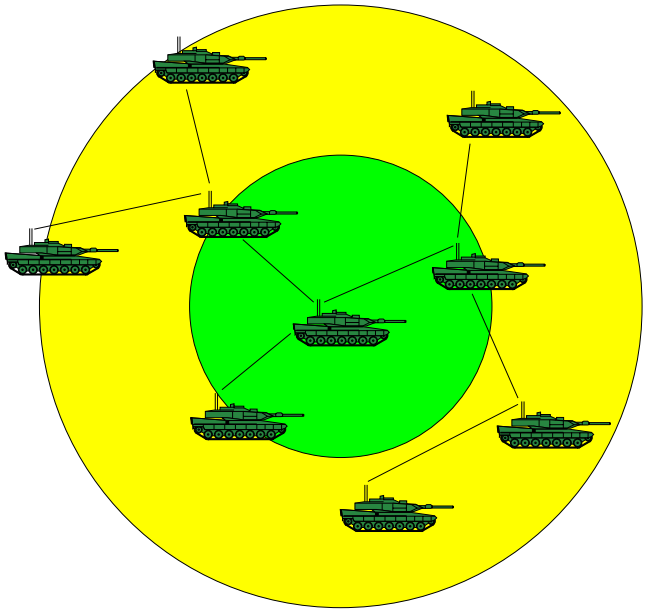
FOI-R- -1058- -SE
December 2003
ISSN 1650-1942
Methodology report

Katarina Persson, Erika Johansson,
Ulf Sterner, Mattias Sköld

The Fisheye Routing Technique in Highly Mobile Ad Hoc Networks



Command and Control Systems
P.O. Box 1165
SE-581 11 LINKÖPING
SWEDEN

FOI - SWEDISH DEFENCE RESEARCH AGENCY
Command and Control Systems
P.O. Box 1165
SE-581 11 LINKÖPING
SWEDEN

FOI-R--1058--SE
December 2003
ISSN 1650-1942
Methodology report

Katarina Persson, Erika Johansson,
Ulf Sterner, Mattias Sköld

The Fisheye Routing Technique in Highly Mobile Ad Hoc Networks

Issuing organization FOI - Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 LINKÖPING SWEDEN	**Report number, ISRN** FOI-R--1058--SE	**Report type** Methodology report
	Research area code 4. C^{4}ISR	
	Month year December 2003	**Project No.** E7035
	Customers code 5. Contracted Research	
	Sub area code 41. C^{4}I	
Author/s Katarina Persson, Erika Johansson, Ulf Sterner, Mattias Sköld	**Project manager** Mattias Sköld	
	Approved by Lennart Nyström	
	Sponsoring Agency FM - The Swedish Armed Forces	
	Scientifically and technically responsible Jan Nilsson	
Report title The Fisheye Routing Technique in Highly Mobile Ad Hoc Networks		

Abstract

In tactical operations, it is important to have a robust and reliable communication system. Mobile ad hoc networks are distributed multi-hop wireless networks that allow the nodes to freely move through the terrain and still be able to exchange information. Topology changes in the network may occur often, and a routing protocol that finds and upholds routes for information through the network is of great importance.

In this report we investigate how the Fisheye State Routing (FSR) protocol works in a highly mobile ad hoc network. We conclude that the Fisheye technique is efficient in the studied networks and that it is crucial to choose the protocol's parameter settings based on the actual network capacity. Higher user capacity can be achieved by adapting the settings when the network environment changes. We also found that it is possible to combine the transmitting of Situation Awareness messages and the update messages used in the FSR protocol. However, to fulfill the Situation Awareness demands of accuracy, frequent update messages are required which leads to a high amount of routing traffic generated in the network.

Keywords routing, fisheye, FSR, ad hoc network, QoS, Situation Awareness, SA	
Further bibliographic information	**Language** English
ISSN 1650-1942	**Pages** 56 p.
	Price acc. to pricelist

Utgivare Totalförsvarets Forskningsinstitut - FOI Ledningssystem Box 1165 581 11 LINKÖPING	**Rapportnummer, ISRN** FOI-R- -1058- -SE	**Klassificering** Metodrapport
	Forskningsområde 4. Spaning och ledning	
	Månad, år December 2003	**Projektnummer** E7035
	Verksamhetsgren 5. Uppdragsfinansierad verksamhet	
	Delområde 41. Ledning med samband och telekom och IT-system	
Författare Katarina Persson, Erika Johansson, Ulf Sterner, Mattias Sköld	**Projektledare** Mattias Sköld	
	Godkänd av Lennart Nyström	
	Uppdragsgivare/kundbeteckning FM - Försvarsmakten	
	Teknisk och/eller vetenskapligt ansvarig Jan Nilsson	
Rapportens titel Fisheye-routing i ad hoc-nät med hög mobilitet		

Sammanfattning

Ett robust och säkert kommunikationssystem är viktigt vid taktiska operationer. Mobila ad hoc-nät är trådlösa flerhoppsnät där noderna kan röra sig fritt genom terrängen och samtidigt överföra information. Topologiförändringar kan ske snabbt i nätet och ett routingprotokoll som söker reda på vägar genom nätet för informationen samt upprätthåller dessa är av största vikt.

Vi studerar i den här rapporten hur protokollet Fisheye State Routing (FSR) fungerar i ad hoc-nät med hög mobilitet. Vi drar slutsatsen att Fisheye-tekniken är effektiv i våra nät men att det är viktigt att välja parameteruppsättningar för protokollet efter vilken nätverkskapacitet som finns. Med adaptiva parameteruppsättningar kan kapaciteten utnyttjas ännu effektivare. Det är också möjligt att kombinera sändning av positionsrapporteringsmeddelanden (SA) med uppdateringsmeddelandena i FSR-protokollet. Det krävs dock täta uppdateringsmeddelanden för att uppfylla kraven som SA-tjänsten ställer vilket leder till att mycket routingtrafik genereras i nätet.

Nyckelord

routing, fisheye, FSR, ad hoc-nät, garanterad tjänstekvalitet, Situation Awareness, SA

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN 1650-1942	**Antal sidor:** 56 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Contents

1 Introduction **9**

2 Fisheye State Routing **11**
2.1 Data Structures 11
2.2 Sending Update Messages 12
2.3 Receiving Update Messages 15
2.4 Our Parameter Design 15

3 Situation Awareness (SA) Service **19**
3.1 SA Service 19
3.2 Distributing SA Information Using Routing Update Messages . 20

4 Performance Measures **23**
4.1 Definitions of Routing Traffic and Route Length 23
4.2 Utilization of Network Capacity 24
4.3 How Old is the SA Information? 25

5 Scenario and Assumptions **29**
5.1 Assumptions 29
5.2 Network Movement 32

6 Results **35**
6.1 Accessibility 35
6.2 Routing Traffic 37
6.3 Route Length 38
6.4 Utilization of Network Capacity 38

6.5 FSR and SA 41

7 Conclusions **45**

8 Future Work **47**

A **49**

Chapter 1

Introduction

Radio networks are important components in tactical operations. In some situations the network needs to allow the radio units, so-called nodes, to freely move through unknown terrain while still exchanging information. To achieve tactical goals, the network must also sometimes be able to operate without the use of pre-deployed infrastructure. One method for obtaining area-coverage and robustness in this type of network is to enable the nodes to relay messages, thus creating a so-called *multi-hop* network. To further improve the robustness of the network, there should be no central nodes, i.e. the network management should be distributed. Such mobile distributed multi-hop networks are usually referred to as *ad hoc* networks.

A military ad hoc network must also supply a wide category of services [1], e.g. group calls, situation awareness data, and intranet connections. The different services can have different *Quality of Service (QoS)* demands, i.e. different demands on delay, packet loss ratio, throughput, etc. An important component in providing these services is the routing protocol, i.e. the protocol that finds and determines by which route through the network a packet should be forwarded on its way to its destination. The issue of finding a suitable routing protocol for QoS in a military ad hoc network has been examined in [2]. The Fisheye State Routing (FSR) protocol [3, 4, 5] was found interesting in this report since the routing traffic is attenuated as it propagates through the network. This is a proactive routing protocol that gives a node a good picture of the network near itself, while it only has a vague picture of the network far away. The Fisheye technique for spreading routing information is also similar to that of the Situa-

tion Awareness service, where position information is disseminated through the network.

A classical evaluation method for routing algorithms is to study the algorithms behavior when the mobility is varied while keeping the network capacity and the algorithms parameter settings relatively constant, see e.g. [6, 7]. In this paper we analyze how efficient the FSR protocol is in a highly mobile ad hoc network where the total network capacity is varied. We investigate the importance of the FSR parameter settings on the overall behavior of the algorithm, on the network capacity available for user traffic, and on the accessibility a node has to other nodes in the network. Furthermore, we try to determine whether Fisheye State Routing can be used together with a Situation Awareness (SA) service.

We conclude that the protocol is efficient in our highly mobile ad hoc networks since use of the Fisheye technique reduces the routing control traffic while yielding a sufficient quality of routes. We also see that different FSR parameter settings have a great effect on the user capacity available. From our simulations, it can be determined that if a fixed parameter setting is to be used, it must be based on the lowest capacity that can occur at any time in the network. It is also possible to gain capacity by dynamically adapting these parameters to network changes. Furthermore, it was possible to transmit SA messages together with routing control traffic and still fulfill the demands of position accuracy.

Thesis outline

The report is organized as follows. In Chapter 2 we introduce the Fisheye State Routing protocol. In Chapter 3 we outline the Situation Awareness service. Some definitions and equations necessary for the remainder of the report are presented in Chapter 4. This is followed in Chapter 5 by a description of the scenario at hand and of the assumptions and limitations we have used, and a presentation of some basic traits of our networks. Our simulation results can be found in Chapter 6, and in Chapter 7 we draw some conclusions. Finally, in Chapter 8, we comment on future work. Some tables and figures that are useful for further understanding can be found in the appendix.

Chapter 2

Fisheye State Routing

In this chapter we describe the Fisheye State Routing (FSR) protocol for wireless ad hoc networks [4]. It is a proactive link state protocol whose objective is to keep control traffic low and still provide accurate information about the routes. In Chapter 2.4 we introduce the parameters related to the protocol used and subsequently analyzed in our simulations.

The FSR protocol uses the Fisheye technique, which was originally used to reduce data required to represent graphical data. A node's perception of its surroundings, according to this technique, is similar to that of a fisheye, where the level of detail is high near the "focal point" and decreases with the distance from the focal point. This means that when a user packet is sent, the intermediate nodes will have increasingly better routing information available as the packet approaches its destination and will use this to gradually improve the route.

2.1 Data Structures

Each node running FSR has to maintain a neighbor list, a topology table, and a routing table. In the neighbor list, the node keeps the addresses of all nodes one hop away and the time they received the last information from that neighbor. If the node does not receive any link state information from a neighbor for a certain interval, the neighbor is removed from the neighbor list.

In the topology table, information about all destinations in the network is stored. From this table, the node creates the routing table used for routing data

messages. Each entry in the topology table consists of:

- A destination address
- A destination sequence number. This number is used to determine whether the information is new or not
- Last heard time, i.e. the last time this entry was updated
- A list of the neighbors to each destination
- A flag for "Need To Send", (NTS). This flag is set, for example, when new information is added to a list entry and means that the node wishes to send this entry to its neighbors.

The topology table is updated through update messages sent from neighbors. An example of an update message is shown in Figure 2.1.

2.2 Sending Update Messages

To obtain lower levels of overhead traffic in a mobile network, the generation of update messages is not event-driven, but periodic. When a node obtains a new neighbor or loses one, it updates its neighbor list, topology table, and routing table. However, the node does not generate an update message immediately.

To generate the nodes' perception of their surroundings, each node divides the network into a number of *scopes*. For a certain node v_j, a scope is defined as the set of destinations that it can reach within a given set of hops. The number of scopes used to cover the network, and how they are chosen, i.e. how their borders are chosen, differs, and is not specified in [3, 4]. An example where three scopes are used is shown in Figure 2.2. Here, the scopes are defined so that they contain nodes one hop away, two hops away, and three hops away, respectively.

To each scope i we assign a time T_s^i which decides how often node v_j may transmit information about the nodes in scope i to its neighbors. To reduce the amount of traffic that is transmitted, the node only includes information about nodes with certain periodicities and if the NTS flag is set to *true*. Furthermore, the node sets the NTS flag to *true* for the entry in the topology table that contains the node's own topology data with the periodicity T_u regardless whether any of the node's neighbors have changed.

Packet length	**Reserved**
Destination address no. 1	
Destination Sequence number no. 1	**Number of neighbors** N_1
Neighbor address 1 **to destination** 1	
Neighbor address 2 **to destination** 1	
⋮	
Neighbor address N_1 **to destination** 1	
⋮	
Destination address no. M	
Destination Sequence number no. M	**Number of neighbors** N_M
Neighbor address 1 **to destination** M	
⋮	
Neighbor address N_M **to destination** M	
⋮	

Figure 2.1: *Proposed format of an update message [4].*

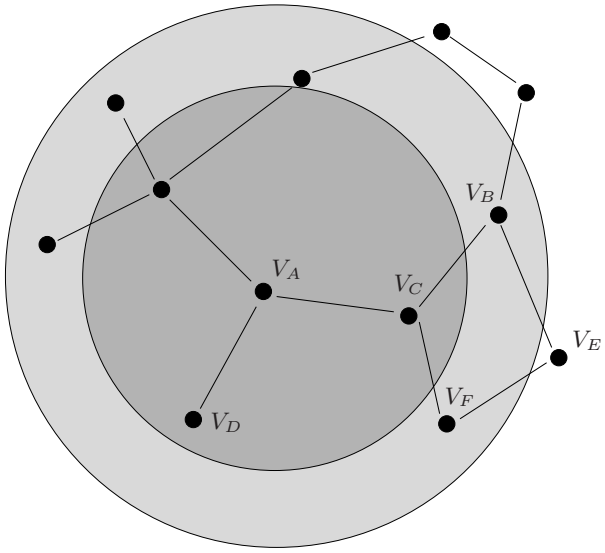


Figure 2.2: *In this example three scopes are used when the network is represented the way node v_A experiences it. Every node in the network divides its destinations into different scopes.*

2.3 Receiving Update Messages

When a node receives an update information message, it first checks whether the sender is a neighbor and if it is included in the neighbor list. New neighbors are added, and the time the node last received an update message from the neighbor is updated. Then the topology table is updated with the information. The contents of all entries in the packet are studied. If it is a new destination, a new topology table entry is created, and the NTS flag is set, since this information needs to be forwarded. Otherwise, if the entry already exists in the topology table, the sequence numbers are checked and compared. If the incoming entry has a larger sequence number than the existing number, the information has changed, and the entry in the table needs to be updated. The NTS flag is set since the information is new.

If, instead, the sequence number is lower, this means that the sender is not updated with the latest information. No changes are made in the topology table but the NTS flag is set and the information is forwarded. If the sequence number is equal to the previous number nothing happens. When new information has been added to the topology table, the routing table is also updated with the most recent information about routes in the network. Information is also discarded when it is too old.

2.4 Our Parameter Design

We have also chosen to let each node divide its surroundings into different scopes i with a time T_s^i to determine how often information about destinations in this scope can be sent. We will describe this time as

$$T_s^i = \delta \cdot \textit{scope update factor} = \delta \cdot \text{round}(h^\alpha),$$

where δ is the minimum time between two updates, h is the distance in number of hops to the nodes in this scope, and α determines the grade of attenuation in the network. If we use $\alpha = 0$, all nodes will belong to the same scope i, i.e. the Fisheye technique will not be used. When using $\alpha \leq 1$, we know that the number of scopes is equal to the maximum scope update factor, *round*(h^α), and we will get large scopes in this case. If, for example, $\alpha = 0.5$, all nodes within two hops will belong to scope 1 since the scope update factor is 1 for both

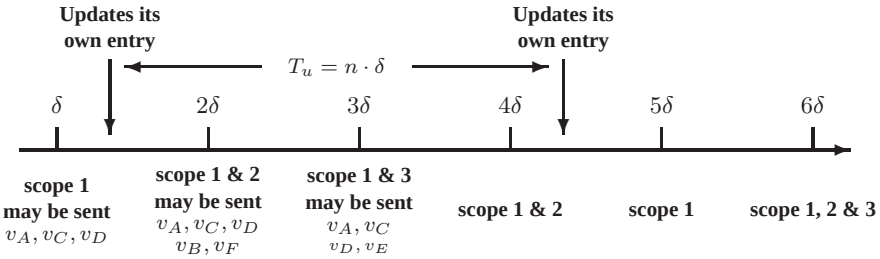


Figure 2.3: *An example of how the timers and scopes work when* $\alpha = 1$ *and* $n = 3$.

$h = 1$ and $h = 2$. If, instead, $\alpha > 1$, the number of scopes is always equal to the maximum number of hops. The larger the value of α we use, the greater the scope update factors will differ, i.e. the faster the routing traffic will attenuate. Furthermore, we assume that the periodicity of the nodes self-generated update messages is

$$T_u = n \cdot \delta,$$

where n is a parameter we can change to improve the performance of the algorithm.

To exemplify the sending of update massages we use the example in Figure 2.2. We assume that the NTS flag is set for node $\{v_A, v_B, v_C, v_D, v_E, v_F\}$ in the topology table in node v_A. Furthermore, we assume $\alpha = 1$ and $n = 3$. The destinations belong to different scopes and are therefore not sent out with the same periodicity. This means that with a period of δ only destinations that belong to the current scope may be sent. This can be seen in Figure 2.3.

Depending on where on the time axis we are in this example, for each δ we look in our topology table and pick out the entries with the NTS flag set. Then we need to check whether they belong to the current scopes. At time $t = \delta$, we can transmit information about node v_A, v_C and v_D, but we cannot transmit information about node v_B and v_F until $t = 2\delta$.

A second timer is used to ensure that each node periodically transmits messages. With a period $T_u = n \cdot \delta$, the nodes update their own sequence number and set the NTS flag. This is done even when no changes have occurred. As a result, there is at least one new entry to transmit. Since every node does this, many entries in the topology table will be updated and control traffic will flow

even in a stationary net. It is also obvious from this that routing information that is not always of importance will flow through the network.

Chapter 3

Situation Awareness (SA) Service

We briefly describe the SA service and provide examples of demands on position accuracy for this service. Furthermore, we analyze how distribution of SA information could be combined with routing control traffic.

3.1 SA Service

Having data concerning other nodes in the network, including their position, speed, and direction of movement, is becoming increasingly important both for avoiding unfortunate incidents (e.g. friendly fire) and for maintaining information superiority. Therefore future communication networks will likely be expected to support SA services. It is also likely that demands on position accuracy for a node, for example, will vary, usually depending on the distance between the node and one's own node, see Table 3.1.

As an example of demands on an SA service, we present user requirements in respect of position information. These requirements originate from scenarios developed by, or in cooperation with, military personnel. In [1] and [8] these requirements are presented as demands for position accuracy (in meters) for nodes at different distances. In [9] and [10], the demands are expressed instead as maximum time (in seconds) between updates.

If the maximum velocity of a node is known, the demands for accuracy in

Distance	[8]	[10]	[9]	[1]
$\leq$ 3 km	1 s	1 s	1 s	0.5 s
$\leq$ 10 km	10 s	10 s	10 s	5 s
$\leq$ 15 km	10 s	60 s	-	-
$\leq$ 30 km	25 s	60 s	-	-
$>$ 30 km	-	60 s	-	-

Table 3.1: *Example of required SA position accuracy for nodes with maximum speed 70 km/h. The demands are expressed as maximum time (in seconds) between received updates from a node at a certain distance.*

meters can be converted into accuracy in seconds. For the scope of this report, time between updates is the more convenient alternative. An example of the resulting different demands is shown in Table 3.1, where we assume that the maximum speed of a unit is 70 km/h.

3.2 Distributing SA Information Using Routing Update Messages

One method of distributing SA information is to attach the SA data to all outgoing packets. Another method would be to attach the SA data to only some of the transmitted packets. But which ones? User traffic can be sporadic, both as to when it is generated and to whom it is transmitted. Thus it cannot be used since there is no guarantee that the SA data will reach all the nodes that may be concerned. When using a proactive routing protocol, such as Fisheye State Routing (FSR), the nodes continuously try to uphold routes to one another. This means that periodically there will be routing control traffic flowing through the network. An efficient method of distributing SA data might thus be to "piggyback" the SA data onto existing control traffic. Another advantage of this is that the update messages already contain the node names. If the routing algorithm and the SA service could exchange information, that would further improve the combination. The routing algorithm could, for example, use the positions and velocities of other nodes to predict route changes.

The amount and instances of routing control traffic varies, however, due for

example, to the movements of the node. In a stationary network, where routes have been established, control traffic is only generated by the updates each node transmits about itself to its neighbors with period $T_u = n \cdot \delta$. The information is then propagated throughout the network, eventually reaching all nodes, see Chapter 2 for more information. These updates are the only packets that are transmitted no matter what. Hence, they are the ideal candidates for piggybacking SA data on. This, however, imposes the SA update demands on the FSR protocol, i.e. only some FSR parameter choices will result in the fulfillment of a given set of SA demands.

Chapter 4

Performance Measures

In this chapter the methods for analyzing our results are presented. We will define how we measure the route length and the amount of routing traffic, but also how we combine them to analyze the fraction of the network's capacity that can be utilized for user data. Finally, we present a method for calculating the delivery time for SA messages.

4.1 Definitions of Routing Traffic and Route Length

When analyzing the performance of a routing protocol, the amount of routing traffic and the length of the found routes are of great interest. The availability of the nodes is also of importance, i.e. how many of the node pairs that are connected by single- or multi-hop. Minimum route length can often be achieved if we accept a high cost. If we update the protocol immediately when changes occur, we will obtain high quality routes, but we will also generate a great amount of routing traffic. If we instead minimize the routing traffic, our routing table will contain inaccurate information, which will give us bad routes.

Routing traffic, RT, is defined as the total average amount of control traffic per second generated in the network during the simulation. We have not considered that the routing traffic is inserted into packets, nor how these packets are suited into the time slots of the Medium Access Control (MAC) protocol when we have analyzed this parameter.

Route length is measured in number of hops. When using FSR to generate

the routes, optimal route length is not always achieved. We have compared the length of the routes that FSR has found with optimal route length, i.e.

$$\text{Route length difference} = \frac{\hat{h}_r - \hat{h}_o}{\hat{h}_o}$$

where $\hat{h}_o$ is the average length of the optimal routes and $\hat{h}_r$ is the average length of the routes found by the FSR. In cases where FSR has failed to find a route to a destination that is possible with an optimal routing algorithm, no comparison has been made. Furthermore, our comparison of route length does not take into consideration how many of the possible routes that were actually found.

4.2 Utilization of Network Capacity

To get a measure that takes into account both the route length and the cost of finding good routes, we assume that the total network capacity is G Bytes/s. If all routes were optimal and we ignore the cost of finding them, the users could transmit $\lambda_{max} = G/h_o$ Bytes/s through the network, where h_o is the average length of a route. In a more realistic network, capacity is lost due to routing traffic, imperfect routes, and incorrect routes, see Figure 4.1, i.e.

$$\begin{aligned}\text{Network Capacity} = {} & \text{User traffic} + \text{Routing traffic} \\ & + \text{Imperfect routes traffic} + \text{Incorrect routes traffic}\end{aligned}$$

We define S as the traffic generated in the network by the users. If ϵ is the fraction of incorrect routes in the network, $S\epsilon$ is the part of the user traffic that will be lost. Furthermore, $U = S(1-\epsilon)$ is the part of the user traffic that is successfully delivered to its destinations. We can then estimate the extra traffic caused by longer, imperfect routes as $U(h_r - h_o)$. The traffic caused by incorrect routes is estimated as $Sh_o(1-\epsilon)$, i.e. we assume that the traffic is sent through the network and discarded at the (wrong) destination, see Figure 4.1. Using these assumptions, we get

$$G = \lambda_{max} \cdot h_o = Uh_o + RT + (h_r - h_o)U + \frac{\epsilon}{1-\epsilon} h_o U.$$

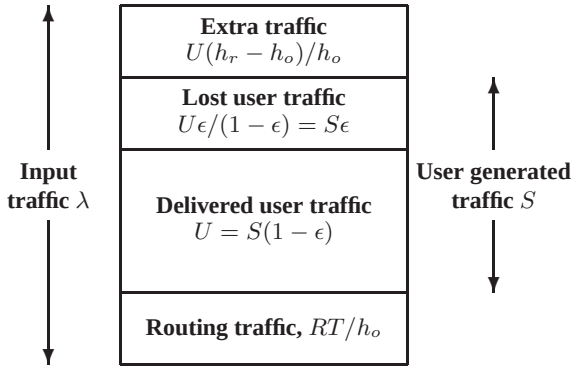


Figure 4.1: *Illustration of the partitioning of the total network capacity into different kinds of traffic.*

To find optimum parameter settings for different network capacities G, we can then maximize the fraction of user traffic, i.e.

$$\frac{Uh_o}{G} = \frac{U}{\lambda_{max}}.$$

This measure expresses how efficiently the total network capacity is utilized when using a certain routing algorithm in the network.

4.3 How Old is the SA Information?

As mentioned in Chapter 3.2, the only way to attach SA data to routing control traffic and still be able to guarantee distribution within a specific time frame is to attach the data to the updates each node transmits about itself to its neighbors. As mentioned before, these updates have a period of

$$T_u = n \cdot \delta,$$

hence the FSR parameters δ and n become important factors in fulfilling the demands of maximum time (in seconds) between SA updates. Furthermore, it is not enough to send SA information to immediate neighbors only. Nodes

further away are also likely to require the information. It is therefore necessary to calculate the delivery time of an update packet (with SA data attached) to any given node in the network, i.e. how long it takes for the packet to transverse different distances. Examples of update demands for position information from other nodes, as a function of their distance to the receiving node, are presented in Chapter 3.1.

Using the "worst case" approach, the delivery time Δ_{DT} of an update packet to a node at distance h hops can be calculated as

$$\Delta_{DT}(h,\alpha,\delta,n) = \begin{cases} T_u + \Delta_{delay} = n\cdot\delta + \Delta_{delay}, & h = 1 \\ \Delta_{DT}(h-1,\alpha,\delta,n) + \Delta_{delay} + \delta(h-1)^{\alpha} & h \geq 2 \end{cases}$$

where the FSR parameters α, δ, and n are defined in Chapter 2. The equations can also be rewritten in a more compact form as

$$\Delta_{DT}(h,\alpha,\delta,n) = \begin{cases} T_u + \Delta_{delay} & h = 1 \\ T_u + h\cdot\Delta_{delay} + \delta\sum_{k=2}^{h}(k-1)^{\alpha} & h \geq 2. \end{cases}$$

Furthermore Δ_{delay} is the total "worst case" delay before the receiving unit gets the packet once the FSR algorithm has determined that it is its turn for (re)transmission. This delay is calculated as

$$\Delta_{delay} = \Delta_{transm} + \Delta_{MAC} + \Delta_{queue}, \tag{4.1}$$

where Δ_{transm} is the time it takes a transmission to travel the longest possible hop, Δ_{queue} is the time spent in queue at the node, and Δ_{MAC} is the maximum delay at a node until the MAC (Multiple Access Control) protocol allows the node to transmit.

We can now calculate how old the SA information, regarding a certain node, is at a node as

$$\Delta_{SA}(h,\alpha,\delta,n) = \Delta_{DT}(h,\alpha,\delta,n) \;\; \forall h \tag{4.2}$$

when we use the definitions above. We can then, given the minimum length of a hop, compare this with the demands in Chapter 3.1. We must thus choose our FSR parameters α, δ, and n so that

$$\Delta_{SA}(h,\alpha,\delta,n) \leq \Delta_{demand}(h) \;\; \forall h \tag{4.3}$$

is fulfilled if we want to provide a good SA service.

An example of this is shown in Figure 4.2. Here node B is at a distance $h = 1$ hops from node A. Node A transmits its position as it passes 1, this message is received in node B a time Δ_{delay1} later (just as node A passes position 2). A time T_u after passing position 1, it is again time for node A to transmit its position (position 3). The new position is received by node B after Δ_{delay2}, and by then node A is at position 4.

The maximum inaccuracy in position information occurs just before node B receives the second position message - at this time node B presumes node A to still be at position 1 when it is almost in position 4. The position information in node B is at this time

$$T_u + \Delta_{delay2}$$

seconds old. This is in accordance with equation (4.2) where $h = 1$ and $\Delta_{delay2} \leq \Delta_{delay}$.

Node B passes the SA information on to node C, this means that the NTS-flag for node A is set in node B's topology table. Within δ seconds, this will be noted since node A is in node B's first scope (regardless of α) and a packet will be transmitted. This packet will reach node C a time $\Delta_{delay3} \leq \Delta_{delay}$ seconds later. The resulting maximum position inaccuracy in node C, regarding node A, will thus be

$$T_u + \Delta_{delay2} + \delta + \Delta_{delay3} \leq \Delta_{DT}(2, \alpha, \delta, n).$$

Node C then passes the information on to node D. Depending on the value of α, node A might be in any of node C's scopes and must wait until it is time to transmit information regarding the correct scope. The waiting time is $\delta(h-1)^{\alpha}$ seconds, the information is then transmitted and reaches node D a time $\Delta_{delay2} \leq \Delta_{delay}$ later.

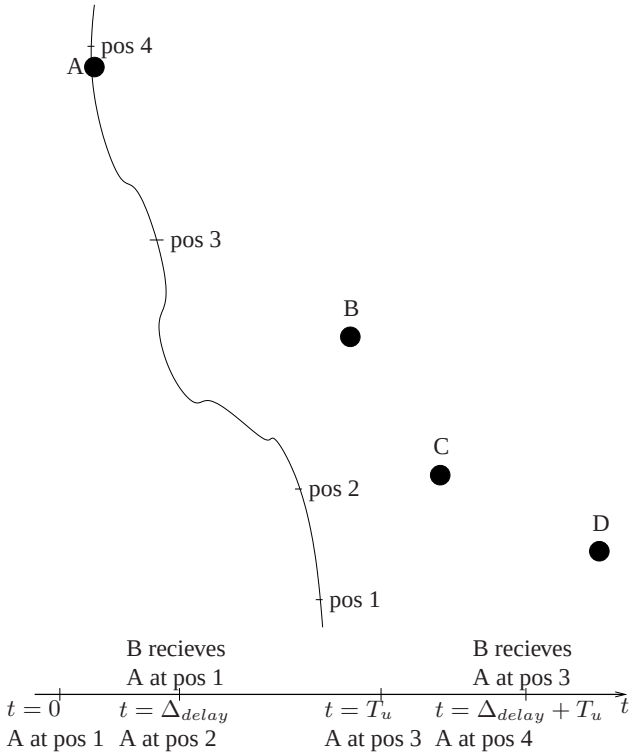


Figure 4.2: *An example of the ageing of position information. Node B receives SA information regarding node A at the same time as it receives routing update messages. The information is then passed on to node C and D.*

Chapter 5

Scenario and Assumptions

In this Chapter the scenario used for simulations is presented. Furthermore, to be able to study the Fisheye State Routing protocol, it was necessary to limit the problem somewhat and make certain assumptions. These limitations and assumptions will be described and motivated in Chapter 5.1. Finally, some basic traits of our simulated network are presented in Chapter 5.2.

5.1 Assumptions

The Scenario and the Mobility Model

In our scenario, we have 64 nodes from a mechanized battalion. The nodes communicate by radio and support multi-hop packet delivery. The radio network is also assumed to be decentralized, i.e. all nodes are equal, thus increasing robustness. Nodes can join or leave the network. The network can be divided into several smaller networks, e.g. due to terrain obstacles, and then united again. Furthermore, the nodes use a *Situation Awareness* (SA) service to keep track of each others movements.

The nodes are deployed in an area of 4x4 km. They move at a constant velocity of 70 km/h, i.e. the maximum speed possible for a mechanized battalion, thus resulting in a highly mobile scenario. To simplify simulations, no actual mechanized battalion mobility patterns have been used. Instead, the nodes are assumed to be independent of each other and are assigned a new random direction at certain intervals. When a node reaches the border of the battlefield,

it is assumed to turn back into the area in the same way that a ball bounces (inelastically) off a wall, and then proceed in this new direction.

The Link Model

To put the FSR protocol in focus, we have assumed that the path loss encountered on a certain link is only a function of the link's physical length (according to the plane-earth propagation model), i.e. no terrains have been used. Furthermore, all nodes use the same transmission power, and we have assumed that no congestion and no packet loss take place in the network.

We also assume that the MAC protocol is distributed (see e.g. [11] or [12]) and that it can also handle some form of scheduling (see e.g. [13] or [14]). This means that the MAC protocol needs to maintain certain network information. An important feature of this information, from a routing point of view, is that a node is thus aware of at least its neighbors. A protocol that does this is a simple Time Division Multiple Access (TDMA) protocol where each node has its own time slot, i.e. no spatial reuse or traffic adaptation is used. This protocol will thus be used for our *SA calculations*.

Routing Traffic

To get a picture of the network throughput due to routing control traffic, we have made some assumptions concerning address lengths and packet sizes. At a given moment, a node wants to share a number of rows in its routing table with its neighbors. Each row in the routing table, see Chapter 2.1, consists of

- the address of the route destination
- the node's latest known sequence number for the destination in question, and the number of neighbors to that node.
- the addresses for the node's neighboring nodes, see Chapter 2.

We have assumed that the size of an address is 16 bits, and that the destination sequence number and information about the number of neighbors requires another 16 bits. The average size of a packet containing one row of the routing table (an entry) can then be calculated as

$$\text{Average row size} = A \cdot (N_n + 1) + I = 16 \cdot (N_n + 1) + 16 \text{ [bits]},$$

where A is the size of an address, N_n is the average number of neighbors a node has in the simulation, and I is the information the package carries about each destination's sequence numbers and its number of neighbors. At a maximum, a routing packet can carry information about all nodes in the network, i.e. in this case 64 rows.

SA Calculations

The delivery of SA information is done according to Chapter 3.2. To be able to calculate the delivery time Δ_{DT} of a SA packet to a unit at distance h hops as in Chapter 4.3, some further assumptions have to be made.

To calculate the time it takes for a transmission to travel the longest possible hop, we need to know the distance covered by one hop. In [1], it is assumed that it takes at most two hops, i.e. one transmission and one retransmission, to traverse a distance of 3 km. From this assumption, we conclude that we cover 1.5-3.0 kilometers with one hop.

We also assume that there is no queuing for routing traffic in the network, i.e. $\Delta_{queue} = 0$. This is a somewhat ideal approach but becomes more realistic if we assume that small packets that arrive at a node are "bundled together" and transmitted in the same time slot.

To calculate Δ_{delay} in equation (4.1), we also have to consider the Medium Access Control (MAC) protocol we use, i.e. TDMA. Furthermore, for these calculations, we assume that an update packet with attached SA information has a size of 500 bytes and that the link capacity is 2 Mbit/s. It can be noted here that in our simulations each node had approximately 10 neighbors, which means that in a packet of 500 bytes, 20 rows from the routing table can be fitted into one packet. The length of a time frame in a TDMA protocol where each node has one slot per frame to transmit in can be calculated as

$$t_{frame} = N_{tot}\frac{S}{C},$$

where N_{tot} is the total number of nodes in the network, S is the maximum size of a packet that can be transmitted in one slot, and C is the maximum link capacity. Using the assumptions made above, the length of a time frame in the TDMA protocol used for our SA scenario is

$$t_{frame} = 64\frac{8 \cdot 500}{2 \cdot 10^6} = 0.128\,s.$$

The length of a time slot in this protocol is $t_{slot} = \frac{S}{C} = 2$ ms. For this protocol, the longest delay before a node is due to transmit is the frame length. That is, in equation 4.1, the worst possible situation with regard to the MAC protocol is when $\Delta_{MAC} = t_{frame}$. By further use of our previous assumptions, we can now calculate the maximum value of Δ_{delay} as

$$\Delta_{delay} = \frac{3000}{3 \cdot 10^8} + 0.128 + 0 = 0.12801\, s, \tag{5.1}$$

since the transmission range is 1.5-3.0 km and we assume that there are no queues.

5.2 Network Movement

To get a picture of what occurs in the network during the simulations, we will show some basic results. Depending on the scenario, the amount of node pairs in the network that are connected by single- or multi-hop will vary. If we have 100% connectivity, every node in the network can reach any node in the network during the whole simulation. In a mobile network, it is of interest to see how mobility affects the number of links that go up or down at any given time, and the duration of the links. These link parameters are shown below in Table 5.1 for the different connectivities, ϕ. It can be noted that the link parameters are quite similar between the two lower connectivities, and that a link spends 3-7 times as much time being down as being up.

When comparing the mean number of neighbors that a node will have for the different connectivities, see Table 5.1, it can be noted that in a fully connected network of 64 nodes, every node is in direct contact with more than 20% of the possible destinations. For lower connectivities, this figure is lower.

In Figure 5.1, the probability density functions of the route length for the three connectivities are depicted. When the network has full connectivity, both the mean route length and the maximum route length are at their lowest. This is a natural cause of the fully connected networks larger number of neighboring nodes. A network that is not fully connected is generally more sensitive to performance losses than a network with 100% connectivity. The route length is longer in such a network, and a consequence of this is that if changes occur, the delays in the network will be larger than in a fully connected network. We thus focus the remainder of this report on networks with the demands of 90% and

	ϕ 90%	ϕ 95%	ϕ 100%
Mean time a link is up [s]	57	65	87
Mean time a link is down [s]	391	359	288
Mean number of changes in the network [1/s]	8.1	8.7	10.0
Mean number of neighbors	7.4	8.9	13.9
Mean route length [hops]	5.6	5.2	3.5
Max route length [hops]	24	23	9

Table 5.1: *Some parameters that describe our 64-node networks and the movement of the nodes.*

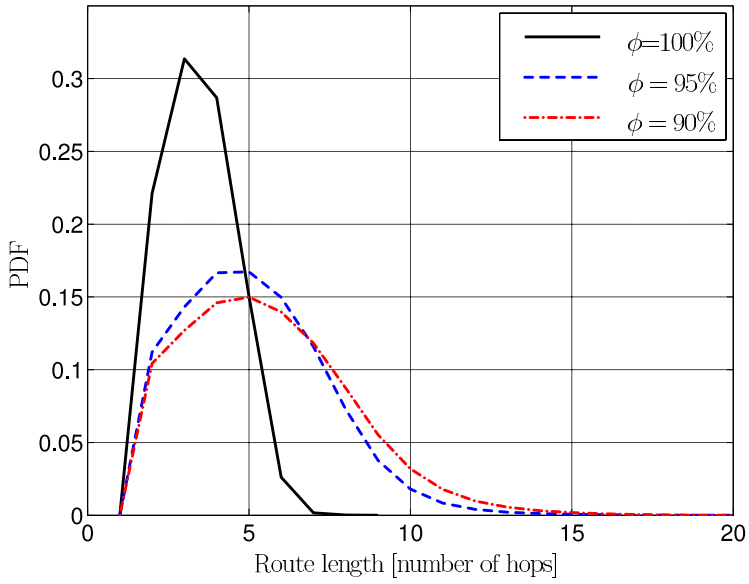


Figure 5.1: *Probability density functions of route lengths for the three different connectivities 90 %, 95 % and 100 %. We can see that the route length increases when the connectivity decreases in the network.*

95% connectivities, where we believe that the effect of the routing algorithm is of greater importance.

Chapter 6

Results

This chapter presents the results from the different simulations carried out on a network of 64 nodes and with the simulation time 3600 s. The connectivity in the simulated networks, i.e. the amount of node pairs that are connected, is 90% and 95%. The parameter α, used to distinguish between scopes in Fisheye, varies from 0 to 3.0, the time when a node will check its topology table to see whether there is any information that needs to be sent, δ, varies from 0.1 to 4.9 seconds in the simulations. The parameter n that defines how often the node will update its own entry and set its own NTS flag, expressed in times δ, will vary from 1 to 6.

First we present some basic results concerning accessibility in the network, routing traffic, and route length. In Chapter 6.4 we study the degree of utilization of different network capacities, and in 6.5 the combination of SA service and FSR is discussed.

6.1 Accessibility

An optimal routing algorithm would find a route between any node pair that can be connected through single hop and multi-hop. Due to delays and inaccuracy in routing tables, FSR will not find all these routes. To see how well the FSR algorithm performs, the percentage of routes that FSR found, compared to those found by an optimal routing algorithm, was studied. This percentage represents a users *accessibility* to other nodes in the network. In figure 6.1, examples of

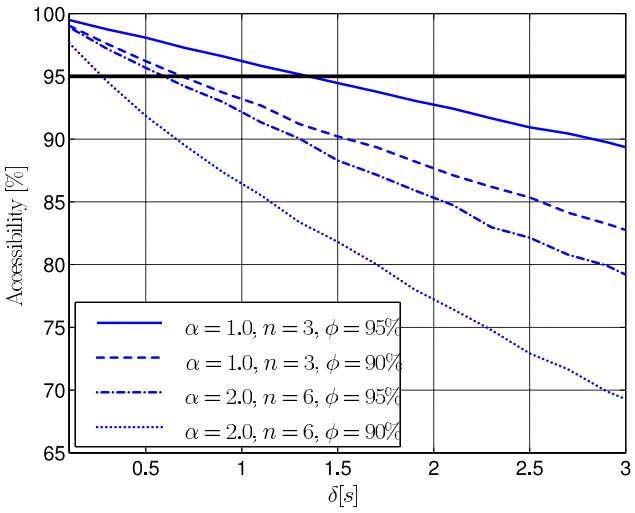


Figure 6.1: *Accessibility as a function of δ for a few combinations of α, n and the network connectivity.*

accessibility for different connectivities ϕ and with different parameter settings are shown. We can see from this figure that accessibility decreases faster for a network with 90% connectivity than for a network with 95% connectivity. One reason for this is that the routes are longer in a network with low connectivity and these routes are more difficult to find, see Chapter 5.2. We have required that accessibility must be at least 95%, irrespective of the connectivity in the network. By this demand we are forced to limit the possible parameter settings.

If we require that FSR finds at least 95% of the possible routes, we can (for each combination of α and n) find a maximum value of δ that fulfills this requirement, δ_{max}, see Figure 6.1. If, for example, we look at accessibility when $\alpha = 1.0$, $n = 3$, and the network is connected to 95%, we see that for $\delta \leq 1.3$ s the requirement is fulfilled. Accessibility decreases as α, n and δ increase, due to more seldom updates of the routing table. The maximum δ for all different combinations of α and n is shown in Table A.3 and Table A.2 for connectivities 90 and 95%, respectively.

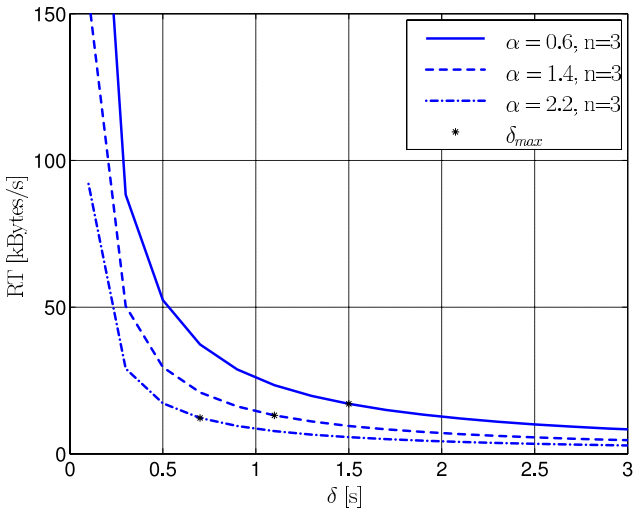


Figure 6.2: *Amount of routing traffic generated in our network for different settings on α and n when the network is connected to 95%.*

6.2 Routing Traffic

An important consideration regarding routing protocols is the amount of routing traffic that is generated, i.e. how much overhead is necessary to maintain the routes. It is, of course, desirable to minimize this traffic, thus leaving as much as possible of the network capacity to user traffic. In this section, we will show the results for FSR when we have tried to minimize the routing traffic.

When n increases, updates messages are generated with a larger periodicity. If δ and α also increase, update messages are sent more seldom and the routing traffic is attenuated. We thus get lower amounts of routing traffic when we increase δ, α, and n. However, we also get lower accessibility and worse routes, see Chapter 6.1.

Our demand for at least 95% accessibility gives us a maximum valid value of δ. For a given set of FSR parameters (α, n), the minimum traffic load is always given by δ_{max}. This result is consistent for all choices of FSR parameters; an example is depicted in Figure 6.2. The control traffic decreases as δ increases, due to more seldom updates. If we let $\delta = \delta_{max}$ in a network with connectivity

95%, we will find that the routing traffic is minimized for $n = 1$ and $n = 2$ when $\alpha = 1.6$. The difference between these parameter combinations is low, however, and we can conclude that the choice of δ is the most important parameter if we wish to reduce the routing traffic. Routing traffic load for the two connectivities and all combinations of α and n when $\delta = \delta_{max}$, is shown in Table A.4 and Table A.5.

6.3 Route Length

To evaluate the quality of routes a routing algorithm gives us, we measure route lengths. We have compared the route length FSR generates with the optimum route length for different parameter settings. Route length was measured in hops, and only routes that FSR found were compared with the optimum route between the same pair of nodes. The accessibility is not considered here.

When comparing the different parameter settings, the lowest route lengths were found for low δ, α and n. This is because the information about the routes improves if we update the tables often. In Figure 6.3, route length difference is shown as a function of δ for a few parameter combinations. This shows that α and n do not affect the performance as much as δ does. It is crucial to choose a low δ to be able to find the best routes. Unfortunately, this will generate a high amount of routing traffic.

6.4 Utilization of Network Capacity

We have seen in Chapter 6.2 and 6.3 that different optimum values of parameter settings were found when routing traffic and route length were minimized. If we optimize the routing algorithm for minimizing the routing traffic, we get high values for δ, α, and n thus generating rather bad routes. If we instead optimize on route length, low values of δ, α, and n will be chosen and this will generate a high amount of routing traffic. Neither of the two suggestions will thus give the routing algorithm a good overall performance

To be able to find a good parameter setting that minimizes both routing traffic and route length, we instead look at the fraction of total network capacity that can be used for user traffic, see Chapter 4.2. The total network capacity is divided into routing traffic, extra traffic, lost traffic, and user traffic. In an

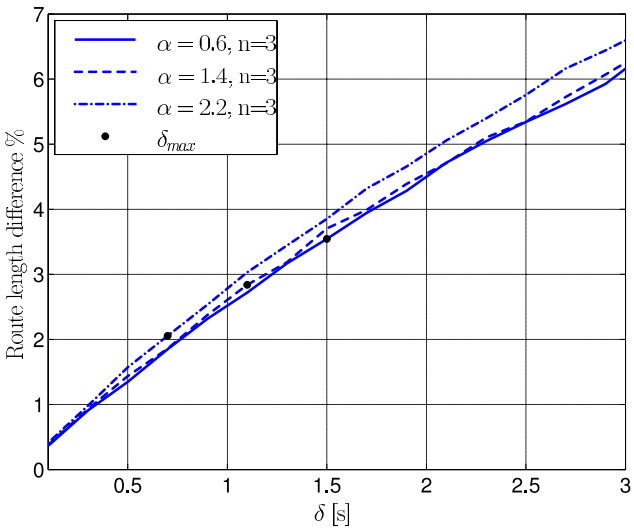


Figure 6.3: *Quality of routes with FSR.*

optimum network no routing traffic is needed, and all routes are found and are perfect. Here we want to maximize the fraction of user traffic for different network capacities, see Equation 4.1. This will give us optimum parameter settings for different network capacities.

In Figure 6.4 we can see the fraction of user traffic that is possible as a function of maximum network capacity for a few different cases when we have 95 % connectivity in the network. The algorithm needs a certain amount of network capacity to be able to update the routing tables and retain accessibility above 95%.

The choice of parameter settings is important for obtaining good performance. The user traffic for fixed parameter settings are shown in Figure 6.4 as dashed curves. The parameter settings here are chosen to maximize performance for a specific network capacity, G. These curves show that for a given parameter setting, the fraction of network capacity available to the user is greatly affected by the total network capacity. If a fixed parameter setting is used, it should be based on the lowest network capacity that might occur, otherwise the loss in user traffic can be considerable. For example, if the network is optimized

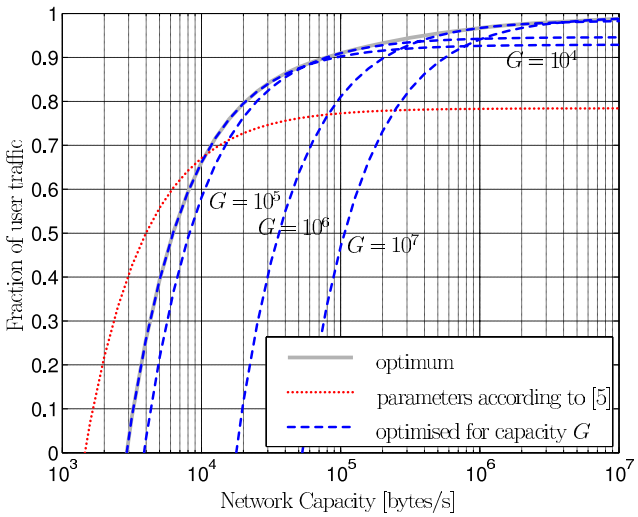


Figure 6.4: *Fraction of the maximum user capacity available for user traffic in the network.*

for a capacity $G = 10^7$ bytes/s while the actual network capacity is $G = 10^5$ bytes/s, the capacity available to the user drops from 90% to 50% of the total network capacity.

The solid curve in Figure 6.4 shows the optimum performance for our scenario when using adaptive parameter settings, i.e. when the optimum FSR parameter setting is chosen for each network capacity. For the capacities shown here, the optimum user capacity is achieved for $\alpha \approx 1.5$. This elucidates that the Fisheye technique, for our type of networks, yields a higher capacity available for user traffic than a proactive link-state protocol without this technique, i.e. where $\alpha = 0$. However, if the total network capacity further increases, the values for α, δ, and n decreases. This is due to the fact that for very high capacities, it is important to keep the route length close to optimum, while the routing traffic becomes insignificant.

In [4], there is a parameter proposal for FSR. When used in our network, it results in an accessibility of 84%, thus not meeting our requirement. Still, this dotted curve is shown in Figure 6.4, and it is obvious that this choice keeps the user from fully taking advantage of the network capacities. The corresponding

results from our simulation with 90% connectivity can be found in Figure A.1, and the optimum parameter settings for different network capacities are shown in Table A.1.

6.5 FSR and SA

The assumptions in Chapter 5.1 and in Chapter 5.1 enable us to calculate the total "worst case" delay $\Delta_{delay} \approx 0.12801$ seconds, see equation (4.1) and (5.1). If we use this numeric value in equation (4.2), we get

$$\Delta_{SA}(h,\alpha,\delta,n) = \begin{cases} T_u + 0.13 & h = 1 \\ T_u + h \cdot 0.13 + \delta \sum_{k=2}^{h} (k-1)^{\alpha} & h \geq 2. \end{cases}$$

Thus it becomes clear that the age of the position information, i.e. the delivery time for the chosen type of routing control packet (see Chapter 3.2), to a node only depends on the FSR parameters n, δ, α and the number of hops h the node is from the source of the SA information.

Given the assumptions in Chapter 5.1, it is also possible to translate the demands of maximum time between SA updates at different distances, see Chapter 3.1, into maximum time between SA updates at different number of hops away from the source. By choosing a specific set of demands from Chapter 3.1, we can calculate for which FSR parameter settings the set of demands can be fulfilled. For a distance of h hops, it is then possible to express equation (4.3) as

$$\begin{gathered} \Delta_{SA}(h,\alpha,\delta,n) \leq \Delta_{demand}(h) \\ \therefore \begin{cases} T_u + \Delta_{delay} \leq \Delta_{demand}(1) & h = 1, \\ T_u + h \cdot \Delta_{delay} + \delta \sum_{k=2}^{h} (k-1)^{\alpha} \leq \Delta_{demand}(h) & h \geq 2 \end{cases}. \end{gathered} \tag{6.1}$$

For a distance of 3 km, i.e. at most 2 hops, this can be simplified to

$$\begin{aligned} \Delta_{demand}(2) &\geq n \cdot \delta + 2 \cdot \Delta_{delay} + \delta \sum_{k=2}^{2} (k-1)^{\alpha} \\ &= 0.26 + \delta(n+1) \end{aligned}$$

This renders n and δ the most important FSR parameters from an SA point of view when the distance is short ($\leq$ 2 hops). If, for example, we choose the

demands in [10], the demand of SA updates at least every second for nodes within a 3 km radius results in the following equation

$$\Delta_{demand}(2) = 1 \;\; \Rightarrow \delta(n+1) \leq 0.74$$

which effectively limits the values that can be used for n and δ.

The expression for delivery time to a node further away can be similarly expressed but does not simplify as well. An example of how it looks for 7 hops ($\geq$ 10.5 km) is given below

$$\begin{aligned} n \cdot \delta + 7 \cdot \Delta_{delay} + \delta \sum_{k=2}^{7} (k-1)^{\alpha} &\leq \Delta_{demand}(7) \\ \Rightarrow 2 \cdot \Delta_{delay} + \delta\big(n + 1 + 2^{\alpha} + 3^{\alpha} + \cdots + 6^{\alpha}\big) &\leq \Delta_{demand}(7). \end{aligned}$$

This shows how the FSR parameter α becomes increasingly important as the distance (the number of hops) increases.

From the equation (4.3) or (6.1) we can discover for which parameter settings the different SA update demands are fulfilled, see Table 6.1 for examples. It must be noted that if a parameter setting is valid from an SA point of view, it does not necessarily fulfill the accessibility demands presented in Chapter 6.1. These must be taken into account before choosing parameter settings.

The first example in Table 6.1 fulfills the demands given in [10]; in the second example the 3 km demand is the only change. We can see from this that the demand on updates within 3 km is very decisive in our network for the number of valid parameter settings that can be found. A valid parameter setting here is one that fulfills all demands made, i.e. for 3 to 30 km. The last example is based on the demands given in [1]. We can see that all these demands are fulfilled for parameter settings with low δ and α. A consequence of this is that we will yield low route lengths, but also a great amount of routing traffic.

Four examples of how the age of the SA information Δ_{SA}, i.e. the position inaccuracy, increases with the number of hops from the source of the SA information are shown in Figure 6.5 along with the demands from [10]. The square markers represent a parameter setting form Table 6.1, i.e. a setting that fulfills the demands. As we can see from the diamonds, a small change of δ results in a large change of packet delivery time for all distances. A change in n also effects nodes at all distances, as can be seen from the asterisks. Furthermore, a larger

	3 km	10 km	30 km	Valid $\{\delta, n, \alpha\}$
[10]	1 s	10 s	60 s	{0.1, 1-6, 0.0-1.4}
				{0.3, 1, 0.0-1.0 }
	2 s	10 s	60 s	{0.1, 1-6, 0.0-1.4 }
				{0.3, 1, 0.0-1.0 }
				{0.3, 2-4, 0.0-0.8 }
				{0.5, 1-2, 0.0-0.6 }
				{0.7, 1, 0.0-0.4 }
[1]	0.5 s	5 s	60 s	{0.1, 1, 0.0-1.4 }

Table 6.1: *Some examples of FSR parameter settings (right) that meets the SA demands (left).*

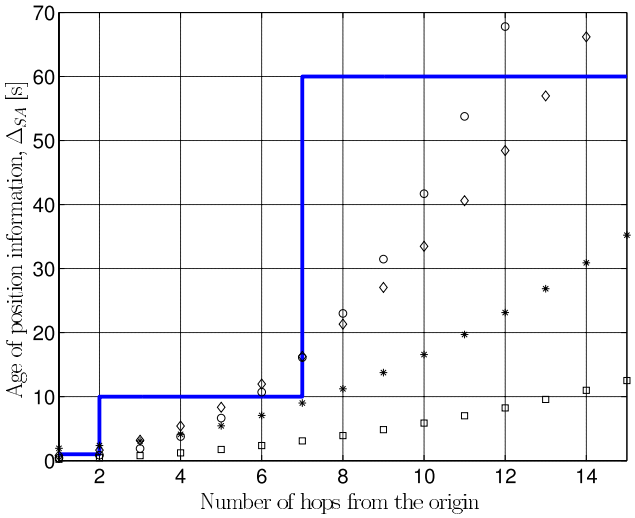


Figure 6.5: *Example of how the routing control packets delivery time increases with the number of hops for four different FSR parameter settings* $\{\delta, n, \alpha\}$*. The demand [10] is shown as a gray line. The squares represent a valid parameter setting* $\{0.1, 1, 1.0\}$*, the other markers represent settings where one parameter is out of bounds:* $\alpha = 1.6$ *(circles),* $\delta = 0.7$ *(diamonds), and* $n = 6$ *(asterisks) respectively.*

value of α results in a more exponential growth of delivery time which mainly has effects on nodes some distance away and their ability to meet the demands, see the circles. It should here be noted that the same effects results from much smaller changes than those shown in Figure 6.5, the settings here were chosen especially for illustrative purposes.

Chapter 7

Conclusions

In this report, we have studied the efficiency of the Fisheye technique in a highly mobile ad hoc network. We have also investigated the effect of badly chosen parameter settings compared to optimal parameter settings, and the possibilities of combining the SA service with the Fisheye State Routing algorithm.

From our simulations, we conclude that the Fisheye technique is efficient in the mobile ad hoc networks studied, since the maximum user capacity is achieved when the routing control traffic attenuation is prominent. We can also gather from the simulations that FSR will suffice in a highly mobile network.

A high fraction of the network capacity can be made available to the user with the right parameter settings, see Figure 6.4. The optimum, with respect to different network capacities, is achieved when the FSR parameter settings can adapt to the available capacity. In a network where it, for practical purposes, is necessary to use a fixed parameter setting, there can be heavy losses in user capacity if badly chosen parameter settings are used. Our results show that, if the total network capacity varies, it is crucial to choose the parameter settings based on the minimum network capacity predicted.

It can also be noted that in a real network, finding suitable parameter settings for all occasions will be very difficult for the user. It would hence be advantageous if the algorithm could automatically adapt its parameter settings to changes in the network environment. Also, by adapting the parameter settings, higher user capacity can be achieved.

We found that it is possible to combine the transmission of SA messages with the update messages used in the FSR algorithm. Parameter settings were

found that fulfilled the SA demands, but required frequent updates in the network. This implies good routing information in the network, but also a high amount of routing traffic. Thus the efficiency of this combination depends on the maximum user capacity in the network.

Chapter 8

Future Work

An interesting area for future work would be to study how FSR works in an environment with adaptive data rates, i.e. where the "cost" of using a link is a function of the transmission time. Furthermore, we want to compare the performance of FSR with that of a reactive routing algorithm, for example AODV [15], with and without the use of adaptive data rates.

Another issue is whether FSR performance can be improved by allowing the parameter settings to vary based on the behavior of the network. It would be interesting to let different nodes use different parameter settings in the network, depending on their traffic situation and the dynamics of their immediate environment. It would also be interesting to change certain functions in the FSR algorithm. The phase of initialization could be shortened, and the routing traffic that is transmitted in the network could also be divided into different classes to provide quality of service. Old and new routing information should be transmitted with different priorities through the network.

Appendix A

In this chapter we have collected the results that are not shown earlier but might interest the reader.

	$\phi = 95\%$			$\phi = 90\%$		
λ [Bytes]	α	n	δ	α	n	δ
10^4	1.6	2	1.3	1.2	1	0.9
10^5	1.6	1	1.3	1.4	1	0.7
10^6	1.4	1	0.3	1.0	1	0.3
10^7	1.4	1	0.1	0.8	1	0.1
10^8	0.4	1	0.1	0.2	1	0.1

Table A.1: *Optimum values of α, n, and δ for the different connectivities ϕ.*

	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$	$x=6$
$\alpha=0$	3.7	2.3	1.7	1.3	0.9	0.9
$\alpha=0.2$	3.7	2.3	1.5	1.1	0.9	0.9
$\alpha=0.4$	3.3	2.1	1.5	1.1	0.9	0.7
$\alpha=0.6$	2.9	1.9	1.5	1.1	0.9	0.7
$\alpha=0.8$	2.7	1.7	1.3	1.1	0.9	0.7
$\alpha=1.0$	2.3	1.7	1.3	0.9	0.9	0.7
$\alpha=1.2$	2.1	1.5	1.1	0.9	0.7	0.7
$\alpha=1.4$	1.7	1.3	1.1	0.9	0.7	0.7
$\alpha=1.6$	1.7	1.3	0.9	0.7	0.7	0.5
$\alpha=1.8$	1.5	1.1	0.9	0.7	0.7	0.5
$\alpha=2.0$	1.1	0.9	0.7	0.7	0.5	0.5
$\alpha=2.2$	0.9	0.7	0.7	0.5	0.5	0.5
$\alpha=2.4$	0.9	0.7	0.5	0.5	0.5	0.3
$\alpha=2.6$	0.7	0.5	0.5	0.5	0.3	0.3
$\alpha=2.8$	0.5	0.5	0.5	0.3	0.3	0.3
$\alpha=3.0$	0.5	0.3	0.3	0.3	0.3	0.3

Table A.2: *Maximum δ [sec] values that fulfill the requirement of 95% accessability when the simulated network is 95% connected.*

	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$	$x=6$
$\alpha=0.0$	1.9	1.3	0.9	0.7	0.5	0.5
$\alpha=0.2$	1.9	1.1	0.9	0.7	0.5	0.5
$\alpha=0.4$	1.5	1.1	0.7	0.7	0.5	0.5
$\alpha=0.6$	1.3	0.9	0.7	0.5	0.5	0.3
$\alpha=0.8$	1.1	0.9	0.7	0.5	0.5	0.3
$\alpha=1.0$	0.9	0.7	0.5	0.5	0.3	0.3
$\alpha=1.2$	0.9	0.7	0.5	0.5	0.3	0.3
$\alpha=1.4$	0.7	0.5	0.5	0.3	0.3	0.3
$\alpha=1.6$	0.5	0.5	0.3	0.3	0.3	0.3
$\alpha=1.8$	0.5	0.3	0.3	0.3	0.3	0.1
$\alpha=2.0$	0.3	0.3	0.3	0.3	0.1	0.1
$\alpha=2.2$	0.3	0.3	0.1	0.1	0.1	0.1
$\alpha=2.4$	0.1	0.1	0.1	0.1	0.1	0.1
$\alpha=2.6$	0.1	0.1	0.1	0.1	0.1	0.1
$\alpha=2.8$	0.1	0.1	0.1	0.1	0.1	0.1
$\alpha=3.0$	0.1	0.1	0.1	0.1	0.1	0.1

Table A.3: *Maximum δ [sec] values that fulfill the requirement of 95% accessability for a network with 90% connectivity.*

	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$	$x=6$
$\alpha=0$	2.1	1.8	1.7	1.6	1.9	1.6
$\alpha=0.2$	2.0	1.8	1.9	1.9	1.9	1.6
$\alpha=0.4$	1.5	1.8	1.8	1.9	1.9	2.0
$\alpha=0.6$	1.4	1.8	1.7	1.9	1.9	2.0
$\alpha=0.8$	1.3	1.7	1.7	1.7	1.8	1.9
$\alpha=1.0$	1.3	1.4	1.5	1.9	1.6	1.8
$\alpha=1.2$	1.2	1.4	1.5	1.7	1.9	1.6
$\alpha=1.4$	1.3	1.3	1.3	1.4	1.6	1.4
$\alpha=1.6$	**1.2**	**1.3**	1.5	1.6	1.4	1.9
$\alpha=1.8$	1.3	1.2	1.3	1.4	**1.3**	1.6
$\alpha=2.0$	1.5	1.3	1.4	1.3	1.6	1.4
$\alpha=2.2$	1.7	1.5	**1.2**	1.5	1.4	**1.3**
$\alpha=2.4$	1.7	1.4	1.6	1.4	1.3	1.9
$\alpha=2.6$	2.0	1.8	1.4	**1.2**	1.9	1.8
$\alpha=2.8$	2.7	1.7	1.3	1.9	1.7	1.6
$\alpha=3.0$	2.6	2.7	2.0	1.7	1.5	1.4

Table A.4: *Routing Traffic, [10^4 Bytes], generated in the total network during the simulation when the network is 95% connected. Minima are shown in bold text.*

	$x=1$	$x=2$	$x=3$	$x=4$	$x=5$	$x=6$
$\alpha=0$	0.33	0.26	0.26	0.25	0.28	0.23
$\alpha=0.2$	0.31	0.31	0.25	0.25	0.28	0.23
$\alpha=0.4$	0.27	0.28	0.32	0.24	0.27	**0.22**
$\alpha=0.6$	0.24	0.30	0.29	0.33	0.27	0.38
$\alpha=0.8$	0.25	0.25	0.25	0.30	0.25	0.37
$\alpha=1.0$	0.25	0.27	0.31	0.26	0.38	0.34
$\alpha=1.2$	**0.22**	**0.23**	0.26	0.23	0.34	0.30
$\alpha=1.4$	0.24	0.25	**0.22**	0.32	0.29	0.26
$\alpha=1.6$	0.31	0.23	0.34	0.28	0.26	0.24
$\alpha=1.8$	0.29	0.36	0.30	0.25	**0.22**	0.63
$\alpha=2.0$	0.42	0.30	0.25	**0.22**	0.60	0.54
$\alpha=2.2$	0.39	0.27	0.69	0.61	0.54	0.48
$\alpha=2.4$	1.1	0.83	0.64	0.56	0.50	0.43
$\alpha=2.6$	1.0	0.75	0.57	0.49	0.44	0.40
$\alpha=2.8$	0.99	0.69	0.52	0.44	0.39	0.36
$\alpha=3.0$	0.96	0.65	0.48	0.40	0.36	0.33

Table A.5: *Routing Traffic, [10^4 Bytes], generated in the total network during the simulation when the network is 90% connected. Minima of each row are shown in bold text.*

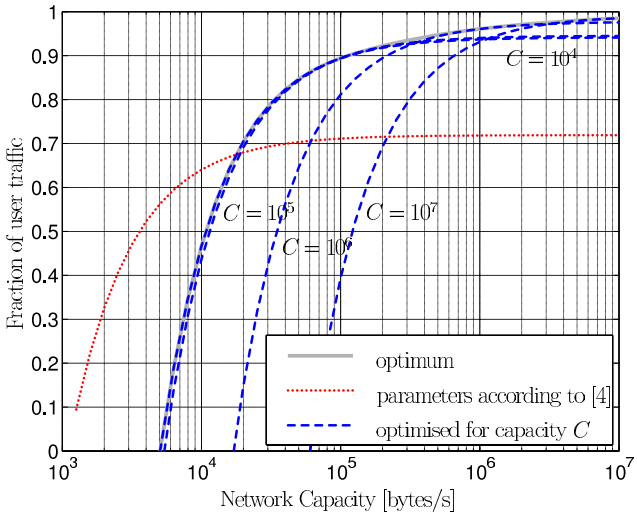


Figure A.1: *Fraction of the network capacity available for user traffic in the network with 90% connectivity* .

Bibliography

[1] M. Sköld, "Senariobaserad utvärdering av positionsförmedling i en mekaniserad bataljon," FOI memo Dr. nr. 01-4233, Command and Control Systems, FOI, Swedish Defence Research Agency, dec 2001.

[2] K. Persson, "Quality of Service Routing in Tactical Mobile Ad Hoc Networks," Tech. Rep. FOI-R–0886–SE, Div. of Command and Control Systems, FOI, Swedish Defence Research Agency, June 2003, In swedish.

[3] G. Pei, M. Gerla, and T-W. Chen, "Fisheye State Routing in Mobile Ad Hoc Networks," *Workshop on Wireless Networks and Mobile Computing*, pp. D71–D78, 2000.

[4] M. Gerla, X. Hong, and G. Pei, "Fisheye State Routing Protocol (FSR) for Ad Hoc Networks," IETF Internet Draft (work in progress), June 2002, draft-ietf-manet-fsr-03.txt.

[5] A. C. Sun, "Design and Implementation of Fisheye Routing Protocol for Mobile Ad Hoc Networks," Master's thesis, Massachusetts Institute of Technology, May 2000.

[6] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," in *ACM MOBICOM '98*, October 1998.

[7] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," in *Proceedings of INFOCOM 2000 Conference*, March 2000.

[8] B. Johansson F. Eklöf, "Positionsförmedlingstjänst för mekaniserade förband," Tech. Rep. FOA-R–00-01734-504–SE, Div. of Command and Control Systems Warfare Technology, FOI, Swedish Defence Research Agency, dec 2000, In swedish.

[9] U. Sterner M. Sköld, "Evaluation of Tactical Radio Networks - Simulation method and results," FOI memo 01-1486/L, Command and Control Systems, FOI, Swedish Defence Research Agency, jun 2001.

[10] Markstridsskolan Försvarsmakten, "PTTEM StridsLedningssystem Bataljon," Dec 2000, Bilaga till MSS 09 621.

[11] I. Chlamtac and S. Pinter, "Distributed nodes organization algorithm for channel access in a multihop dynamic radio network," *IEEE Trans. Comput.*, vol. 36, no. 6, June 1987.

[12] I. Cidon and M. Sidi, "Distributed assignment algorithms for multihop packet radio networks," *IEEE Trans. Comput.*, vol. 38, pp. 1353–1361, oct 1989.

[13] R. Nelson and L. Kleinrock, "Spatial-TDMA: A Collision-Free Multihop Channel Access Protocol," *IEEE Trans. Comput.*, vol. 33, no. 9, pp. 934–944, Sept. 1985.

[14] C. David Young, "USAP: A unifing dynamic multichannel TDMA slot assignment protocol," in *Proc. IEEE MILCOM'1996*, 1996.

[15] C. Perkins, E. M. Royer, and S. R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF Internet Draft (work in progress), feb 2003, draft-ietf-manet-aodv-13.txt.