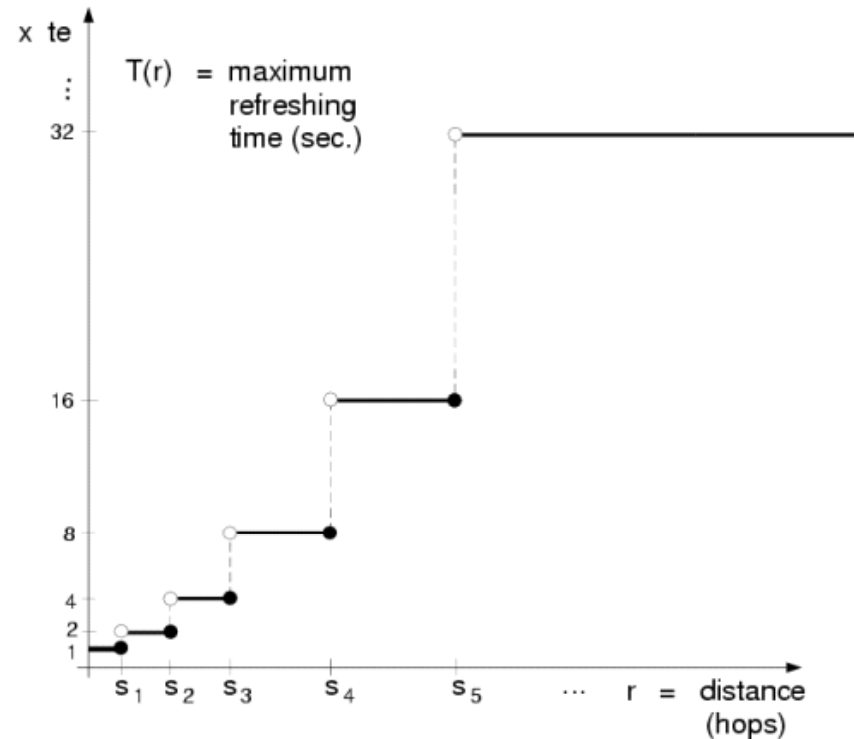
FSLS Algorithms

➤ 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.

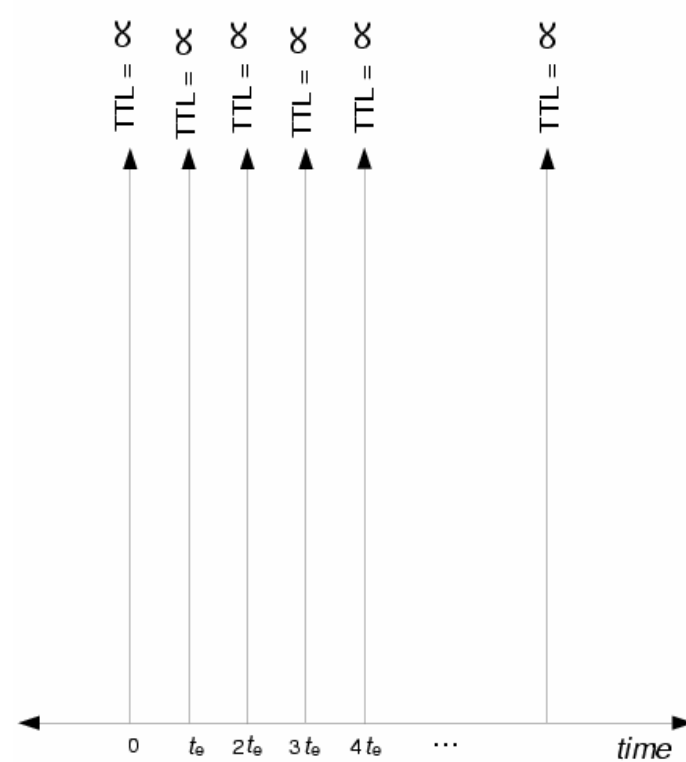
- Discretized Link State (DLS)
- Near Sighted Link State (NSLS)



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

FSL 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)

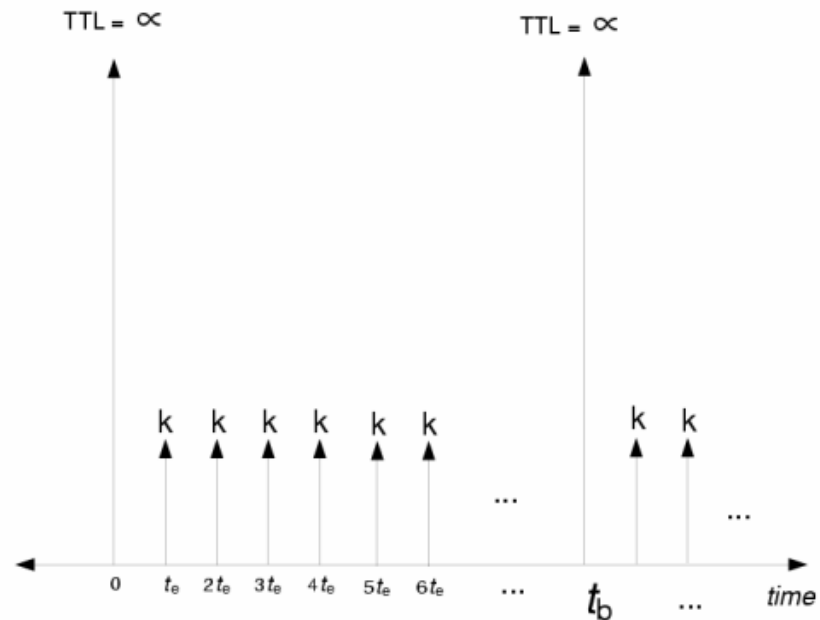
FSL Algorithms: Discretized Link State (DLS)

- 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 than $\frac{1}{t_{flood}}$)
- Under high mobility (LSUs are sent every t_e seconds) DLS induces the same proactive overhead as GSR (by setting $t_e = t_{flood}$)
(they both require control packets transmission of the equivalent of N times the average topology table size (in bits) each t_e (t_{flood}) seconds)

FSL 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

Problem

- 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

- 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.

FSL 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

FSL 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

⇒ NSLS induces the same control overhead as FSR

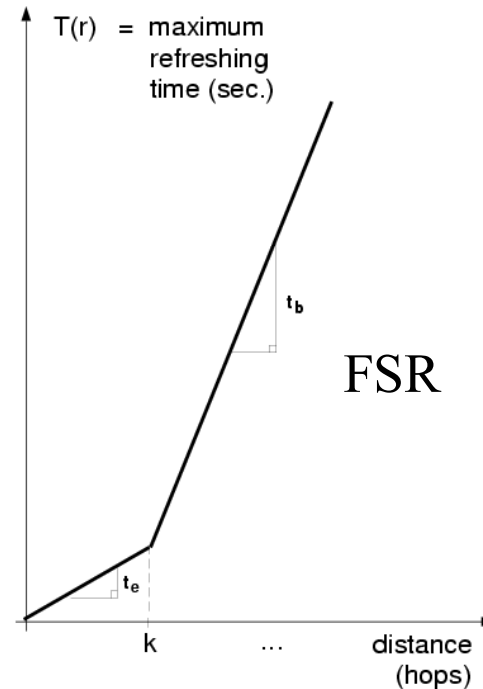
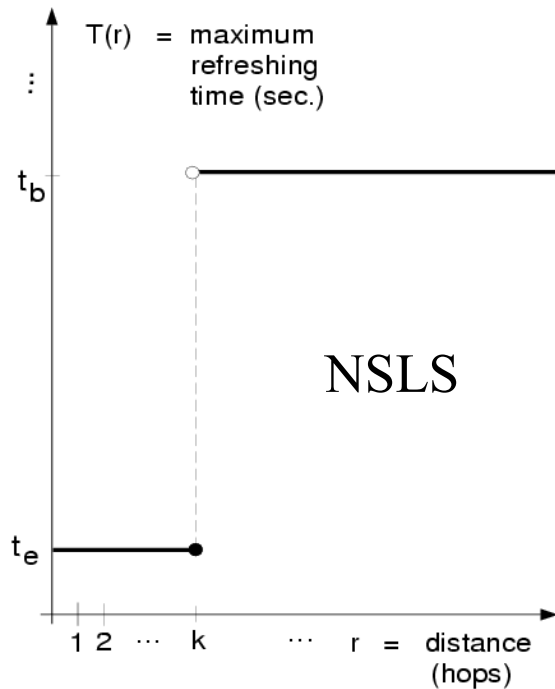
FSLS Algorithms: NSLS

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

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

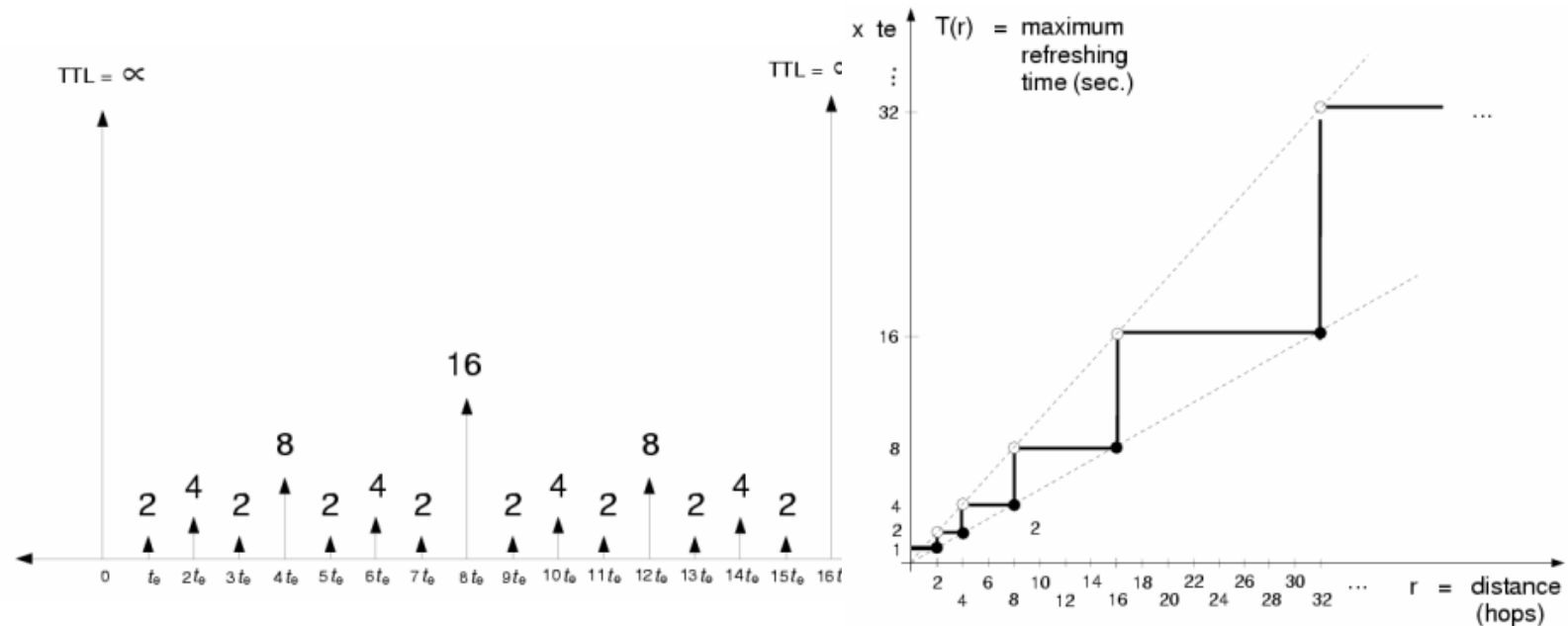
FSR: $T(r) = t_e * r$ for $r \leq k$, and $T(r) = k * t_e + (r - k) * t_b$

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



FSL 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 \Rightarrow 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 to the entire network (link state, route request, or data)
⇒ 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)
⇒ 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)
⇒ 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* \sim 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 routing
(to improve quality)
2. contradicts 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 of 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
($\lambda_c = \Theta(\lambda)$ 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's 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 depends on the number of hierarchical levels
(which depend on the cluster size (independent of the traffic load))
⇒ 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.))

⇒ 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: DAWN project modification of the MMWN clustering protocol
 - 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 ($9 \leq \text{cluster size} \leq 35$)
 - 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
 - high implementation complexity
 - hard to analyze
 - 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

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