# Content-based routing in publish/subscribe systems with dynamic clients behaviour

## A general model applied to military networks

**Folkert van Vliet**

Master's thesis
June 2007

THALES

Thales Land & Joint Systems BV
Department of System Engineering
Huizen

University of Twente
Enschede - The Netherlands

University of Twente
Department of Applied Mathematics
Enschede

# Content-based routing in publish/subscribe systems with dynamic clients behaviour

## A general model applied to military networks

**Supervising committee**
Prof. Dr. R.J. Boucherie, University of Twente
Dr. Ir. M. de Graaf, Thales Land & Joint Systems BV
Dr. J.L. Hurink, University of Twente
Dr. J.C.W. van Ommeren, University of Twente

**Student number**
0008982

THALES

University of Twente
Enschede - The Netherlands

# Abstract

One of the main problems with information distribution in military radio-networks is that available bandwidth is extremely low. A second problem is that due to movement of clients using these radios, connections are lost so now and then. The publish/subscribe system seems to be interesting to use in the radio-network as these provide a decoupling of clients and asynchronous interaction. Moreover, routing decisions are based on the content of messages which makes it possible to reduce the network usage. Within Thales content-based routing is regarded as an interesting development that may solve the problem as it works more efficiently than traditional addressing-based algorithms. The research in this thesis is concerned with these content-based routing algorithms.

More precisely, we will compare the efficiency of two content-based routing algorithms: a flooding scheme and a more sophisticated routing scheme called identity-based routing. By flooding, the information sent by publishers, is spread in the entire network and locally is determined if it should be delivered to an interested subscriber on this information. This causes a lot of *data-traffic*. By identity-based routing, the interests of subscribers is spread into the network and at the moment the publisher sends information this is filtered through the entire network. This causes less data-traffic when compared with flooding, however with an additional price of *control-traffic* for the dissemination of clients interest through the network. From this, the central question of this thesis is: which of them performs better when the total amount of messages that will flow through the network has to be minimised? In environments where only a few clients are interested in some specific information it is intuitively clear that identity-based routing is superior and in situations where a lot of clients are interested in the information flooding seems to be superior. However, the environment (e.g. the size of the network and the behaviour and number of clients contributing the network) can be different for any real networked situation and thus a different routing scheme can be preferable. Not only for radio-networks, but for all kind of networks. Concerning this aspect, it is preferred to gain more accurate insight in which environment which scheme is preferable. Therefore, the research in this thesis is focused on an analytical approach by modeling both routing schemes from which we are able to compare them in a various range of possible environments.

At first we derived a model by considering a static situation and simplified assumptions. Next, we extended the model by considering a dynamic situation where the subscribers behaviour is modeled by using queueing systems: they arrive according a Poisson process, stay in the system for a while and are active as subscriber before leaving again. Furthermore we assumed a self-stabilising solution for recovering from temporarily failures like connection break downs. This is done by (re-)sending messages periodically and, on account of subscriptions, re-new the old subscriptions each refresh-period. The basis of our model is derived from the work by Jaeger and Mühl [6]. They used a steady state approach, from which they calculated the amount of traffic originating in a fraction of the time, on base of the activity of subscribers and compared it with flooding. We extended this basic model in many ways to compare flooding with identity-based routing in more different environments. Among these, a transient state approach were subscribers behaviour is possible to change in time, different arrival and departure processes at each location in the network, more general networks and positions of subscribers and publishers, the use of advertisements and ideas for covering-based routing and roaming clients.

The most important parameters of the model include the arrival rate of subscribers $\lambda$, the average holding time of an active subscriber $1/\mu$, the rate at which publishers send their information $\omega$, the re-subscription period $r$ and parameters for defining the network and number of possible subscribers. We applied the model in a reference model for a military radio-network and in the CAFES-environment. CAFES is a system that supports Voice over IP (VoIP) communication between clients over an IP (Internet Protocol) based network.

In an exemplary setting of the reference model of the radio-network, we considered the dispersion of situational awareness information and found that the break-even-point for the activity of subscribers lies around 12%. When more than 12% of the subscribers are active, flooding is preferred, when less than 12%, identity-based routing is preferred. By varying the parameters, $\omega$, $\mu$, $r$ and the distribution of the interest of subscribers (i.e. locally distinct interest than farther away from the publisher) this break-even-point will become higher when: the notification $\omega$ rate is higher, the period of being active as subscriber becomes longer (so $\mu$ lower), the re-subscription periods $r$ becomes longer and when locally the interest is higher than further away in the network. In the CAFES environment, we found that only when publishers are quite active and subscriber very inactive flooding (of notifications) would *not* be better. So, in all other situations flooding of notifications is preferred. When considering the presence information that is proposed to use in the CAFES-environment, flooding of notifications will be best as it is assumed that the activity of clients is rather higher than quite low.

By using the model we derived, we are able to answer the research question in various environments. However, there are more extensions possible so even more environments can be considered. For further research, the model we derived can be used as a basic model.

# Preface

This project is performed as the final graduation project of my studies at the Stochastic Operation Research chair at the Applied Mathematics department of the Faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente. It is performed at Thales Land & Joint Systems BV where they became interested in an analytical approach on the performance of content-based routing in military applications. I was situated at Thales in Huizen from the beginning of November 2006 until the end of April 2007.

For succeeding this project I have to thank many people who encouraged me and advised me. Special thanks go to Richard Boucherie. As my mentor during this project and during my study, his enthusiasm, guidance and encouragements made me complete the project and made me proud on the final results. Furthermore, I am gladly to thank Maurits de Graaf who coached me at Thales during the project. Also his advices and enthusiasm for the project were a great help for me. I also want to thank Thales Land & Joint Systems for the possibility and facilities they reached me to perform this research.

Furthermore I want to thank all people who contributed to this report in any way. Among them my colleagues at Thales, whom I always could ask questions. Moreover, I want to thank my family and friends. Despite my complaints and frustrations they were always there to encourage me. I am proud on having such reliable relatives and I appreciated their support a lot!

Folkert van Vliet
June 2007

# Contents

# List of Figures

# Chapter 1

# Introduction

In telecommunication, client/server models became a standard for communication between interconnected communication devices (e.g. computers). In this model, the communication between clients and servers occurs by requests and responses. A client who is interested in information from the server sends a message (through the communication network) with a request towards the server. The receiving server responds to the clients request by sending the requested information. This model became widely used, mainly because of its simplicity and familiarity.

However, there are several disadvantages. One of the main disadvantages is the tight coupling among the involved clients and servers. A client needs to explicitly address the server and the server must be ready to process the request from the client. These models are relevant for real-time communication, such as a voice conversation. Nevertheless, due to this synchronously exchanging of messages and explicitly addressing, especially information-driven applications (such as news delivery, stock quoting or air traffic control) cannot be realized efficiently: the model becomes less scalable in a more dynamical environment.

To overcome the drawbacks of the client/server model in a highly dynamical environment, a message-orientated middleware paradigm called 'publish/subscribe system' emerged in the 1980s. In this system, the middleware is an application based system which controls the delivering of clients published information (by publishers) towards interested clients (subscribers). The middleware can be represented as an operating application *between* the clients. In this design, the drawbacks of the client/server model are neutralised: clients are loosely coupled (they do not have to explicitly address each other) and messages can be passed asynchronously.

The work in this thesis is focused on the publish/subscribe paradigm. Therefore, most of the important aspects and terminology concerning publish/subscribe system are described in section 1.1 first. Next, in section 1.2, is described on which aspects the research in this thesis is focused and defines the research objectives. In section 1.3 a literature review is given in the light of the contribution of this thesis, followed by the scope of the research in section 1.4. Finally in section 1.5 the structure of the remainder of this thesis is given.

## 1.1 Publish/subscribe systems

A publish/subscribe system consists of a set of clients that asynchronously exchange messages over the network. The clients can be producers or consumers and the network between the clients acts as a message forwarding- and delivering service. The producers are responsible for the input of information by publishing 'notification messages' (or short *notifications*) into the network. The consumers subscribe to information they are interested in by issuing 'subscription messages' (or short *subscriptions*). A published notification is created in reaction to an 'event' (e.g. when new information is available). According to this terminology, the forwarding- and delivering-service of messages is called 'event notification service' (or short *event service*). The event service is in fact one application built on top of the underlying network which connects the clients. In figure 1.1 the interaction of clients with the event service on application level is shown.

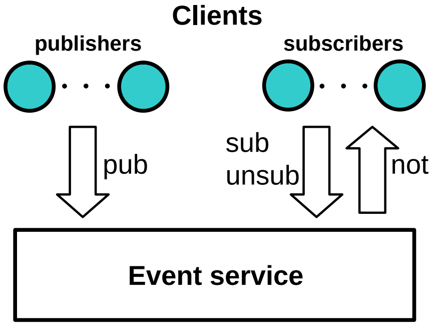


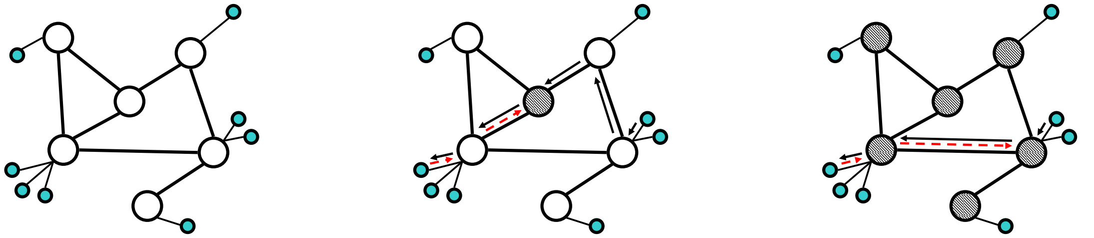
Figure 1.1: Blackbox view of a publish/subscribe system

Consistent with the terminology, from now on we will call the clients acting as producers of information simply *publishers* and clients acting as consumers of information *susbcribers*. As can be seen in the figure we distinguish between four types of interaction between the clients and event service, namely publish (pub), subscribe (sub), unsubscribe (unsub) and notify (not):

- **Publish**. The action which is performed by the publisher at the moment he sends a notification containing information towards the event service.
- **Subscribe**. The action which is performed by the subscriber at the moment he likes to receive information by sending a subscription on the subject of this information towards the event service.
- **Unsubscribe**. This action is performed by the subscriber at the moment he does not want to receive information on a specific subject anymore. This means that an unsubscription is sent towards the event service in order to remove the previously announced interests.
- **Notify**. At the moment the event service receives a notification from a publisher, it checks which subscribers need to receive this notification. The action performed by the event service at the moment it actually delivers the notification to the interesting subscriber is called notifying.

Physically the clients (represented as devices with running applications) are connected to a network consisting of routers and interconnected links. On this level it is possible that a message has to travel several hops to reach a desired location. The event service is designed in the application layer on top of this underlying network. This means that the applications running at the routers together form the event service. So we do not have to be concerned with the way how messages are physically traveling from source to destination, but need to be concerned with how communication between the routers should be implemented in their applications so they together form a properly working event service. In this context, we distinguish between two possible implementations. Namely a centralised and a distributed solution:

**Centralised event service**. Figure 1.2(a) shows an example of a physical network, where a large node represents a router (or 'broker'), a small node represents a client and the lines represents the links interconnecting the clients and brokers. Figure 1.2(b) gives an example for a centralised solution for the event service. Here one broker (mostly in the center of the network) holds a large database in which all subscriptions are saved. So, if a client sends a subscription, this message will always be propagated towards the central broker and saved in the database. In the figure an example of a forwarded subscription is given by the dashed arrows. Also published notifications are directly forwarded towards the central broker. Now, by consulting the saved subscriptions in its database, the application at the broker is able to determine in what directions received notifications have to be forwarded, so they can be delivered to the interested subscribers. In our example, we see the path that a published notification which matches the subscription travels, by passing the central broker (the solid arrows).



(a) Exemplary network (b) Centralised event service (c) Distributed event service

Figure 1.2: Different solutions for the event service

**Distributed event service**. In a distributed system, all brokers (or a part of all brokers) together form the event service. So here all contributing brokers contain a (small) database. The brokers interact with each other by sending messages based on the definitions in their database. In publish/subscribe terminology this database is called the *routing table*. At each broker the routing table consists of a list of routing entries, which are based on the currently attending subscriptions in the network. Figure 1.2(c) shows an example of this solution. We see that the paths of the forwarded subscription and published notification differ from the centralised solution.

The advantage of a centralised system is the ease of simply forwarding the messages towards one point in the network. However, this advantage is degrading when the disadvantages are compared with the distributed system. The database should be very large in a large scaled network and even worse, the single broker acts as a single point of failure: if it collapses, the whole system is down. Besides, unnecessary use of links can be avoided when messages do not have to travel to the central broker first (as can be seen in the example). Therefore we focus on a distributed publish/subscribe system.

A properly working distributed publish/subscribe system in a dynamic environment has to fulfill many requirements. The main requirement is that different applications used by the clients are able to exchange information with each other by using the overall event service. In order to fulfill this requirement we need a consistent language for all applications in the publish/subscribe system and agreements how information is exchanged. The definition of the language of the messages and agreements how algorithms should read the messages is defined by the *data/filtermodel*. Agreements how messages are forwarded and delivered is defined by the *routing algorithm*. In paragraphs 1.1.2 and 1.1.3 the most important background information about these models and algorithms are given. But before turning to this, we first explain two important terms expressing the quality and performance of a distributed publish/subscribe system: expressiveness and scalability.

### 1.1.1 Expressiveness and scalability

The quality of a distributed publish/subscribe systems can be expressed in terms of the *expressiveness* and *scalability*. The expressiveness refers to the power of the language used in the data/filtermodel. A high expressiveness is reached when as little as possible unnecessary information is propagated through the network. So the more powerful the data/filtermodel, the higher the expressiveness.

Scalability is a measure for the environment in which a publish/subscribe system works properly. It depends on parameters like the size of networks and number of devices connected to it and is determined according to network performances like the continuity or reliability of the connection and the latency. We say a system is highly scalable if it performs well in a large networked environment with many devices and interaction.

The relation between expressiveness and scalability is opposite: the higher the expressiveness, the harder it is to realize properly working implementations on large scale. However, both these performances have to be optimized to fulfill the requirements of a flexible publish/subscribe system.

### 1.1.2 Data/filtermodel

The datamodel defines the construction of messages based on a programming language. The filtermodel consists of the method how notifications are matched by using the structure of the data model.

A notification is represented as a set of pre-defined attributes. A common method is to declare an attribute as a type with a name and a value, in which the types correspond with the types (e.g. integer, string, float) used in programming languages (XML, Java-object orientated). Figure 1.3 shows an example of a notification representing a stock price change as it is used in the SIENA system by Carzaniga, Rosenblum and Wolf [3]. The subscriptions are in essence a kind of filters. A filter expresses the interests of the subscriber in terms of constraints on the values of the attributes of information. If, for example, a client is interested in the price of the stock DIS when it has a value larger than 110 dollars, the subscription will look like {finance/exchanges/stock/NYSE/(DIS $\geq$ 110)}. The notification containing the value of the stock, which is frequently published by a publisher, will thus only be delivered to this subscriber if the value is larger than 110 dollars. From this design a filter on the information of the notification is also called a filter on the *content* of a message.

| | | |
|---|---|---|
| string | class | finance/exchanges/stock |
| time | date | Mar 4 11:43:37 MST 1998 |
| string | exchange | NYSE |
| string | symbol | DIS |
| float | prior | 105.25 |
| float | change | -4 |
| float | earn | 2.05 |

Figure 1.3: Example of a notification

### 1.1.3 Content-based routing algorithms

Another important aspect for a properly working distributed publish/subscribe system is the routing mechanism for forwarding and delivering of messages in the network. There are several algorithms designed for this purpose and consistent with the definition that filters are expressed in terms of the content of messages, these algorithms are called *content-based routing* algorithms.

The idea of the algorithms is that a new arriving subscription (at first received by local brokers) is propagated through the network so the brokers in the network 'know' about the subscription. As a result, in the future sent notifications which match the filter of the subscription are directed on the path that is established by the brokers towards the local broker, and finally the subscriber. This idea is implemented in algorithms which are invoked each time a message arrives at a broker. Naturally, as we are concerned with subscriptions and notifications, we distinguish between the following two actions which trigger the algorithms:

**Arrival of subscription**. Each broker receiving a subscription saves the received filter of this subscription as a routing entry in its routing table and forwards the subscription towards his neighbour brokers. The routing entry consists of the filter and the address of the broker the subscription is received from. The neighbour brokers receiving the subscription run through the same process.

**Arrival of notification**. At the moment a broker receives a notification, it consults the routing table to check whether it matches one of its filters saved in the routing entries. The notification is forwarded to those brokers for which the filter matches.

In this research we are interested in the amount of messages (which is the sum of all subscriptions and notifications) flowing through the network. In addition, we will call the amount of messages in the network caused by the forwarding of notifications *data-traffic* and the amount of messages in the network caused by subscriptions *control-traffic*.

Concerning the different algorithms, we distinguish between two extremes with regard to the distinct message traffic:

- **Flooding**. A subscribers interest is only saved locally (i.e., in the routing table of the local broker) and will not be forwarded into the network. At the moment a publisher sends a notification to its local broker, this notification is forwarded towards all brokers in the network. In other words, the notification is *flooded* through the network. At the moment the notification is received by a local broker, the process of determining to which local subscribers the notification has to be forwarded is performed. From this point of view we can say that the filtering of notifications is performed at the edge of the network.
- **Sophisticated filtering**. Contrarily we have the situation where the subscriptions received by the local brokers are forwarded into the network in such way they establish the future path (or more paths if there are more publishers containing the same information) the published notifications have to travel towards the interested subscriber. I.e., the filtering of notifications is more sophisticated compared to flooding and performed through the entire network.

Considering the message traffic caused by flooding in the network, we observe that there will be no control-traffic (as subscriptions are not forwarded), but a lot of data-traffic. Considering the message traffic caused by a more sophisticated filtering, we observe that there usually will be less data-traffic in comparison with flooding, however with an additional price of control-traffic. Using these algorithms in a distributed publish/subscribe system, intuitively, the more subscribers are interested in information, the more data- and control-traffic will flow through these networks. In this situation it is possible that the total traffic due to the extra costs for control-traffic will become larger than in a situation where flooding, merely generating data-traffic, is implemented.

## 1.2 Research

An important aspect when considering the implementation of a publish/subscribe system is in what (dynamical) environment it should be realised. The environment comprises elements as the number of devices that are used, how these are interconnected (i.e., what is the network topology?) and the behaviour and number of clients using the devices. If the environment is known a decision should be made on which routing mechanism to use. In previous section we described the flooding and more sophisticated filtering mechanisms. In environments where only a few clients are interested in some specific information it is intuitively clear that more sophisticated filtering is superior and in situations where a lot of clients are interested in the information flooding seems to be superior. From this intuition we are interested in a more accurate determination which of the schemes performs best by considering as performance indicator the total amount of messages in the network. The research will focus on this aspect.

In the following paragraph we formulate the research objectives. Next, in paragraph 1.2.2, we describe our approach how we performed the research.

### 1.2.1 Research objectives

The contribution of this thesis will focus on this aspect. In more detail, we will derive a model in which we are able to answer the following research question:

> **In what environments is a more sophisticated filtering algorithm superior to a simple flooding algorithm when total traffic is minimised?**

This question is difficult to answer, because the efficiency of the routing mechanisms depends on many parameters. Answers to this question can be of great support, since on base of better insight in which environments which of the routing mechanisms is preferable, better decisions can be made when considering to implement this system.

The research will focus on modeling the more sophisticated filtering algorithms and flooding principle. This model will be applied in the environment of radio-networks and CAFES-networks. Thales is namely interested in the possibility of the implementation of a publish/subscribe system in these environments. The radio-network is applied in the land force and has to fulfill data-communication between the participants of this network (i.e., army units). Typical information that is proposed to sent through the network includes situational awareness information, such as the position information of army units. As the available bandwidth in a radio-network is extremely low, answers to the research question for this type of information will be very useful. CAFES is a system that supports Voice over IP (VoIP) communication between clients over an IP (Internet Protocol) based network. In this system, presence information is used to indicate the status of a client (e.g. available, busy, away, etc.). For this type of information we will apply the model and derive conclusions in what environment one of the algorithms performs best when total traffic is minimised.

### 1.2.2 Approach

The approach how the research is performed in order to answer the research question and fulfill the research objectives, is described here. At first a literature study is performed to gain insight in the working and possibilities of applying publish/subscribe systems and content-based routing. Moreover, from this study we were able to find the relative work in the area of this research project. The results of this literature study are described in section 1.3.

The approach for answering the research question is by the derivation of an analytical model. From the literature study and considering the applications at Thales (i.e. the radio and CAFES environment) we defined a scope for the range of modeling parameters. These are described in section 1.4. Within this range, we built the model. At first we started to derive a model for a static situation with simplified assumptions. Next, we extend most of these assumptions and derive models for more dynamic situations, where clients behaviour is modeled by means of queueing theory.

Finally we applied the model on the radio-network and CAFES-network and derived results by considering exemplary settings.

## 1.3 Literature review

For a clear understanding of the publish/subscribe paradigm a literature study is performed. We focused on all ins and outs in general (paragraph 1.3.1) and refined our search on the performance of the (content-based) routing schemes used in publish/subscribe systems, which is described in paragraph 1.3.2.

### 1.3.1 Development of publish/subscribe systems

A lot of research is focused on the design of a properly working publish/subscribe system in practice. Several research groups designed there own prototypes, created in their own environment and networked applications. Some example prototypes are the SIENA system (Carzaniga, Rosenblum and Wolf [3]), the ELVIN system (Segall and Arnold [12]) and the REBECA system (Mühl [9]). During the development of these systems a lot of different routing schemes and expressions for data/filter models are proposed. In addition, the work by Mühl in 2002 consists of an extensive overview of all components in a publish/subscribe system. He presents a formal specification for correct working publish/subscribe systems and evaluates many different routing schemes and data/filter models on base of its scalability and expressiveness. Also Eugster et all. [4] give an overview of many variants of the publish/subscribe system. They identify the differences and similarities by comparing them on base of their performance in decoupling in time, space and synchronisation[1].

A common factor concerning the prototypes described by Mühl and Eugster et al. is that they are designed for most static environments, i.e., systems with a fixed infrastructure. More recent work is focused on supporting mobility in publish/subscribe systems. For example, Fiege et all. [5] propose a scheme which supports physical mobility and logical mobility in the REBECA system. Here physical mobility refers to a roaming client traveling through the area and needs to change accesses to different base stations to stay connected. Logical mobility refers to a changing interest in different areas of a client acting as subscriber (e.g., when a client enters a specific room, it is notified about the temperature in that room according to its subscription on that information).

### 1.3.2 Evaluating content-based routing algorithms

Many of the content-based routing algorithms are evaluated on their performance by simulating studies and working prototypes. Two common performances that are measured include the *sizes of the routing table* and the *amount of messages* flowing through the network. The size of the routing table indicates the computational complexity that is needed to evaluate where received notifications have to be forwarded to. The larger the routing table, the 'slower' the system. In general, the more advanced the routing algorithm is, the less unnecessary subscriptions are forwarded which results in smaller sizes of the routing tables. The amount of messages flowing through the network include all notifications and subscriptions traveling over the links. Naturally, the lower this amount, the better the performance of the system. We are focusing on this last performance measure, because using this measure we are able to compare flooding with more sophisticated filtering schemes.

Mühl et al. [10] evaluated several routing algorithms in a working prototype on base of both performance measurements described above. By taking into account the amount of messages flowing through the network, their conclusion about a flooding mechanism compared with more sophisticated filtering algorithms is that the use of sophisticated filtering algorithms is advantageous in more dynamic scenarios than was previously thought. In their experiment they used a hierarchical network of brokers with one central publisher and subscribers uniformly distributed over the brokers. By varying the number of active subscriptions, the

---

[1] Time, space and synchronisation are three different indicators how clients are decoupled from each other. Decoupling in time means that clients do not have to be connected to the event service at the same moment in time. Decoupling in space means that clients do not know which other clients are participating the event service (e.g. a subscriber does not know who publishes the information he wants to receive). Decoupling in synchronisation means that there is no explicitly addressing and simultaneous interaction.

performance was evaluated. A logical outcome is that with small numbers of active subscriptions the sophisticated filtering algorithms are much better than the performance of flooding. In addition, they observed that locality among the interests of subscribers (i.e., subscribers close together issue the same subscriptions) even improves the sophisticated filtering algorithms when compared with flooding.

Instead of using simulations and working prototypes, we are interested in an analytical approach for measuring the performance. There is not many written in this scene, mainly because of the difficulties by defining the parameters of the dynamic environment for a properly working publish/subscribe system. Baldoni et al. [2] present a formal framework for modeling content-based routing. Just as in Mühl [9] their approach is based on the correctness of the system when it is evolving in time. A system is correct if it is fault tolerant (i.e. is able to recover faults in a certain time) and ensures that the right notifications will only be delivered at the interested clients as much as possible. In this light they built a model which calculates the notification loss by considering the publish/subscribe system as a blackbox system. They defined functions for the dissemination of messages in a network following epidemic like behaviour and parameters include the average duration subscriptions stay in the system and the time notifications match a filter. By using this model a designer of a publish/subscribe system is able to predict the behaviour of their application considering fault tolerance.

However, a comparison between flooding and sophisticated filtering algorithms can not be derived from this model. Another approach for modeling a fault tolerant publish/subscribe system is proposed by Mühl et al. [11]. They define a self-stabilising algorithm that is responsible for frequently renewing all messages flowing through the publish/subscribe system and remove old messages. By this design, the system will automatically recover faults and is implementable in highly dynamical environments. In a simulation study they compare the self-stabilising filtering algorithm with flooding and conclude that the filtering algorithm is superior to flooding for a large range of practical settings.

In contribution, Jaeger and Mühl [6] represent an analytical approach to analyze the system. Their work is based on a stochastic approach. In a hierarchical tree-based network, they assumed subscribers at the bottom of the network and one publisher at the top of the network. They used queueing systems at the end of the network to model the behaviour of subscribers. They assumed similar behaviour for all participants and calculated the traffic in the steady state. Their work seems to be a good approach for the model we are intending to create. Therefore we will use their approach as a basic for our model and extend it in many ways. For example, we will assume distinct behaviour at each point in the network and allow all kind of tree networks in which the model is applicable. Moreover we will consider a more transient behaviour of clients and roaming clients. In paragraph 2.3.1 of this report, the basic idea for their approach is described.

## 1.4 Scope

This section describes the scope of the research. As there are many decision parameters possible for a publish/subscribe system in a dynamic environment, we need some well-defined assumptions which we use for modeling a realisation. These assumptions are described and we will argue why these choices are realistic. Considering the publish/subscribe paradigm we can roughly division the decision parameters into three parts. Namely, the design of the network of distributed brokers who form the event service and the distribution of the clients (i.e., the network topology), the content-based routing algorithms and the dynamic aspects we take into account. These are described in resp. paragraph 1.4.1, 1.4.2 and 1.4.3.

### 1.4.1 Network topology

As already explained in section 1.1 we consider a distributed publish/subscribe system, consisting of a set of brokers who together form the event service. As the event service is designed in the application layer this set of brokers and links between the brokers (the 'overlay network') can be represented as a virtual network designed above the underlying physical network. It is a kind of mapping where nodes in the overlay network match with physical nodes (routers) and a virtual link between two nodes matches with paths of several physical links between the two nodes. Figure 1.4 shows a representation of an overlay network above a physical network. In this figure we see that some of the clients that are connected in the physical network to a physical router, are assigned to a different broker in the overlay network. This is as not all routers do contain the application of the event service. From now on, we refer to the term network we intend the overlay network consisting of the brokers which forms the event service. Moreover, if we will call brokers where clients are directly attached with *local brokers* and brokers without clients *routing brokers*.

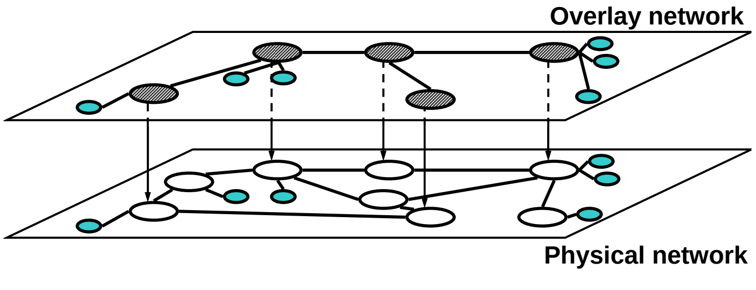


Figure 1.4: Representation of an overlay network

The purpose of our research is to model the amount of messages flowing over the links of the network as a measure for the network load. Therefore, the topology is an important issue. In our modeling approach we will use a tree-shaped network. We take this assumption, because due to this structure paths between brokers are unique, so it is much easier to define the calculation for messages passing these paths in our model. The choice for using this tree-shaped network is grounded as any network can be reduced to a tree when considering routing and many networks in real world are based on a hierarchical structure (like for example the Internet).

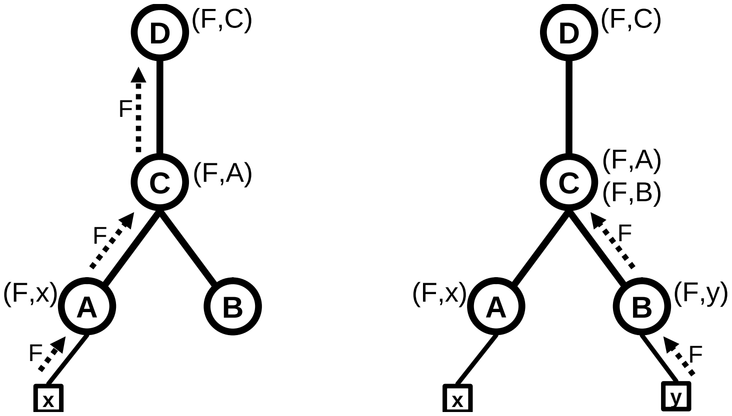
During the first steps of our modeling we will use a more regular hierarchical tree-shaped network structure (an $m$-ary tree with $k$ levels, which will be explained at the beginning of Chapter 2). We assume one publisher in the root of this tree and various subscribers at the end-leafs. This structure facilitates calculations even more. Later on we will propose a more extended approach by considering more general tree networks where publishers and subscribers have the possibility to connect the system on any location.

### 1.4.2 Content-based routing algorithms

The simplest content-based routing algorithm is flooding. Among the more sophisticated filtering algorithms of content-based routing we distinguish between *simple*-based routing, *identity*-based routing, *covering*-based routing and *merging*-based routing. These algorithms concentrate on the forwarding of subscriptions and unsubscriptions. In addition to these algorithms there is the possibility to use *advertisements*. Before turning to the assumptions we took here, we describe the functionality of each algorithm and the use of advertisements.

**Simple-based routing**. The simple-based routing algorithm is based on a flooding of the (un)sub-scriptions towards all brokers in the network. Each subscription is uniquely identified and stored in each routing table. The advantage of this algorithm is that it is easy implemented. Disadvantages are that same messages are sent more than once over the same link and routing tables grow very large in large scaled settings. In following algorithms it is tried to minimalise these phenomenons.

**Identity-based routing**. According to this algorithm, a subscription is not forwarded over a link if previous exactly a similar subscription already was forwarded over that link. An unsubscription is not forwarded over a link, if there is still another similar subscription active. Figure 1.5 shows an example of identity-based routing. In the left part of this figure (1.5(a)) subscriber $x$ has sent a subscription with filter $F$, which is forwarded from broker $A$ to $D$. This subscription is saved in the routing tables as an entry $(F, Z)$, where $Z$ equals the broker (or client) the subscription is received from. In the right part of the figure (1.5(b)) another subscriber $y$ sends a subscription with the same filter $F$. The routing tables at $B$ and $C$ are updated as the subscription is forwarded. However, due to the identity-based routing algorithm it is not forwarded to broker $D$ anymore, because the subscription $F$ is already forwarded to $D$.



(a) Forwarded subscription F from x

(b) Forwarded subscription F from y

Figure 1.5: Identity-based routing

**Covering-based routing**. As subscriptions are filters on the content of information, it is possible that a subscription on some information covers another subscription on a smaller part of that information. The covering-based routing algorithm uses this property when considering forwarding subscriptions. Suppose $y$ in figure 1.5(b) sends a subscription on filter $G$ instead of $F$, where $F$ covers $G$ ($G \subset F$). The routing tables at $B$ and $C$ will be updated with resp. $(G, y)$ and $(G, B)$. However, also now the subscription will not be forwarded from $C$ to $D$, as the information in $G$ is covered by $F$.

**Merging-based routing**. The merging-based routing algorithm can be used on top of the previously described algorithms. It is based on merging subscriptions together and forward it as one subscription containing all filters. Naturally, it has to be defined how long messages are gathered, before sending them forward.

**Advertisements**. Also advertisements can be used in addition with all previously described algorithms. Advertisements are used by publishers to indicate which information they contain and are willing to send over the network. Just as subscriptions, the advertisements are filters on the information and forwarded into the network. Furthermore, the publisher sends an unadvertisement to cancel its notifications on that subject. By using advertisements, the subscriptions can be propagated on the path established by the advertisements and future notifications will travel the same path the other way around. So, brokers need two routing tables: one for subscriptions and one for advertisements. From this we see that notification traffic and subscription traffic are minimalised, however with an additional price of advertisement traffic.

In our models we will assume that the subscriptions on information of a publisher containing one type of information are all exactly the same. Due to this assumption, the covering-based routing algorithm does not differ from the identity-based routing algorithm. Accordingly, as the simple-based routing algorithm is not optimal, we will use the identity-based routing algorithm as basis for our model in Chapter 2. Moreover we will extend the model by using advertisements.

### 1.4.3 Dynamic aspects

We assume the publish/subscribe system to be dynamic. With dynamic we mean that the contributing parties of the system are able to change in time. We distinguish between three dynamic aspects, namely due to a changing network topology, clients behaviour at one point and roaming clients:

- **Changing network topology**. Changing topology due to broker or link failures and the addition of new brokers or links.
- **Clients behaviour at one point**. Connected to the same broker, clients change between connected and disconnected states and change their interests (subscribers) or notifications (publishers).
- **Roaming clients**. Clients are able to move physically and change from (local) broker which attach them to the network.

The publish/subscribe system must be reliable in real situations and recover from failures (not only link or broker failures, but for example also faults in content of messages). Especially military networks deal with these failures. To deal with this property, we will assume a self-stabilising solution by assuming subscriptions to be refreshed periodically. So, as long as a subscriber is active it frequently re-sends its subscription (a *re-subscription*). Its local broker will re-new the routing entry by removing the old subscription and replace it by the re-subscription. Next it forwards the re-subscription to its neighbour brokers which will perform the same tasks. In this manner all routing entries in the network for this subscription are refreshed. Jaeger and Mühl [6] also used this property in their analysis. We assume notifications are already sent frequently, containing up-to-date information. Note that we will not model a changing network topology where for example links or brokers drop out or appear. For comparing flooding with more sophisticated filtering we need a similar network for both schemes, so we consider a fixed network structure as described in 1.4.1.

In our model we will analyse the clients behaviour at one point by assuming a client changing between active and inactive state. When a client becomes active it sends a subscription on information, while active frequently re-sends the subscription and when it switches to the inactive state an unsubscription is sent. As there is one type of information for a subscription, we will not consider the situation where a subscriber changes its interests while active. However, we will also propose a model where we consider roaming clients.

Important to mention is that we will assume *no delay* in the system for both sending a message over a link as the time needed to consult the routing tables at the brokers. So a message sent somewhere, will directly reach all locations it is propagated to. Another aspect which can be applied in publish/subscribe systems is the buffering of information at brokers. For example to guarantee some degree of delivering of (old) notifications at moments clients where inactive and change to the active state. Or another service could be when a client roams from one broker to another: the information stream towards its old local broker should be buffered and stored until the client reaches the other broker (where the buffered information should be propagated to). In our model we assume that the notifications active clients receive, always contain the newest information. So, we will not consider buffers in the network. However, for further research it is interesting to examine buffering of information and its impact on for example buffer sizes and amount of message traffic.

## 1.5 Structure of this thesis

In this chapter we described a lot of background information and defined the research objectives for this research project. Moreover, we gave a thorough overview of relevant literature and defined the scope in which we will perform the research.

The remainder of this thesis is structured as follows. In Chapter 2 we will describe our model from which we are able to compare more sophisticated routing with flooding. It starts with the derivation of a basic model and ends with the description of some extensions to this model. Next, in Chapter 3, a reference model of a radio-network is created from our model and conclusions are drawn in exemplary settings. Similar work is done in Chapter 4 for the CAFES-environment. Finally in Chapter 5 the main conclusions and recommendations are described.

In the Appendix A a list of frequently used symbols is shown.

# Chapter 2

# Modeling content-based routing

In this chapter the models for content-based routing algorithms are derived. Our goal is to compare a flooding algorithm and a more sophisticated filtering algorithm (which will be the identity-based routing algorithm). At first we will derive a model with simplified assumptions and throughout the remainder of the chapter extend this model by taking more assumptions into account.

In section 2.1 we will define the basic assumptions of the modeling environment concerning the network topology and routing mechanism. Next, in section 2.2 we derive a model by considering the behaviour of clients in a static situation. From this model some first conclusions about the comparison of flooding with identity-based routing can be derived. In section 2.3 we will define the basic approach for the model in a dynamic situation (i.e. dynamic behaviour of clients) according a steady state approach and will extend some of the assumptions. Subsequently, in section 2.4 the dynamic situation is extended by considering a transient approach where the behaviour of clients is varying over time. Additionally, we investigate the influence of a varying arrival rate when compared with a homogeneous arrival rate in this section. In the final section 2.5 of this chapter we describe several extensions on the model. Among these a generalisation, where clients are possibly located anywhere, the use of advertisements in the model and two approaches how to deal with covering-based routing and roaming clients.

## 2.1 Preliminaries

In section 1.4 of the introduction we defined the scope of the modeling environment we are considering. In this section we will define the basis of our modeling approach. For the construction of the environment for a publish/subscribe system we make simplified assumptions about the network topology and the routing mechanisms in the network.

We consider a hierarchical network topology according to a $m$-ary tree with $k$ levels representing the distributed brokers of the event service. The $m$ denotes the number of children each node contains and the $k$ denotes the depth of the tree. In figure 2.1 the nodes represent this structure where $m = 2$ and $k = 4$. So, the nodes represent the brokers of the distributed publish/subscribe system and the links interconnect these brokers. We assume subscribers only being present at the leaf nodes (i.e., the *end-brokers*) of the network. Furthermore we assume one publisher being present at the root node (*root-broker*[1]) of this tree. The nodes without clients are the routing brokers. The number of brokers on the $i$-th level equals $m^i$. Hence, the total number of end-brokers is $N = m^{k-1}$. The total number of communication links between the brokers is the total number of brokers minus one: $l = \sum_{i=0}^{k-1} m^i - 1 = \frac{m^k-1}{m-1} - 1 = \frac{m^k-m}{m-1}$.

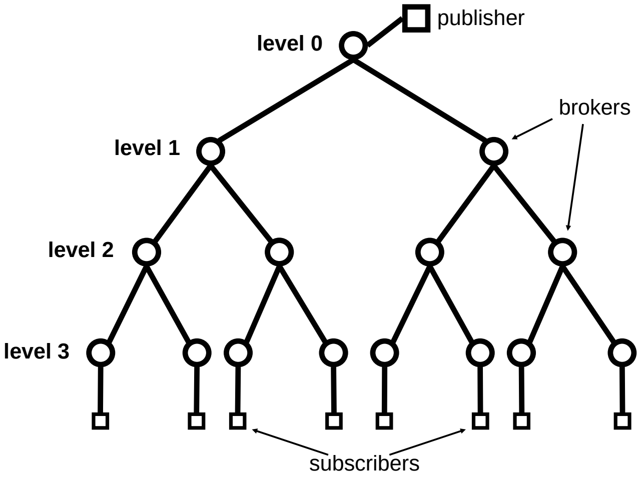


Figure 2.1: Example of $m$-ary tree with $k$ levels, with $m = 2$ and $k = 4$

The routing algorithms we consider are flooding and identity-based routing. In this network flooding means that when a publisher sends a notification into the network, its local broker (the root-broker) will forward this message towards all its neighbour brokers (in this case its children), which on their turn will forward this message towards their children until the (local) end-brokers are reached.

If identity-based routing is implemented, two streams of message traffic are flowing through the network: the subscriptions from the subscribers at the end-brokers and the notification

---

[1] Note that the end-brokers and root-broker are the local brokers as defined previously.

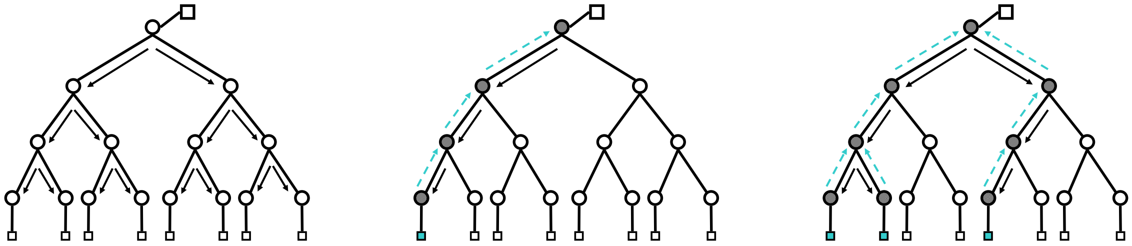
from the publisher at the root. We assume subscriptions only have to be propagated towards the root (as we assume a prior knowledge where the publisher is located). When a subscriber sends a subscription into the network, its local end-broker will consult its routing table to check whether it has already forwarded a same subscription towards its parent broker. I.e., the broker matches the new arrived subscription with its routing entries. The subscription is only forwarded to its parent broker if there is no match. Each broker receiving a subscription will start the same process. So a subscription is forwarded from the bottom of the network towards the root, until a broker is reached which already has a routing entry on this notification in its routing table. The notifications sent by the publisher will follow the paths which are established by the routing tables. So when a notification is published, the broker which receives the notification will consult their routing table again and will forward the notification towards those child brokers which are subscribed on this notification.

We assume that there is one notification-message holding one type of information. This means that each message is a duplicate of this message and contains the same content. Analogue we have one subscription-message, which is a filter on the notification. The size of a message is 1 unit and the costs for sending one message over a link equals 1 unit of traffic amount. A final assumption is that messages sent over the links from clients to local brokers and vice-versa is free, because we are only interested in the traffic between the brokers.

## 2.2 Static situation

We are interested in which situations identity-based routing is preferred above flooding when total traffic in a network has to be minimised. To give some first insights for answering this problem, we assume the environment to be static. Namely, by considering the amount of traffic that arises through one flood of a notification message in the network and the amount of traffic that arises when identity-based routing is used when one notification is published. The latter depends on the number of subscribers that is interested in the notification and the distribution of these subscribers over the network. The following example clarifies this concept. Remember that the traffic caused by the notification is data-traffic and traffic caused by subscriptions control-traffic.

**Example**. *Consider* $k = 4$ *and* $m = 2$ *and the representations in figure 2.2. A message traveling over a link is indicated by an arrow in the direction it is forwarded. Figure 2.2(a) shows flooding. Here all links in the network are traveled, independent on the amount of interested subscribers, resulting in a total of 14 units of data-traffic. In the other two figures two examples of identity-based routing are shown. In figure 2.2(b) just one subscriber is interested in the information, so there will be a forward of subscriptions towards the root. Conversely, the notification will travel directly towards this subscriber along the path the subscription created. This results in a total of 6 units of traffic (3 data plus 3 control). In figure 2.2(c), three subscribers* *are interested, resulting in a total of 14 units of traffic (7 data plus 7 control). This is equal to the amount of traffic with flooding. However, if the three clients where distributed to the three most left end-brokers, the traffic would be* 12*; less than flooding.*



(a) Flooding

(b) Identity-based routing with 1 subscriber

(c) Identity-based routing with 3 subscribers

Figure 2.2: Traffic flow

So the amount of traffic using identity-based routing depends on the number of subscribers *and* the distribution of the subscribers over the end brokers. Intuitively, a distribution of subscribers as close as possible to each other causes less message traffic than a distribution of subscribers as dispersed as possible. We will distinguish between three possible distributions of the subscribers, namely (1) a best-case distribution, (2) a worst-case distribution and (3) a homogeneous distribution. Before describing these, we will first define the modeling parameters we use.

Let $B_{i,j}$ denote the state of a broker at position $j$ at level $i$ in the network, where $i = 0, ..., k-1$ and $j = 1, ..., m^i$. Figure 2.3 shows an example of this numbering for the tree with $m = 2$ and $k = 4$. We assume the state of a broker $j$ at level $i$ equals either 0 or 1 according to:

$$B_{i,j} = \begin{cases} 1 & \text{if the routing table is occupied with a filter;} \\ 0 & \text{if it is not occupied.} \end{cases} \tag{2.1}$$

For calculating the amount of messages flowing through the network, we need the number of brokers with an occupied routing table. This number tells us how many links are used for propagating both subscriptions as notifications. For each given distribution of active subscribers at the end brokers we can determine this number according the following observation. If there is an active subscriber at end broker $j$ (in level $k-1$), the state of this broker will be assigned $B_{k-1,j} = 1$, as the routing entry at this broker will be occupied by the filter of the subscriber. According to the identity-based routing algorithm, the subscription is forwarded up in the hierarchy towards the root. So, the parent brokers up in the hierarchy have to be assigned the value 1 too. Hence, given a number of active subscribers, distributed over the end brokers, we are able to assign all values $B_{k-1,j} = 1$ if there is an active subscriber at

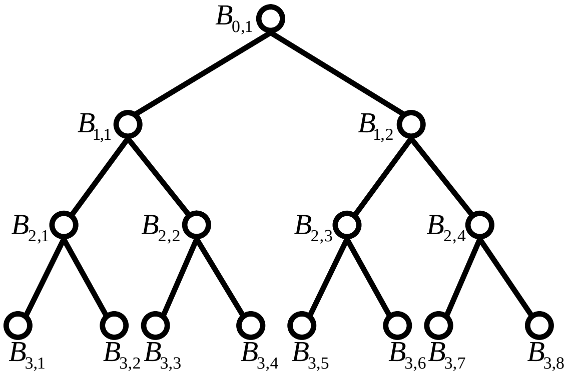


Figure 2.3: Broker numbering $m$-ary tree with $k$ levels

broker $j$ and $B_{k-1,j} = 0$ if there is no subscriber. For each parent broker at position $(i, j)$ up in the hierarchy its routing table is occupied if at least one of its child brokers (with positions $(i + 1, jm - m + 1)$ to $(i + 1, jm)$) is occupied. So, by using the following recursive relation we find all occupied brokers in the network:

$$B_{i,j} = \begin{cases} 1 & \text{if } \sum_{h=jm-m+1}^{jm} B_{i+1,h} \geq 1; \\ 0 & \text{else.} \end{cases} \tag{2.2}$$

Let the $k$ by $N$ matrix $\mathbf{B}$ contain all state values $B_{i,j}$ derived from the assignment to the end brokers and recursive relation 2.2, where the states that do not exist receive the value 0. It is easy to express the number of links used ($L$) for propagating messages from this matrix. This is, all brokers with an occupied routing table minus 1:

$$L(\mathbf{B}) = \sum_{i=0}^{k-1} \sum_{j=1}^{m^i} B_{i,j} - 1 \tag{2.3}$$

For calculating the amount of messages caused by identity based routing ($A_{ibr}$) we observe that subscriptions are sent once over all links and notifications are sent once over all links, i.e.:

$$A_{ibr} = 2L(\mathbf{B}) \tag{2.4}$$

As already discussed, $A_{ibr}$ varies for the number of subscribers and the distribution of the subscribers over the end brokers. Hence, for each possible number of active subscribers $S = 0, ..., N$ and three distinct distributions of subscribers over the end brokers we will calculate expression 2.4 and compare these with flooding. Note that the different distributions need to assign values to $B_{k-1,j}$ according the input of $S$ subscribers, so from 2.2 and 2.3 $A_{ibr}$, can be found. Given the number of subscribers that are active, we distinguish between the following three ways to assign them to the end-brokers:

1. **Best-case distribution**. The distribution that causes as less traffic as possible as subscribers will be active as close together as possible. We will model this by assigning the first active subscriber to the most left end-broker and add each next active subscriber to the end-broker on the right of the previous.
2. **Worst-case distribution**. The distribution that causes as much traffic as possible as subscribers will be active as dispersed as possible. This is formed by assigning the first active subscriber to the most left end-broker and add each next active subscriber to the end-broker in a sub-graph of the tree that is most far away from the already assigned active subscribers. Below, we will explain this in more detail when we describe the corresponding algorithm.
3. **Homogeneous distribution**. According this distribution, active subscribers are added randomly to the end-brokers following a uniform distribution.

For each of the distribution we created an algorithm to assign the subscribers to the end-brokers $B_{k,i}$. Now $A_{ibr}$ can be found for each number of active subscribers $S$ and each distribution by assigning the subscribers according the distribution algorithms and next using the recursive relation 2.2 and finally expressions 2.3 and 2.4. Below we will describe each of the distribution algorithms. In appendix B the corresponding algorithm for the recursive relation 2.2 is given.

```
ALGORITHM BEST-CASE($S$)
initialize $B_{k-1,j} = 0, \quad \forall j \in N, \; i = 1$
start
Step 1. Is $(i \leq S)$? If yes $\Rightarrow$ step 2, if no $\Rightarrow$ stop
Step 2. $B_{k-1,i} = 1, \; i = i + 1 \Rightarrow$ step 1
```

Algorithm BEST-CASE $(S)$ will assign the $S$ subscribers at the end-brokers on positions $1, ..., S$.

```
ALGORITHM WORST-CASE($S$)
initialize $B_{k-1,j} = 0, \quad \forall j \in N$, $h = 1$ and $z = k - 2$
start
Step 1. Is $(S > 0)$? If yes $\Rightarrow$ step 2, if no $\Rightarrow$ stop
Step 2. Is $(B_{k-1,hm^z-m^z+1} = 0)$? If yes $\Rightarrow$ step 3a, if no $\Rightarrow$ step 3b
Step 3a. $B_{k-1,hm^z-m^z+1} = 1$, $h = 1$, $S = S - 1 \Rightarrow$ step 4
Step 3b. $h = h + 1 \Rightarrow$ step 2
Step 4. Is $(h \geq m^{(k-1)-z})$? If yes $\Rightarrow$ step 5, if no $\Rightarrow$ step 1
Step 5. $z = z - 1$, $h = 1 \Rightarrow$ step 1
```

The algorithm WORST-CASE$(S)$ works as follows. A subscriber is added to the most left (free) end broker in the subgraph of broker $B_{(k-1)-z,h}$. The algorithm at first varies $h$ as the position from most left to right (which is bounded by $m^{(k-1)-z}$) and adds a subscriber to each most left end broker in the subgraph of level $(k-1)-z$. At start this level equals 1 as $z$ is initiated as $z = k - 2$. If every subgraph in level 1 has one subscriber at the most left end broker (which is broker $B_{k-1,hm^z-m^z+1}$), $z$ is reduced by one, so the subgraphs at level 2 are considered for adding the remaining subscribers. Again $h$ varies from most left to right position. However, a subscriber is not added to an end broker in a subgraph of level 2 if due to previous steps a subscriber is already added there. This is why in step 2 is checked if $(B_{k-1,hm^z-m^z+1} = 0)$, i.e. if there is a free position.

**Example**. *If we consider figure 2.3 and assume $s = 3$ subscribers to be added. The algorithm will at first consider the subgraph below broker $B_{1,1}$ and add the first subscriber to the most left end broker of this subgraph, which is broker $B_{3,1}$. Next, for the second subscriber, the algorithm considers the subgraph below $B_{1,2}$ and adds it to broker $B_{3,5}$. Considering the third subscriber, the algorithm will consider the subgraphs in level 2, so at first the subgraph below $B_{2,1}$. However, as there is already added a subscriber in this subgraph, it will skip this subgraph and consider the subgraph below $B_{2,2}$. Finally the third subscriber will be added at the most left end broker $B_{3,3}$ as this entry is free.*

```
ALGORITHM HOMOGENEOUS($S$)
initialize $B_{k-1,j} = 0, \quad \forall j \in N, pos_j = j, \quad \forall j \in N$
start
Step 1. Is $(S > 0)$? If yes $\Rightarrow$ step 2, if no $\Rightarrow$ stop
Step 2. $z = \text{COUNT}_{i=1}^{N}(pos_i > 0) \Rightarrow$ step 3
Step 3. Generate $R \sim U(0,1)$, $h = 1 \Rightarrow$ step 4a
Step 4a. Is $(R > \frac{h}{z})$? If yes $\Rightarrow$ step 4b, if no $\Rightarrow$ step 5
Step 4b. $h = h + 1 \Rightarrow$ step 4a
Step 5. $B_{k-1,pos_h} = 1$, $pos = \text{REMOVE}(pos_h)$, $S = S - 1 \Rightarrow$ step 1
```

In the algorithm HOMOGENEOUS($S$) the variable *pos* is a list of length $N$ and is created to store all free remaining positions at the end-brokers when assigning the subscribers one by one. At initiation, all $j = 1, ..., N$ entries of the list will be valued with resp. $j = 1, ..., N$, i.e. all free positions. The algorithm uses two functions on this list, namely REMOVE and COUNT. The function REMOVE is invoked when a subscriber is added to a random chosen free position (which is entry $h$ in list *pos*). It will remove this position from the list by pushing forward all higher free positions one place in the list and add a 0 to the last entry ($N$) in the list. The function COUNT will count all possible free positions in the list, i.e. it counts the entries in the list that are larger than 0. According this algorithm the $S$ subscribers are randomly assigned to the end-brokers following a uniform distribution.

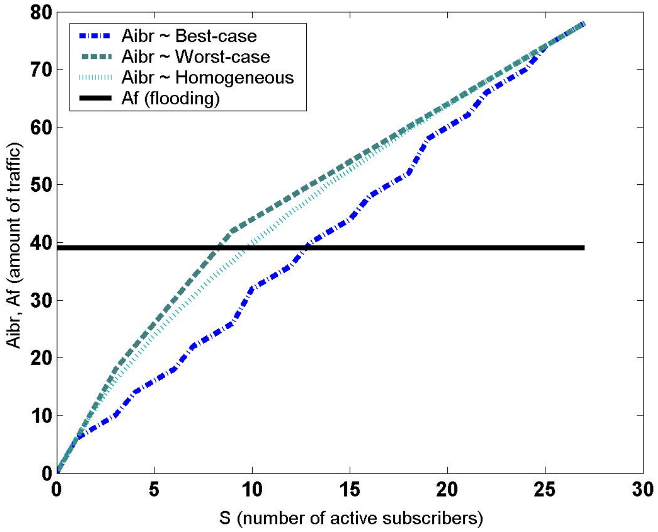


Figure 2.4: Result static situation $k = 4$, $m = 3$

Figure 2.4 shows the result of the three functions in a tree with $m = 3$ and $k = 4$, with a maximum of 27 clients. The break-even-points where identity-based routing crosses the flooding curve lie around 8 clients for the worst-case and 13 clients for the best-case. For the homogeneous distribution we took a sample of size 1000 for each $S = 1, ..., 27$ and plotted the average amount of traffic for each number of subscribers. We see that the this curve crosses the flooding line just before 10 active subscribers. As the maximum activity is 27 clients, the break-even-points lie at resp. 30% activity, 37% activity and 48% activity. Note that the jumps in the curves of the best- and worst-case occur when a new added subscriber has to be allocated in another sub-graph of the tree, so more traffic is originating. Furthermore, observe that the homogeneous distribution lies closer to the worst-case than the best-case. This follows the intuition as it is likely that according the homogeneous distribution subscribers are more dispersed than grouped close together.

From the results we are able to conclude the following. At lower activity then 30% of the subscribers, identity-based routing is always preferred. At higher activity then 48%, flooding is always preferred. However as it is likely that subscribers are dispersed a little better than the worst-case, e.g. according a homogeneous distribution, the break-even-point lies around 37% activity. Finally, we are able to conclude that when subscribers are closer together the identity-based routing algorithm improves when compared with flooding.

## 2.3 Dynamic situation - Steady state approach

In the static situation we calculated the traffic for one specific moment for a given number of subscribers. Now we are interested in the effects on the amount of generated traffic for a more dynamic situation by considering the subscribers behaviour changing in time. For modeling a more realistic environment, we take some additional assumptions to describe the behaviour in a publish/subscribe system.

In addition to the static situation, we now assume that more than one subscriber can be attached to the same end-broker. We assume subscribers independently arriving at an end-broker and become active for a while until leaving again. At the moment the subscriber arrives a subscription is sent into the network and during the active moment the subscription will frequently be re-sent to satisfy the conditions for a reliable system.

We will model the behaviour of subscribers by using the theory of queueing systems. For each separate end-broker we will consider one independent queueing system. For one such system, we assume an input process according to the arrival process of subscribers, a service distribution according to the probability distribution of the time being active and an infinite number of servers presented. The latter assumption is needed to ensure that an arriving subscriber is being served immediately (and does not have to wait). So by the service distribution we are able to indicate the time a subscriber is active. Furthermore, we assume the subscribers arriving independently to one end-broker $i$ according to a homogeneous Poisson proces, with exponentially distributed rate $\lambda_i$. For the service distribution we assume the time an individual subscriber is active is exponentially distributed with an expected duration of $\mu_i^{-1}$. This system corresponds with a $M/M/\infty$ queueing system.

The number of subscribers in the queueing system at end-broker $i$ forms a continuous time Markov process $X_i$. The values $X_i$ can take, depends on the state space $S$, denoting all possible numbers of subscribers that are allowed to be present. In this section we distinguish between two possible state spaces. At first we will derive a model where we assume that the state space is infinitely large, which means that there is an infinite population of possible subscribers. In paragraph 2.3.1 the steady state distribution corresponding with this approach is given. Next in paragraphs 2.3.2, 2.3.3 and 2.3.4 we derive the basic model for calculating the message traffic for identity-based routing step by step, using the steady state distribution with an infinite population. Finally in paragraph 2.3.5, we will consider the state space finite and derive the expressions we need for the model there.

### 2.3.1 Infinite population

We assume there is an infinite population of possible subscribers at each end-broker $i = 1, ..., N$. We define the stochastic process $X_i$ indicating the number of active subscribers at end-broker $i$, which takes values from the state space $S = \{0, 1, 2, ...\}$. The arrival process $\lambda_i$ is Poisson. Likely we have a service distribution, where the parameter $\mu_i^{-1}$ is exponentially distributed and equals the expected holding time for one subscriber at end-broker $i$. This system forms a $M/M/\infty$ queueing system for which the state-transtion diagram is given in figure 2.5.

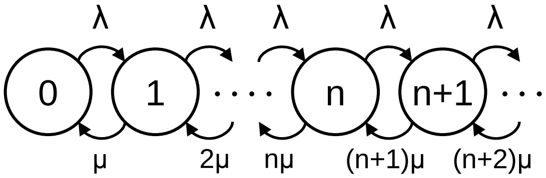


Figure 2.5: State-transition diagram of $M/M/\infty$ queue

We are interested in the steady state probabilities $P(X_i = n_i)$, which equals the proportion of time the process spends in state $n_i$. By considering the system in statistical equilibrium we are able to derive the balance equations based on the principle of global balance. However, the steady state distribution only exists if there is a $h$ which satisfies:

$$\frac{\lambda_i}{n_i \mu_i} < 1, \quad \forall n_i \geq h \tag{2.5}$$

Well, as we will not take $\lambda_i$ infinite large, this condition is satisfied and the corresponding steady state probability distribution of being in state $n_i$ for end-broker $i$ is given by:

$$P(X_i = n_i) = \frac{\rho_i^{n_i}}{n_i!} e^{-\rho_i} \tag{2.6}$$

Where $\rho_i = \frac{\lambda_i}{\mu_i}$ is defined as the *load* of the queue and equals the expected number of subscribers being active at end-broker $i$. As we can see, the probability distribution 2.6 follows a *Poisson distribution* with finite mean $\rho_i$. As we assume that the queueing systems at each end-broker are independent the joint probability distribution of having $\mathbf{X} = (X_1, X_2, ..., X_N) = \mathbf{n} = (n_1, n_2, ..., n_N)$ subscribers in the system is the product of all the single probability distributions and given by:

$$P(\mathbf{X} = \mathbf{n}) = \prod_{i=1}^{N} \frac{\rho_i^{n_i}}{n_i!} e^{-\rho_i} \tag{2.7}$$

In the next sections we will derive the formulas for calculating the message-traffic by using the steady state distribution. Again, we will use the $m$-ary tree with $k$ levels where one publisher is active at the root broker and possible subscribers are active at the end-brokers. We assume the publisher being active all the time and frequently publishes notifications according to an exponentially distributed inter-arrival time between consecutive notifications with an expected length of $\omega^{-1}$. Furthermore we assume that subscribers who are active will frequently update their subscription by sending a re-subscription each period $r$.

### 2.3.2 Traffic calculation - General approach

For calculating the total message-traffic in the network we distinguish two types of message flows through the network. We will call these *toggle traffic* and *state dependent traffic*. Toggle traffic originates at the moments new subscribers arive or at the moment subscribers leave: this traffic is caused by subscriptions and unsubscriptions. The state dependent traffic consists of the notifications and re-subscriptions that are flowing through the network in a specific state of the system. From this point of view we are able to define a function which calculates the traffic in the system when identity-based routing is implemented.

Let $a : \mathbf{N}^N \rightarrow \mathbf{N}$ be a function indicating the amount of messages per time unit that originate in a state $\mathbf{n}$. Suppose the transition rate between the states is defined as $q(\mathbf{n}, \mathbf{n}')$ for all possible transitions from $\mathbf{n}$ to $\mathbf{n}'$ and let the function $b : (\mathbf{N}^N, \mathbf{N}^N) \rightarrow \mathbf{N}$ define the amount of messages that are caused by a transition. By using the steady state distribution 2.7 which gives the proportion of time the system stays in state $\mathbf{n}$ we are able to express the total amount of traffic per time unit considering identity-based routing $A_{ibr}$ as:

$$A_{ibr} = \sum_{\mathbf{n}} a(\mathbf{n}) P(\mathbf{X} = \mathbf{n}) + \sum_{(\mathbf{n}, \mathbf{n}')} b(\mathbf{n}, \mathbf{n}') P(\mathbf{X} = \mathbf{n}) q(\mathbf{n}, \mathbf{n}') \tag{2.8}$$

The left part of this expression consists of all state dependent traffic, which is the sum of the traffic originating in each of the possible states per time unit multiplied with the proportion of time the system stays in each state. The right part of this expression consists of the average toggle traffic, which is the sum of the traffic originating between each possible transition from $\mathbf{n}$ to $\mathbf{n}'$ multiplied with the proportion of time the system stays in each state and the rate at which is toggled. In the following we will derive expressions for the functions in 2.8 for the $m$-ary tree with $k$ levels network.

As already defined, the vector $\mathbf{n} = (n_1, ..., n_N)$ of length $N$ consists of the number of active subscribers at each end-broker $j = 1, ..., N$ and the steady state distribution $P(\mathbf{X} = \mathbf{n})$ is given by 2.7. Transitions between states only occur when a subscriber arrives at one end-broker or leaves the end-broker after the completion of service (i.e. the period it is active). So, the transition rates $q(\mathbf{n}, \mathbf{n}')$ are given by:

$$\begin{aligned} q(\mathbf{n}, \mathbf{n} + \mathbf{e}_j) &= \lambda_j \quad \rightarrow \text{arrival at end-broker } j, \\ q(\mathbf{n}, \mathbf{n} - \mathbf{e}_j) &= \mu_j \quad \rightarrow \text{departure from end-broker } j. \end{aligned} \tag{2.9}$$

Where $\mathbf{e}_j$ equals the $j$-th unit vector of length $N$ with value 1 on place $j$ and value 0 elsewhere.

For expressing the function $a(\mathbf{n})$ (indicating the amount of messages that originate in state $\mathbf{n}$ per time unit) we can use the derivation of the static situation described in the previous section. There we used $B_{k-1,j}$ to indicate the activity state at end-broker $j$ and for each possible activity at the end-brokers we were able to derive the number of links that would be involved for forwarding messages. In the situation with possible states $\mathbf{n}$, we are considering now, observe that the state $B_{k-1,j}$ for each end-broker $j$ equals 1 (activity) if there is at least one subscriber active at this end-broker (when $n_j > 0$) and equals 0 (no activity) if there is no active subscriber (when $n_j = 0$), i.e.:

$$B_{k-1,j} = \begin{cases} 1 & \text{if } n_j > 0, \\ 0 & \text{if } n_j = 0. \end{cases} \tag{2.10}$$

So, for each state $\mathbf{n}$ we are able to express the activity state $B_{k-1,j}$ for all end-brokers $j = 1, ..., N$ and thus, by using the recursive relation 2.2 we can find the number of links that will be used for forwarding the messages (from expression 2.3). For convenience we will denote $L(\mathbf{n})$ as the expression for the number of links that will be used in state $\mathbf{n}$. From this, it is easy to find the amount of traffic that is flowing through the network in a state $\mathbf{n}$. Namely, notifications are sent according an average rate $\omega$ and re-subscriptions according a rate $1/r$ (as $r$ is the refresh period) over all used links in that state, i.e.: $a(\mathbf{n}) = (\omega + 1/r)L(\mathbf{n})$. By summing this expression over all possible states multiplied with the proportion of time the system stays in each state we get the total amount of state dependent traffic $A_{sdt}$:

$$A_{sdt} = \sum_{\mathbf{n}} a(\mathbf{n}) P(\mathbf{X} = \mathbf{n}) = \left(\omega + \frac{1}{r}\right) \sum_{\mathbf{n}} L(\mathbf{n}) P(\mathbf{X} = \mathbf{n}) \tag{2.11}$$

To derive the amount of messages caused by toggle traffic we need to express the function $b(\mathbf{n}, \mathbf{n}')$. We will derive this expression according the following observations. First observe that a toggle between states only affects the amount of traffic when there is a transition between states 0 and 1 at an end-broker. Hence, if the state at one end-broker toggles from 0 to 1, there is an arrival of a subscriber and at the moment of arrival, the subscriber sends a subscription towards the end-broker which directly forward this subscription towards its parent broker. Next, depending on the other active subscribers in the system, this subscription will be forwarded further into the network or not. This process is the same for a toggle from state 1 to 0, However, the only difference is that an unsubscription will be forwarded into the network. Transitions between higher states (e.g. 1 and 2) will not add anything to the toggle traffic as the identity based routing algorithm makes that duplicate subscriptions are not forwarded (and unsubscriptions are not forwarded as there is still one subscriber active). Secondly, observe that the number of links an (un)subscription has to be forwarded along due to a toggle between state $\mathbf{n}$ and $\mathbf{n}'$, is exactly the difference between the links that were used in state $\mathbf{n}$ and the links that will be used in state $\mathbf{n}'$, i.e.:

$$b(\mathbf{n}, \mathbf{n}') = |L(\mathbf{n}) - L(\mathbf{n}')| \tag{2.12}$$

Furthermore, by defining the vectors $\mathbf{n}_{j,0} = (n_1, ..., n_{j-1}, 0, n_{j+1}, ..., n_N)$ and $\mathbf{n}_{j,1} = (n_1, ..., n_{j-1}, 1, n_{j+1}, ..., n_N)$ which indicate all possible states of the system with the state at end-broker $j$ being resp. 0 and 1, we are able to express the total amount of toggle traffic $A_t$

as:

$$A_t = \sum_{(\mathbf{n},\mathbf{n}')} b(\mathbf{n},\mathbf{n}')P(\mathbf{X}=\mathbf{n})q(\mathbf{n},\mathbf{n}') =$$

$$\sum_{j=1}^{N}\left[\lambda_j \sum_{(\mathbf{n}_{j,0},\mathbf{n}_{j,1})} b(\mathbf{n}_{j,0},\mathbf{n}_{j,1})P(\mathbf{X}=\mathbf{n}_{j,0}) + \mu_j \sum_{(\mathbf{n}_{j,1},\mathbf{n}_{j,0})} b(\mathbf{n}_{j,1},\mathbf{n}_{j,0})P(\mathbf{X}=\mathbf{n}_{j,1})\right] \tag{2.13}$$

This is the sum over all possible transitions at all end-brokers $j$, where the left expression between the large brackets indicates the subscription traffic caused by an arrival of a subscriber at a 0-state of end-broker $j$. The right expression indicates all subscription traffic when the last subscriber leaves end-broker $j$ (for all possible transitions).

Note that we can express the summations for equations 2.13 and 2.11 as resp.:

$$\sum_{(\mathbf{n}_{j,0},\mathbf{n}_{j,1})} = \sum_{n_1=0}^{\infty}\cdots\sum_{n_{j-1}=0}^{\infty}\sum_{n_{j+1}=0}^{\infty}\cdots\sum_{n_N=0}^{\infty} \quad \text{and} \quad \sum_{\mathbf{n}} = \sum_{n_1=0}^{\infty}\cdots\sum_{n_N=0}^{\infty}.$$

By means of above-mentioned derivations 2.11 and 2.13 we are able to find the amount of messages flowing through the network in case of the identity based routing algorithm: $A_{ibr} = A_{sdt} + A_t$. We can derive this for various subscribers behaviour at the end-brokers on base of parameters $\lambda_j$ and $\mu_j$ and for each possible $m$-ary tree network with $k$ levels. However, for determining $a(\mathbf{n})$ (which automatically defines $b(\mathbf{n},\mathbf{n}')$) we used an algorithm based on a recursive relation. This has to be determined for all possible $\mathbf{n}$ and can be solved by any computer programs. However, we are interested in an analytical solution for $A_{ibr}$. As Jaeger and Mühl [6] did good work in this area, we will describe their approach.

### 2.3.3 Traffic calculation - Approach Jaeger and Mühl

To simplify the derivation of the functions for the amount of traffic we want to derive, we first assume $\lambda_i = \lambda$ and $\mu_i = \mu$ for all end-brokers $i = (1, ..., N)$. So for each end-broker, the steady state distribution is a Poisson distribution with mean load $\rho = \frac{\lambda}{\mu}$:

$$P(X = n) = \frac{\rho^n}{n!}e^{-\rho} \tag{2.14}$$

From here we will use this expression 2.14 and use the refined notation $p_n = P(X = n)$. Later on, in section 2.3.4 we will use the separate distributions per end-broker again. Now, for both the state dependent traffic as the toggle traffic we are able to derive expressions by considering the following observations.

**State dependent traffic** For all end-brokers we know the steady state distribution. The proportion of time one end-broker spends in the 0 state equals the proportion of time that there will not be a routing entry in the routing table at the end-broker. In other words, we know the proportion of time the end-broker has to be reached by notifications as there are active subscribers and the proportion of time there is not any activity at this end-broker. Every other broker in the network contains a routing entry for a subscription if there is at least one of the end-brokers in its subtree with an active subscriber (i.e. in a non-0-state). So, for all brokers in the network we are able to express the proportion of time they do not contain a routing entry in terms of the proportion of time there is no subscription at the end-brokers. The proportion of time one end-broker does not contain an active subscription equals $p_0$, derived from the steady state distribution 2.14. A broker in the $i$-th level contains $m^{k-1-i}$ end-brokers. The proportion of time that all these end-brokers have no active subscription equals $p_0^{m^{k-1-i}}$. From this it is derived that the proportion of time a broker in level $i$ is occupied with an active subscriptions equals $1-p_0^{m^{k-1-i}}$, i.e. the proportion of time there is at least one subscriber active in its sub-graph. This means that the link from a parent broker to this broker in level $i$ is traveled an amount $1-p_0^{m^{k-1-i}}$ of the time. As we have $m^i$ brokers in level $i$, the sum of the proportions of time each broker in the system is occupied with a routing entry equals:

$$x = \sum_{i=1}^{k-1} m^i \left(1 - p_0^{m^{k-1-i}}\right) = l - \sum_{i=1}^{k-1} m^i p_0^{m^{k-1-i}} \tag{2.15}$$

Note that we did not count the occupancy of the root-broker in level 0. This is because there is no link and no parent broker above this level. Now observe that an occupied routing table at a broker means that the link connected to its parent broker needs to be traveled. So $x$ expresses the proportion of time all links in the network have to be traveled. When $p_0$ approaches 1, the number of routing entries will become 0 (remember $l = \frac{m^k-m}{m-1}$ which equals $\sum_{i=1}^{k-1} m^i$) which is consistent with the intuition as there will be no traffic at all. Using this expression we are able to define the state dependent traffic ($A_{sdt}$). This is simply the rate at which the publisher notifies and the rate at which subscribers re-subscribe multiplied with the average number of links they have to travel:

$$A_{sdt} = \left(\omega + \frac{1}{r}\right) x = \left(\omega + \frac{1}{r}\right) \left[ l - \sum_{i=1}^{k-1} m^i p_0^{m^{k-1-i}} \right] \tag{2.16}$$

**Toggle traffic** We already explained that due to the identity-based routing algorithm toggle traffic only occurs when at one end-broker is switched between states 0 and 1. At the moment a subscriber arrives at an empty end-broker, the state switches from 0 to 1 and the subscriber sends a subscription. At the moment the last subscriber leaves, the state switches from 1 to 0 and an unsubscription is sent into the network. The amount of messages a toggle causes depends on the states of the routing brokers up in the hierarchy. If for example a new subscription arrives at an end-broker $i$ and it is the only active subscription in the complete system, it will be forwarded from end-broker towards the root-broker: a path of length $k-1$. However, when there are already other active subscribers, it is possible that a routing broker on this path already has its routing table occupied from another active subscriber. In this situation, the new subscription is not forwarded due to the identity-based routing algorithm.

If we are able to find an expression for the path-length a (un)subscription has to travel and if we know all moments when there is switched between 0 and 1 states, we are able to find the total toggle traffic. Again, by considering the steady state distribution, we are able to find both expressions in an average time period. Below we will describe how we derived both expressions.

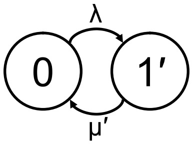


Figure 2.6: State-transition diagram of toggle states

For each end-broker we are able to find an expression for the expected number of toggles in a time period as follows. As we already know, the expected time an end-broker stays in state 0 (when started in state 0) before it toggles to state 1 is given by $\lambda^{-1}$. Now, let the expected time the end-broker remains in all other states $1' = 1, 2, 3...$ before it returns to state 0 is given by $\mu'^{-1}$. The corresponding transition diagram is given in figure 2.6 and is derived from the transition diagram in figure 2.5. Let $G$ denote a general service distribution corresponding with the service in state 1'. Now, by using the insensitive property, which states that the equilibrium distribution for the number of active clients depends on the length of the general service distribution $G$ only through its mean, we can say that the system in figure 2.6 is insensitive for the sojourn time in state 1'. Therefore we may formulate the balance equation: $\lambda p_0 = \mu' p'_1$. By definition the sum of the steady state distribution for all possible states equals 1, i.e. $p_0 + p'_1 = 1$. From both the balance equation and this property we are able to express $\mu'$ as:

$$\mu' = \lambda \frac{p_0}{1 - p_0} \tag{2.17}$$

From theorem 2.1 we know that in time period $\lambda^{-1} + \mu'^{-1}$ an end-broker will toggle two times. So the rate at which is toggled ($t_r$) at one end-broker equals:

$$t_r = \frac{2}{\lambda^{-1} + \mu'^{-1}} = \frac{2}{\lambda^{-1} + \left(\lambda \frac{p_0}{1-p_0}\right)^{-1}} = 2\lambda p_0.$$

**Theorem 2.1.** *In time period $\lambda^{-1} + \mu'^{-1}$ the end-broker will toggle two times.*

*Proof.* The stochastic process $\{X(t), t \geq 0\}$ (in this situation with state space $\{0, 1\}$) has time points where the process probabilistically restarts itself, i.e. it is a regenerative process. This, due to the memoryless property of the exponential distribution. Now, consider the process being in state 0. The expected time the process stays in state 0 equals $\frac{1}{\lambda}$ before switching to state 1. Next, the process remains in state 1 according an expected time period $\frac{1}{\mu'}$ before switching to state 0 again. So the expected length of the cycle in which the process regenerates equals $\frac{1}{\lambda} + \frac{1}{\mu'}$ where the process switches from 0 to 1 and from 1 to 0, i.e. two times. Note, that the expected cycle length for returning in state 1 when started in state 1 is exactly the same (and switches two times: from 1 to 0 and 0 to 1). ☐

Next we need to find the average path-length an (un)subscription needs to travel. At the moment of a toggle, the message will always be forwarded over the link between the end-broker and its parent, contributing in 1 unit of traffic. Now, the parent broker which receives this message, will only forward this message to its parent broker if there is no other active subscriber in its subgraph. Note that a broker in level $i$ has $m^{k-1-i} - 1$ other end-brokers in its subgraph. Hence, the probability that a message is forwarded by a broker in level $i$ equals $p_0^{m^{k-1-i}-1}$. This counts for all brokers from level $k-2$ to 1, so the expected amount of messages that occur at one toggle ($d$) equals:

$$d = 1 + \sum_{i=1}^{k-2} p_0^{m^{k-1-i}-1} = \sum_{i=1}^{k-1} p_0^{m^{i-1}-1} \tag{2.18}$$

For each single end-broker the expected amount of messages that is caused by toggling now equals $dt_r$. As we have $N$ end-brokers for which the stochastic process are independent, the amount of messages used for toggling in the total system equals:

$$A_t = Ndt_r = 2N \sum_{i=1}^{k-1} p_0^{m^{i-1}-1} \lambda p_0 \tag{2.19}$$

Combining 2.16 and 2.19 we have the total traffic: $A_{ibr} = A_{sdt} + A_t$. The amount of message flowing through the network when considering flooding is simply the number of links times the notification rate, i.e. $A_f = \omega l$.

This is the result of the work by Jaeger and Mühl. A remark has to be made that they also considered a $z$ number of different filter classes (i.e. different kind of notifications) to be present. However, as they assumed the process for each filter class independent, the results when comparing flooding and identity-based routing for one or $z$ filter classes are the same. So we only consider one filter class (notification).

From both expressions $A_{ibr}$ and $A_f$ we can derive two extremes, according the arrival rate and holding time of subscriptions:

- If $\lambda \to 0$ (and $\mu$ constant), then $p_0 \uparrow 1$ and $A_{ibr} \downarrow 0$.
- If $\mu \to 0$ (and $\lambda$ constant), then $p_0 \downarrow 0$ and $A_{ibr} \uparrow \left(\omega l + \frac{1}{r} l\right)$.

This is corresponding with the intuition that when the rate of arriving subscriptions is approaching 0, there will be no traffic at all in the system. On the other side when $\mu$ approaches 0, the expected holding time ($\mu^{-1}$) of subscriptions will become infinite large, so that the traffic in the system equals a flooding of notifications plus a flooding of subscriptions according to their rates. Here we can see that flooding depending on $\omega l$ lies between these two extremes. This means there will be a turning point when for example the arrival rate is varied from extremely low to higher values, where at lower values of $\lambda$ identity-based routing will cause less traffic and at higher values of $\lambda$ flooding causes less traffic. In Chapters 3 and 4 we will show some of the results when both these functions are compared. However, before reaching that point we will first extend the model by taking more various optional parameters into account to fulfill reality.

### 2.3.4 Traffic calculation - Final basic model

At the begin of this section we defined the steady state distribution 2.6 for each end-broker $i$ distinct as: $P(X_i = n_i) = \frac{\rho_i^{n_i}}{n_i!} e^{-\rho_i}$. Also for this distinct stochastic processes at the end-brokers we are able to find expressions for $A_{sdt}$ and $A_t$ which together form the total amount of traffic caused by identity-based routing. Again, consider the $m$-ary tree with $k$ levels and let the $N$ end-brokers be labeled from most left to right as $1, 2, ..., N$. By defining $p_{0,i} = P(X_i = 0)$ we are able to find the expressions for $x$, $d$ and $t_r$ as we derived in previous subparagraph.

The expression $x$ becomes:

$$x = l - \sum_{h=0}^{k-2} \sum_{j=1}^{m^{k-1-h}} \prod_{i=m^h(j-1)+1}^{jm^h} p_{0,i} \tag{2.20}$$

This is the sum of the proportion of time a broker has an active routing table for all brokers from level 1 to $k-1$. I.e. this is the proportion of time all links are traveled in the network. The amount of state dependent traffic becomes:

$$A_{sdt} = \left(\omega + \frac{1}{r}\right) x = \left(\omega + \frac{1}{r}\right) \left[ l - \sum_{h=0}^{k-2} \sum_{j=1}^{m^{k-1-h}} \prod_{i=m^h(j-1)+1}^{jm^h} p_{0,i} \right] \tag{2.21}$$

The toggle traffic that occurs is distinct for each end-broker as the processes are distinct. Therefore we distinct the average path-length that is traveled by a toggle-message at each end-broker $j$ as:

$$d_j = 1 + \sum_{h=1}^{k-2} \prod_{i=m^h\left(\left\lceil \frac{j}{m^h} \right\rceil - 1\right)+1}^{j-1} p_{0,i} \prod_{i=i+1}^{m^h\left\lceil \frac{j}{m^h} \right\rceil} p_{0,i} \tag{2.22}$$

This equals 1, as the link between end-broker $j$ and its parent will always be traveled by a (un)subscription, plus the proportion of time each link towards the root is not occupied with forwarding messages (i.e. the routing tables are not occupied). So also the rate at which is toggled depends on the process at each distinct end-broker and according to the derivation in previous subparagraph this becomes $t_{r,j} = 2\lambda_j p_{0,j}$ for an end-broker $j$. From this the total amount of toggle traffic becomes:

$$A_t = \sum_{j=1}^{N} d_j t_{r,j} = 2 \sum_{j=1}^{N} \left( 1 + \sum_{h=1}^{k-2} \prod_{i=m^h\left(\left\lceil \frac{j}{m^h} \right\rceil - 1\right)+1}^{j-1} p_{0,i} \prod_{i=i+1}^{m^h\left\lceil \frac{j}{m^h} \right\rceil} p_{0,i} \right) \lambda_j p_{0,j} \tag{2.23}$$

In the general approach we expressed the total amount of traffic that originates in the system as: $A_{ibr} = \sum_{\mathbf{n}} a(\mathbf{n})P(\mathbf{X} = \mathbf{n}) + \sum_{(\mathbf{n},\mathbf{n}')} b(\mathbf{n},\mathbf{n}')P(\mathbf{X} = \mathbf{n})q(\mathbf{n},\mathbf{n}')$. The left part of this expression includes the state dependent traffic and the right part the toggle traffic. For both these parts we found expressions $\sum_{\mathbf{n}} a(\mathbf{n})P(\mathbf{X} = \mathbf{n}) = A_{sdt}$ from 2.21 and $\sum_{(\mathbf{n},\mathbf{n}')} b(\mathbf{n},\mathbf{n}')P(\mathbf{X} = \mathbf{n})q(\mathbf{n},\mathbf{n}') = A_t$ from 2.23. These are the basic formulas for the model considering identity-based routing ($A_{ibr} = A_{sdt} + A_t$). The formula for flooding ($A_f$), in amount of messages per time unit, is given by:

$$A_f = \omega l \tag{2.24}$$

This is simply the rate at which is notified $\omega$ times the number of links $l$ in the network.

### 2.3.5 Finite population

Up to now we assumed the population to be infinite. However, we are also interested in situations with a finite population. Again, we assume the location of subscribers at the end-brokers of the network. For each end-broker $i$ we assume there is a finite number of subscribers $N_i$ present. The behaviour af a single subscriber can be represented as an on/off source: a client is inactive (off) during a period which is exponentially distributed with intensity $\lambda_i$ and is active (on) during an exponentially distributed period with intensity $\mu_i$. While on, the subscriber will indicate its interests by sending (re-)subscriptions towards the end-broker.

As there are $N_i$ possible subscribers, the state space of the queueing system equals $S = \{0, 1, ..., N_i\}$. In this case the steady state distribution equals the Binomial distribution:

$$P(X_i = k) = \binom{N_i}{k} \left[\frac{\lambda_i}{\lambda_i + \mu_i}\right]^k \left[\frac{\mu_i}{\lambda_i + \mu_i}\right]^{N_i - k} \tag{2.25}$$

For calculating the traffic in a system with a finite population at each end-broker, we can use the formulas 2.21 and 2.23 derived above for resp. the state dependent and toggle traffic for the identity-based routing $A_{ibr}$. However, instead of using the steady state distribution 2.6 for an infinite population, here we take $p_{0,i} = P(X_i = 0)$ from the steady state distribution 2.25 for the finite population. In Chapters 3 and 4 we will use this derivation for a finite population.

## 2.4 Dynamic situation - Transient state approach

In the previous section we assumed a homogeneous arrival process of subscribers. However, in many situations the arrival process of subscribers is varying in time. For example, there can be moments in time a lot of clients are contributing to the network and moments in time it is very quiet. Moreover it is also possible that the exponentially distributed service distribution is varying in time. By taking this behaviour into account in our modeling we use a transient state approach. Just as in the steady state approach we will derive the expressions for both an infinite population as a finite population.

### 2.4.1 Infinite population

We consider a non-homogeneous inter arrival rate of clients $\lambda_j(t)$ at end-broker $j$ which depends on the time $t$ and is Poisson distributed. The time a client is active at time $t$ is exponentially distributed with mean length $\mu_j(t)^{-1}$). The stochastic variable $X_j(t)$ denotes the number of active subscribers at end-broker $j$ at time $t$ and takes values from the state space $S = \{1, 2, 3, ...\}$. This continuous time Markov Chain corresponds with an infinite server queueing system $M_t/M_t/\infty$. For this system the time-dependent distribution is Poisson and is denoted as the probability that $n_j$ clients are active at time $t$ at end broker by:

$$P(X_j(t) = n_j) = \frac{\rho_j(t)^{n_j}}{n_j!} e^{-\rho_j(t)} \tag{2.26}$$

The load of the system (which is $\rho_j(t)$, the number of active clients at time $t$) is given by the differential equation depending on the arrival and service rate:

$$\frac{d\rho_j(t)}{dt} = \lambda_j(t) - \mu_j(t)\rho_j(t) \tag{2.27}$$

By defining functions for $\lambda_j(t)$ and $\mu_j(t)$ and solving the differential equations 2.27 we have expressions for the functions $\rho_j(t)$ and thus $p_{n_j,j}(t) = P(X_j(t) = n_j)$. From this expression we are able to derive the expression $A_{ibr}(t)$ that denotes the amount of traffic that is flowing through the network, caused by identity-based routing, at time $t$. Therefore we need the following observations. Consider one end-broker $j$. For a given moment in time $t_1$ the load of the queue is constant, i.e. $\frac{d\rho_j(t_1)}{dt} = 0$. And thus, by equation 2.27; $\rho_j(t_1) = \frac{\lambda_j(t_1)}{\mu_j(t_1)}$. So at a fixed time moment $t_1$, the time-dependent probability distribution equals a steady state distribution with constant load $\rho_j(t_1)$. In paragraph 2.3.4 we found expressions for state dependent traffic and toggle traffic, by using the steady state distribution. From these expressions we now define:

$$A_{sdt}(t) = \left(\omega + \frac{1}{r}\right) x(t) \tag{2.28}$$

I.e., the amount of messages caused by state dependent traffic at time $t$, where $x(t) = l - \sum_{h=0}^{k-2} \sum_{j=1}^{m^{k-1-h}} \prod_{i=m^h(j-1)+1}^{jm^h} p_{0,i}(t)$. The amount of messages caused by toggle traffic at time $t$ is given by:

$$A_t(t) = \sum_{j=1}^{N} d_j(t) t_{r,j}(t) \tag{2.29}$$

Where $d_j = 1 + \sum_{h=1}^{k-2} \prod_{i=m^h(\lceil \frac{j}{m^h} \rceil - 1)+1}^{j-1} p_{0,i}(t) \prod_{i=i+1}^{m^h \lceil \frac{j}{m^h} \rceil} p_{0,i}(t)$ the average path-length till the root at time $t$ and $t_{r,j}(t) = 2\lambda_j(t) p_{0,j}(t)$ the toggle rate at time $t$. Now, as $A_{ibr}(t) = A_{sdt}(t) + A_t(t)$, the total amount of traffic which is generated in the network in an time-interval of length $T$ is found by integrating this function over the length of this time-interval with respect to $t$:

$$A_{ibr}^{T} = \int_T A_{ibr}(t) dt. \tag{2.30}$$

The corresponding amount of traffic caused by flooding in a time-interval of length $T$ is given by:

$$A_f^T = \int_T A_f(t) dt = T\omega l. \tag{2.31}$$

### 2.4.2 Finite population

Also for a time-varying process it is possible to find the time-dependent probability distribution of clients being in the active and inactive state. Assuming a finite population of $N_i$ clients at end-broker $i$, again we use $N_i$ independently on/off-sources with an exponentially distributed off-period with time varying intensity $\lambda_i(t)$ and an exponentially distributed on-period with time varying intensity $\mu_i(t)$. By defining the stochastic processes $X_{i,1}(t)$ denoting the number of clients that are active at time $t$ and $X_{i,0}(t)$ the number of clients that are inactive at time $t$ (at end-broker $i$), the time-dependent distribution is found by:

$$P(X_{i,1}(t) = n_{i,1}, X_{i,0}(t) = n_{i,0}) = \binom{N_i}{n_{i,1}, n_{i,0}} \rho_{i,1}(t)^{n_{i,1}} \rho_{i,0}(t)^{n_{i,0}} \tag{2.32}$$

Where the time-dependent traffic equations satisfy:

$$\frac{d\rho_{i,1}(t)}{dt} = \lambda_i(t)\rho_{i,0}(t) - \mu_i(t)\rho_{i,1}(t) \tag{2.33}$$

$$\frac{d\rho_{i,0}(t)}{dt} = \mu_i(t)\rho_{i,1}(t) - \lambda_i(t)\rho_{i,0}(t) \tag{2.34}$$

By using the properties $\rho_{i,0}(t) + \rho_{i,1}(t) = 1$ and $N_i = n_{i,0} + n_{i,1}$, the marginal time-dependent distribution for the number of active subscribers at time $t$ is given by:

$$P(X_{i,1}(t) = n_{i,1}) = \binom{N_i}{n_{i,1}} \rho_{i,1}(t)^{n_{i,1}} (1 - \rho_{i,1}(t))^{N_i - n_{i,1}} \tag{2.35}$$

With $\rho_{i,1}(t)$ from: $\frac{d\rho_{i,1}(t)}{dt} = \lambda_i(t)(1 - \rho_{i,1}(t)) - \mu_i(t)\rho_{i,1}(t)$.

Again, by using $p_{0,i}(t) = P(X_{i,1}(t) = 0)$ from 2.35 in 2.30 we find the amount of traffic that originates in a time-interval of length $T$, when there is a finite population at each single end-broker.

### 2.4.3 Influence of a varying arrival rate

We proposed two schemes for calculating the traffic for identity-based routing and flooding: in section 2.3 we assumed homogeneous arrival and service rates and in this section we assumed non-homogeneous (time-varying) arrival and service rates. For gaining some insights in the difference between both processes, we will examine what influence a non-homogeneous arrival process has on the amount of traffic in a network by an exemplary setting.

We will consider the situation with an infinite population and similar processes at each end-broker. For the non-homogeneous arrival process we will use a sinus function of the form:

$$\lambda(t) = c + b\sin at \tag{2.36}$$

This function indicates an in time undulating arrival rate repeated each period around an average arrival rate $c$ (with $c > 0$), where $b$ denotes the maximal amplitude of the wave and $a$ the frequency (depending on the length of the period $\frac{2\pi}{a}$). We assume the function for the service time not depending on time, so $\mu(t) = \mu$.

By using these functions we are able to solve the differential equation 2.27 and find the expression for the time-dependent probability distribution 2.26 in terms of $a$, $b$, $c$, $\mu$ and $\rho(0)$ (the constant integration term that follows from solving the differential equation). Using Laplace integration $\rho(t)$ becomes:

$$\rho(t) = \frac{c}{\mu} + \left(\rho(0) + \frac{ba}{a^2+\mu^2} - \frac{c}{\mu}\right)e^{-\mu t} + \left(\frac{b\mu}{a^2+\mu^2}\right)\sin at - \left(\frac{ba}{a^2+\mu^2}\right)\cos at \tag{2.37}$$

Using this derivation and the expression for the time-dependent distribution 2.26 we are able to express the total amount of traffic caused by identity-based routing (equation 2.30) in terms of the interval length $T$, the tree-shape parameters $k$ and $m$, the notification rate $\omega$, the re-subscription rate $r$, the arrival rate parameters $a$, $b$ and $c$, the service rate $\mu$ and integration constant $\rho(0)$.

For examining the influence of the varying arrival rate first note that when $b = 0$, the arrival rate function $\lambda(t)$ will be a constant $c$. Hence, by assuming the integration term $\rho(0) = \frac{c}{\mu}$, which ensures that the system starts in equilibrium, the time-dependent distribution is exactly the steady state distribution 2.14 with $\lambda = c$. So, by taking $b = 0$ we are able to represent the homogeneous arrival process.

Considering the non-varying arrival rate we assume that the value of the amplitude $b$ is restricted to $0 < b \leq c$ (so $\lambda(t) \leq 0, \quad \forall t$). The larger $b$ the larger the amplitude and the heavier the arrival rate fluctuates. In our exemplary setting we will use $b = c$ as this gives the heaviest possible amplitude (if $b > c$ it is possible that the arrival rate function becomes negative, which is not allowed).

In an exemplary setting we will take fixed value for all parameters, except $a$, $b$ and $c$. So we have the following expression for $A_{ibr}$, where we assumed a similar arrival process at each end-broker (we derived these expressions in paragraph 2.3.3) and an infinite population:

$$A_{ibr}(a,b,c) = \int_T A_{ibr}(t)dt = \int_T A_{sdt}(t) + A_t(t)dt = \int_T \left(\omega + \frac{1}{r}\right)x(t) + Nd(t)t_r(t)dt \tag{2.38}$$

Where:

$$x(t) = l - \sum_{i=1}^{k-1} m^i p_0(t)^{m^{k-1-i}}$$

$$d(t) = \sum_{i=1}^{k-1} p_0(t)^{m^{i-1}-1}$$

$$t_r(t) = 2\lambda(t)p_0(t)$$

$$p_0(t) = e^{-\rho(t)}$$

And $\lambda(t)$ and $\rho(t)$ from resp. 2.36 and 2.37. We will compare the homogeneous arrival process (by taking $b = 0$) with the non-homogeneous arrival process (by taking $b = c$) for different values of the average arrival rate $c$ and different values for the frequency $a$. For the remaining parameters we will use $T = 25$, $k = 5$, $m = 3$, $\omega = 1$, $r = 1$, $\mu = 5$ and $\rho(0) = \frac{c}{\mu}$.

Figure 2.7 shows the difference between non-homogeneous minus homogeneous for varying $a$ and $c$. The plot is created with the computer program Matlab, which uses a numerical approach for calculating the integral of expression 2.38. We took integration steps of size $\pi/20$. At the points where the curve is below zero, the non-homogeneous arrival process causes less traffic and in cases where the curve is above zero, the homogeneous arrival process causes less traffic. Since we want to minimise the amount of traffic, we say that the one causing the least amount of traffic is the best.

If we consider $c$, the figure shows that the non-homogeneous arrival process performs better for lower arrival rates and has a minimum point where the difference between both process is the largest. When $c$ grows, there becomes a point where the homogeneous arrival process performs better and a maximum is reached. When $c$ is low, there will be more toggle traffic due to identity-based routing. So, more traffic is saved in the situation of a non-homogeneous arrival process. When $c$ is higher, toggle traffic will be lower. In case of the homogeneous arrival process there will be almost no toggle traffic at all. However in case of the non-homogeneous arrival process, there will be moments that the arrival process is close to 0, so more toggle traffic exists. The curve shows this effect.

If we consider $a$, we see that for small $a$, which means a very slow varying arrival rate, that the minimum at lower $c$ is quite high, however the maximum for higher $c$ is not that high. When $a$ grows, which means that the arrival rate will fluctuate faster, we see that the curve increases, which means that the non-homogeneous arrival process becomes worse in comparison with the homogeneous arrival process. So more variation is worse for identity-based routing. However, at a certain point there is a maximum reached (at higher values for $c$) and for larger $a$ the differences between both processes converge to 0. This is effect can be derived from the function $\rho(t)$: when we take the limit $a \to \infty$, this function converges to

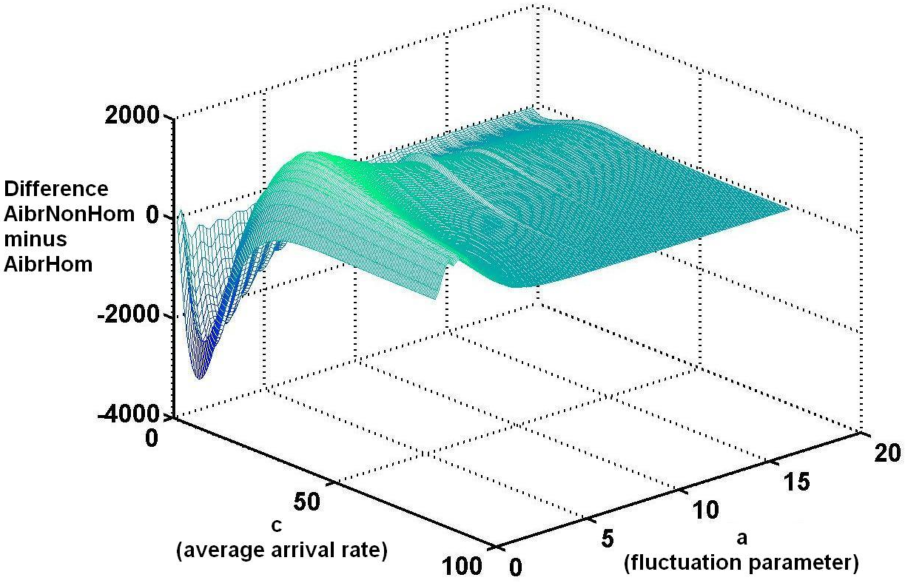


Figure 2.7: Influence on $A_{ibr}$ for non-homogeneous arrival process compared with homogeneous arrival process for different values of $c$ and $a$

$c/\mu$, i.e. exactly the parameters for the homogeneous arrival process. So when the variation is large, the load of the system will become constant.

## 2.5 Various extensions on the model

Until now we assumed one publisher always being active at the root and various subscribers being active at the end-brokers. In this section we will extend these assumptions by taking more general settings into account. In paragraph 2.5.1 we will derive a model for a general tree network, where various subscribers and publishers are possible to attach to any broker in the network. Moreover, we assume the behaviour of subscribers being distinct at different locations and on different notifications. Here, we still assume publishers always being active and identity-based routing as the forwarding algorithm.

In paragraph 2.5.2 we will assume that publishers act like on/off sources. While on they will frequently notify about some information and while off they will not send anything. For establishing the paths from subscriptions towards the publisher, we will introduce the use of *advertisement* messages in the model. By using the advertisements and unadvertisements publishers are able to indicate the network that they are active or inactive.

In paragraphs 2.5.3 and 2.5.4 we propose an idea how it is possible to add covering-based routing to the model and make some remarks on the modeling of roaming clients. In the first three paragraphs, we use the steady state approach and in the final paragraph a transient approach as described in sections 2.3 and 2.4.

### 2.5.1 Generalisation

In this paragraph we describe an approach how we can find an exact expression for calculating the traffic in any tree network. Here we assume subscribers are possibly located at any node in the network and we assume a distinct subscribers behaviour among the nodes. Besides, we assume more publishers possibly attached anywhere in the network.

Suppose we have $N$ brokers numbered from 1 to $N$ and links between the brokers forming a tree network. For each possible assignment of links between the brokers we are able to derive an expression for our model as follows. Let $H_j$ and $B_j$ be resp. the number of possible publisHers and the number of possible subscriBers at broker $j$ numbered from resp. 1 to $H_j$ and 1 to $B_j$. For all publishers $k \in (1, ..., H_j)$, for all brokers $j \in (1, ..., N)$, we assume each publisher contains distinct information, e.g. a distinct notification. So we have $Z = \sum_{j=1}^{N} H_j$ different notifications. Furthermore we assume that each different notification matches a unique subscription. For each distinct stream of subscriptions in the network for a notification we assume the identity-based routing algorithm is applied.

At this point we assume publishers (a finite number) always being active and sending their notifications following an exponentially distributed inter-arrival time $\omega_{j^k}$ for a publisher $k$ at broker $j$. We also assume the number of subscribers are finite and each single subscribe acts as an on/off source as described in paragraph 2.3.5. A subscriber at broker $i$ is not interested in notifications (off) of publisher $k$ at broker $j$ during a period which is exponentially distributed with intensity $\lambda_{i,j^k}$ and is interested (on) during an exponentially distributed period with intensity $\mu_{i,j^k}$. While on, the subscriber will indicate its interests by sending (re)-subscriptions towards its local-broker according a refresh period $r_{j^k}$. By defining the stochastic variable $X_{i,j^k}$ indicating the number of active subscribers at broker $i$ on a publisher $k$ at broker $j$, the steady state distribution of having $n_{i,j^k}$ subscribers active at broker $i$ on publisher $k$ at broker $j$ is given by:

$$P(X_{i,j^k} = n_{i,j^k}) = \binom{B_i}{n_{i,j^k}} \left[\frac{\lambda_{i,j^k}}{\lambda_{i,j^k} + \mu_{i,j^k}}\right]^{n_{i,j^k}} \left[\frac{\mu_{i,j^k}}{\lambda_{i,j^k} + \mu_{i,j^k}}\right]^{B_i - n_{i,j^k}} \tag{2.39}$$

Remember that we derived the following expressions to find $A_{ibr}$ in the steady state situation (see paragraph 2.3.4): $A_{sdt} = (\omega + \frac{1}{r})x$, the amount of messages caused by state dependent traffic and $A_t = \sum_{j=1}^{N} d_j t_{r,j}$, the amount of toggle traffic. The function $x$ is defined as the proportion of time all links are used for propagating the (notification and re-subscription) messages in the network. The function $d_j$ defines the average path-length that is traveled by a toggle-message (subscription or unsubscription) from end-broker $j$ and the function $t_{r,j}$ denotes the rate at which is toggled at this end-broker. Both these functions define the number of messages traveling through the network and depend on the topology of the network $(m, k)$ and $p_0$, the proportion of time a queue is empty (derived from the steady state distribution).

For our general network setting, let the expression $x_{j^k}$ define the total average time each broker $i \in (1, ..., N)/j$ its routing entry is occupied with subscription on the notification of publisher $k$ at broker $j$. Let the expression $d_{i,j^k}$ define the expected number of subscriptions that originates from one toggle at local-broker $i$ on the notification of publisher $k$ at broker $j$. For any given tree network with $N$ brokers, we are able to express these functions in terms of the zero-state of the steady state distribution: $P(X_{i,j^k} = 0)$.

By summing over all attending publishers in the network, we are able to express the total amount of messages flowing through the network for identity-based routing as:

$$A_{ibr} = \sum_{j=1}^{N} \sum_{k=1}^{H_j} \left( \omega_{j^k} + \frac{1}{r_{j^k}} \right) x_{j^k} + \sum_{j=1}^{N} \sum_{k=1}^{H_j} \sum_{i=1}^{N} d_{i,j^k} t_{r,i,j^k} \tag{2.40}$$

Where $p_{i,j^k} = P(X_{i,j^k} = 0)$ denotes the steady state probability for staying in state 0 and $t_{r,i,j^k} = 2\lambda_{i,j^k} p_{i,j^k}$ denotes the toggle rate for subscribers at a broker $i$ on a publisher $k$ at node $j$. In the following example we will show how the expressions for $x_{j^k}$ and $d_{i,j^k}$ can be derived for a given network.

**Example**. *Figure 2.8 shows a tree network with* $N = 5$ *numbered brokers. There are* $B_j$ *subscribers at every broker* $j$ *and there is one publisher at broker* 1 *(so* $H_1 = 1$ *and* $H_j = 0$ *for* $j = 2, ..., 5$*). For the derivation of* $x_{1^1}$*, we make the following observations: the proportion of time link* $4 - 5$ *is traveled equals* $1 - p_{5,1^1}$*. This is, the probability that there will be at least one subscriber active at broker* 5 *on the notification of the publisher* 1 *at broker* 1*. The proportion of time link* $2 - 4$ *is traveled equals* $1 - p_{4,1^1} p_{5,1^1}$*. Similar we get* $1 - p_{3,1^1}$ *for link* $2 - 3$ *and* $1 - p_{2,1^1} p_{3,1^1} p_{4,1^1} p_{5,1^1}$ *for link* $1 - 2$*. Function* $x_{1^1}$ *is the sum of these:*

$$x_{1^1} = 4 - p_{5,1^1} - p_{3,1^1} - p_{4,1^1} p_{5,1^1} - p_{2,1^1} p_{3,1^1} p_{4,1^1} p_{5,1^1}$$

*Note that the constant* 4 *equals the number of links (*$l$*). For the derivation of the expressions* $d_{i,1^1}$ *for all brokers* $i$ *we still assume that a subscription is only forwarded towards broker* 1 *where the publisher is attending. Consider a toggle at broker* 5*. At this moment a subscription or unsubscription has to be propagated over link* $5 - 4$*, with cost* 1*. Arriving at broker* 4*, it is only forwarded to broker* 2 *if there is no similar active*

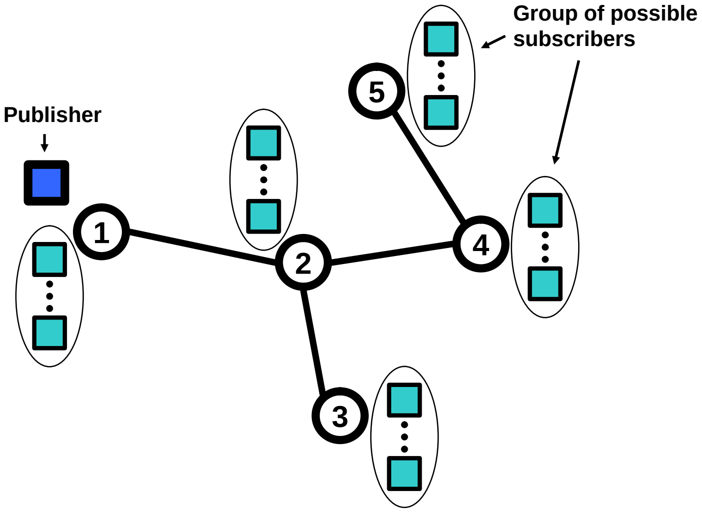


Figure 2.8: Example tree network

*subscription at broker* 4*, i.e. the proportion of time it spends in state* 0*:* $p_{4,1^1}$*. Finally the (un)subscription is only forwarded over link* $1-2$*, if there is no active subscription at brokers* 2*,* 3 *and* 4*:* $p_{2,1^1}p_{3,1^1}p_{4,1^1}$*. So, the average length the (un)subscription travels equals:* $d_5 = (1 + p_{4,1^1} + p_{2,1^1}p_{3,1^1}p_{4,1^1})$*. Consider a toggle at broker* 4*. The proportion of time the (un)subscription is forwarded over link* $4-2$ *equals* $p_{5,1^1}$ *and over link* $1-2$ *equals* $p_{2,1^1}p_{3,1^1}p_{5,1^1}$*, so in total:* $d_4 = (p_{5,1^1} + p_{2,1^1}p_{3,1^1}p_{5,1^1})$*. Similar for toggles at resp. broker* 3*,* 2 *and* 1 *we get* $d_3 = (1 + p_{2,1^1}p_{4,1^1}p_{5,1^1})$*,* $d_1 = (p_{3,1^1}p_{4,1^1}p_{5,1^1})$ *and* $d_1 = 0$ *(as the publisher is attached to broker* 1*).*

### 2.5.2 Advertisements

In this paragraph we will extend the model in the previous paragraph by assuming that publishers are now acting as on/off sources: while on they send notifications into the network according to the rate $\omega_{j^k}$ and while off they will not send any notifications. At the moment the publisher switches from off to on, it sends an advertisement message to indicate its on-state to the network. We assume that the advertisement will be flooded towards all nodes in the network. Likewise, at the moment the publisher switches from on to off, the publisher sends an unadvertisement message to indicate its off-state. Secondly we assume just as subscriptions that re-advertisements are send according to a refresh period $r_{a,j^k}$ while publisher $j$ at broker $k$ is active.

Let the activity state of publisher $k$ at broker $j$ be given by $Y_{j^k}$ (with $Y_{j^k} \in \{0,1\}$). The state $Y_{j^k} = 0$ indicates the inactive state and $Y_{j^k} = 1$ the active state. Publisher $k$ at broker $j$ is inactive during a period which is exponentially distributed with intensity $\gamma_{j^k}$

and is active during an exponentially distributed period with intensity $\nu_{j^k}$. By denoting $P(Y_{j^k} = n) = p_{Y_n,j^k}$ it is easily derived from the balance equation and normalisation that:

$$p_{Y_0,j^k} = \frac{\nu_{j^k}}{\gamma_{j^k} + \nu_{j^k}}, \quad p_{Y_1,j^k} = \frac{\gamma_{j^k}}{\gamma_{j^k} + \nu_{j^k}} \tag{2.41}$$

Before continuing we first redefine two parameters used frequently to distinguish between the processes of publishers and subscribers: $r_{s,j^k}$ will now denote the refresh period of subscriptions (which was $r_{j^k}$ previously) and we will denote $p_{X_n,i,j^k} = P(X_{i,j^k} = n)$ derived from 2.39.

Consider one publisher. During the period a publisher is not active we assume that there will be no traffic at all: subscriptions are blocked immediately at their local brokers as there is no active advertisement and clearly notifications and advertisements are not contributing the traffic. The contribution of notification and subscription traffic is thus only when the publisher is active. This means that both the state dependent as toggle traffic in equation 2.40 should be multiplied with $p_{Y_1,j^k}$ within the sum over $j$ and $k$ to allow for all publishers. Considering the state dependent traffic separately, we need to add the traffic contribution by the use of re-advertisements during the active state of the publisher. As these are flooded over all links $l$ in the network, this contribution equals the number of links multiplied with the refresh rate (which is $1/r_{a,j^k}$). So the total amount of state dependent traffic ($A_{sdt}$) equals:

$$A_{sdt} = \sum_{j=1}^{N} \sum_{k=1}^{H_j} \left[ \left( \omega_{j^k} + \frac{1}{r_{s,j^k}} \right) x_{j^k} + \frac{l}{r_{a,j^k}} \right] p_{Y_1,j^k} \tag{2.42}$$

For each publisher we need to take into account the toggle traffic for advertisement and unadvertisements. As already described we assume these messages to be flooded. So each toggle ‘costs’ $l$ links of traffic. The average number of toggles is similarly derived as the toggle traffic for subscriptions (see section 2.3). The state of a publisher $j$ at broker $k$ will namely toggle on average two times in time period $\gamma_{j^k}^{-1} + \nu_{j^k}^{-1}$. So the total average toggle traffic for advertisements ($A_{t,a}$) becomes:

$$A_{t,a} = \sum_{j=1}^{N} \sum_{k=1}^{H_j} \frac{2l}{\frac{1}{\gamma_{j^k}} + \frac{1}{\nu_{j^k}}} = 2l \sum_{j=1}^{N} \sum_{k=1}^{H_j} \frac{\gamma_{j^k} \nu_{j^k}}{\gamma_{j^k} + \nu_{j^k}} = 2l \sum_{j=1}^{N} \sum_{k=1}^{H_j} \gamma_{j^k} p_{Y_0,j^k} \tag{2.43}$$

Together with the toggle traffic for subscriptions during the active period of the publishers, the total toggle traffic becomes:

$$A_t = 2\sum_{j=1}^{N}\sum_{k=1}^{H_j}\sum_{i=1}^{N} d_{i,j^k}\lambda_{i,j^k}p_{X_0,i,j^k}p_{Y_1,j^k} + 2l\sum_{j=1}^{N}\sum_{k=1}^{H_j}\gamma_{j^k}p_{Y_0,j^k} \tag{2.44}$$

We already know that $A_{ibr} = A_{sdt} + A_t$. To compare this with flooding we need to remark that the flooding of notifications also depends on the activity period of the publisher. This means, messages are flooded over all links $l$ according to the rate $\omega_{j^k}$ for all publishers, while active:

$$A_f = l\sum_{j=1}^{N}\sum_{k=1}^{H_j}\omega_{j^k}p_{Y_1,j^k}. \tag{2.45}$$

### 2.5.3 Covering-based routing

In this paragraph we will propose an idea how it is possible to apply a covering-based routing algorithm in our model. For this purpose we need the attending messages in the network able to cover or overlap each other on their content.

Suppose we have two filters $G$ and $F$ where $F$ totally covers $G$. We assume a network consisting of one link and two brokers like in figure 2.9 where subscribers are attending the broker 1 and one publisher is attached to broker 2. We assume the publisher notifies information randomly chosen out of a set subjects with a rate $\omega$. The probability that the notification matches a filter $E$ follows a probability distribution $P_n(E)$.

We assume an infinite number of subscribers on each of the filters independently. Furthermore, the subscribers arrive independently to the broker according to a homogeneous Poisson process, with exponentially distributed rate $\lambda_G$ for subscribers on filter $G$ and $\lambda_F$ on filter $F$. For the service distribution we assume an exponentially distributed holding time of a subscription with an expected duration of resp. $\mu_G^{-1}$ and $\mu_F^{-1}$. Correspondingly we find the steady state distributions as:

$$P(X_I = n) = \frac{\rho_I^n}{n!}e^{-\rho_I} \tag{2.46}$$

Where, $\rho_I$ equals resp. $\rho_G = \frac{\lambda_G}{\mu_G}$ and $\rho_F = \frac{\lambda_F}{\mu_F}$. Further on we will denote $p_{X_I,n} = P(X_I = n)$.

As we have two filters, we distinct three possible outcomes of the probability function $P_n(E)$,

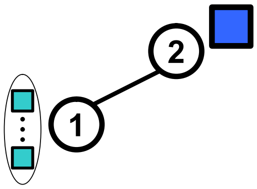


Figure 2.9: Two connected brokers

namely the probability it matches $G$ (and automatically $F$), the probability it matches $F$ but not $G$ (which is the set $H = F \cap G^C$) and the probability it does not match $F$ or $G$ (which is the set $F^C$). Let $P_n(G) = z_G$, $P_n(H) = z_H$ and $P_n(F^C) = z_{F^C}$. Obviously, the sum of these equals 1.

The state dependent traffic for covering-based routing over the link will consist of the traffic that originates at the average time merely at least one subscriber on the filter $G$ is active at broker 2, plus the average time at least one subscriber on the filter $F$ is active, i.e.:

$$A_{sdt,cbr} = (1 - p_{X_G,0})p_{X_F,0}(\omega z_G + \frac{1}{r}) + (1 - p_{X_F,0})(\omega(z_G + z_H) + \frac{1}{r}) \qquad (2.47)$$

Where $r$ equals the refresh period of subscriptions. We see that the amount of messages generated by filter $G$ (the left part of the equation) depends on the average time filter $F$ is not active, because in other situations it will be covered. This also counts for toggle traffic:

$$A_{t,cbr} = 2\lambda_G p_{X_G,0} p_{X_F,0} + 2\lambda_F p_{X_F,0} \qquad (2.48)$$

The total traffic over the single link, thus equals $A_{cbr} = A_{sdt,cbr} + A_{t,cbr}$. Compared with identity-based routing in this setting, the differences of the covering algorithm is that filter $G$ will not be forwarded when filter $F$ is active. This is exactly the term $p_{X_F,0}$ multiplied in the left parts of equation 2.47 and 2.48.

### 2.5.4 Roaming clients

In this paragraph we will add another dimension to the dynamic behaviour of clients. Namely by assuming that clients are able to roam between brokers while they are active. For example, in reality this phenomenon appears in wireless networks, where clients physically move in an area and change their connections as they are becoming in reach of different base stations (which in our model are represented by the brokers).

Lets consider the situation with a time-dependent probability distribution for an infinite server queue at each broker in a network. The stochastic process $X_j$ denotes the number of active clients at a broker $j$. By considering possible movement of subscribers between the brokers and the possibility of subscribers leaving the total system and entering the total system (at any broker), we are able to derive a model by considering the total system as an open queueing network with as many infinite server queues as brokers. From this, the time-dependent probability distribution of having $n_j$ subscribers active at broker $j$ equals:

$$P(X_j(t) = n_j) = \frac{\rho_j(t)^{n_j}}{n_j!} e^{-\rho_j(t)} \tag{2.49}$$

With traffic equations:

$$\frac{d\rho_j(t)}{dt} = \lambda_j(t) + \sum_{i=1}^{N} \rho_i(t)\mu_i(t)p_{ij}(t) - \mu_j(t)\rho_j(t) \tag{2.50}$$

Here $\rho_j(t)$ denotes the load of the queue at node $j$ at time $t$, $\lambda_j(t)$ is the external arrival rate at queue $j$ at time $t$, $\mu_j(t)$ is the service rate of queue $j$ at time $t$ and $p_{ij}(t)$ denotes the probability that a client who finished its active subscription at end-broker $i$ jumps to queue $j$ at time $t$ and becomes active as subscriber there. The probability that a subscriber leaves the system equals $1 - \sum_{j=1}^{N} p_{ij}(t)$. From equations 2.28 and 2.30 in section 2.4 we are able to find the amount of messages flowing through the network at time $t$ again by using $p_{0,j}(t) = P(X_j(t) = 0)$ from 2.49.

Due to roaming clients, the lengths of the queues (i.e. the number of active clients at the brokers) will be dependent of each other. In interesting measure for the dependency between the queue-lengths is the covariance. According to Massey and Whitt [8] in the $(M_t/M_t/\infty)^N/M_t$ model (where the final $M_t$ indicates independent nonstationairy Markov Routing) the vector of queue lengths (which is of length $N$, as there are $N$ brokers) is a time-dependent continuous-time Markov Chain. Therefore the covariance functions satisfy linear ordinary differential equations:

$$\frac{dc_{ij}(s,t)}{dt} = \sum_{k=1}^{N} c_{ik}(s,t)\mu_k(t)p_{kj}(t) - c_{ij}(s,t)\mu_j(t) \tag{2.51}$$

Where $c_{ij}(s,t) = \mathbf{Cov}\left[X_i(s), X_j(t)\right]$. By solving this differential equation we have a measure for the dependency between the queue-lengths at a broker $i$ at time $s$ and broker $j$ at time $t$.

# Chapter 3

# Content-based routing in a radio-network

The land force is a hierarchically structured organisation. In general, the higher a person is ranked, the greater its influence is on making decision and giving orders. According to the ranking, the land force is divided into several groups, including brigades, battalions, companies, platoons and squads. During combat operations, peace support operations or humanitarian operations several groups can be brought into action in which distinct groups have different tasks. For an optimal cooperation between all participants, clear and fast information exchanges, consisting of up-to-date operational knowledge, are of great importance.

Considering this perspective, one of the modernisation projects of the land force is concerned with the implementation of data-communication over radios. The publish/subscribe paradigm suits well for the proposed data to be sent in the network. However, as the available bandwidth in radio-networks is extremely low, the performance of the system is best when the number of messages sent over the network are minimised. This is why we use our model for the hierarchical data radio-network at the land force and compare flooding with more sophisticated routing mechanisms on base of the behaviour of the army units.

At first in section 3.1 a refined problem description is given. Next in section 3.2 we describe the specifications for the reference model of the radio-network followed by a mapping to our model, from Chapter 2, in section 3.3. Finally in section 3.4, the results and conclusions by varying several parameters in the reference model are derived.

## 3.1 Problem description

Up to now communication between groups is exchanged via voice over radios. However, technological developments in the area of telecommunication, make it possible to exchange data-information through these radios. This development offers great opportunities for the communication between operational groups of the land force. Therefore, in one of the modernisation projects, the *data-radio* will become part of the equipment of group members and plays a crucial role for exchanging information. The types of data that are proposed to exchange in the radio-network include *planning support* information about the operation and *situational awareness* information derived from all kind of observations in the area of responsibility. The latter is also called *actual* information and is mainly consisting of geographical position information of all (or most) contributing members of an operational group.

In comparison with a fixed wired network the exchange of data over radios suffers from more strict requirements. The main drawbacks is that the available bandwidth is extremely low. One reason for this low bandwidth is that radios which are connected to the same channel are only able to send information one at a time. In radio terminology, we say that a radio needs *access* to the channel before sending its information. The more frequently radios need access to the channel to send information, the more they are hampered by each other and not able to send all the necessary information.

By observing that information in the radio network has to be updated frequently and the environment in which the radio network has to act is highly dynamical, a possible solution is to implement the middleware of the publish/subscribe system in the radio-network. For some types of information that need to be received by all participants in the network, the flooding mechanism seems be superior above more sophisticated routing mechanisms. However, for other types of information only a small group of members is interested. Therefore it is interesting to investigate the situation where information is sent more specific, towards small groups of interesting members, which can be implemented by the more sophisticated routing schemes. By using our model we are able to indicate in which situation (flooding or more sophisticated routing) the network usage is minimal depending on the number of interested members and their dispersion in the network.

Much of the parameters describing the environment in which the data-radio-network at the land force has to be established are known. So, on base of these parameters we will construct the model to answer the question: *at which number of interested subscribers is a more sophisticated routing scheme preferred above a flooding scheme when total message traffic in the radio-network is minimised?*

## 3.2 Specifications

In this section we describe the specifications for a reference model considering the radio-network which is mapped to our model. At first we will describe the topology of the radio-network in the land force followed by the information that has to be sent in the network. Next, we will derive some additional assumptions and values for the parameters.

### 3.2.1 Topology of the radio-network

The construction of the radio-network for the land force is based on the hierarchical organisation structure as described at the beginning of this chapter. The contributing parties are vehicles in the higher levels and single soldiers in the lowest level. A vehicle contains staff (e.g. a company leader in the company net) or is a supporting vehicle for logistics, maintenance and engineering. Figure 3.1 shows (partly) the hierarchical structure from squad level to battalion level. Each ellipse from battalion level to platoon level forms a distinct radio net consisting of a group of vehicles connected to this net. The connection *between* the radio nets is performed by the vehicles which are connected to two radio nets. In practice this means that this vehicles contain two radios on two different channels. Note that there are always two vehicles interconnecting two nets. This is to guarantee a more reliable connection; if the first vehicle breaks down, the second vehicle is used as back up.

The connection between platoon nets and squad nets is not performed by a vehicle, but by the squad leader. This means, the squad leader has two radios: one for communicating with its squad members, and one for communicating with members in the platoon net. It has to be noted that the communication through the squad leaders radios is only applied when the squad is operating outside their vehicle. However, as during operations this will occur most of the time, we assume the squads always being outside their vehicle.

In our reference model we take the following assumptions considering the size of the radio-network. We assume that the radio-network is present in a complete deployed battalion which consists of 160 vehicles and 768 single soldiers. Table 3.1 shows all nets of the radio-network and the number vehicles/soldiers attending each separate net. As can be derived 1 battalion consists of 4 companies, 24 platoons and 96 squads. By summing all vehicles per net we get $1 \times 4 + 4 \times 22 + 24 \times 4 = 188$ vehicles. However, as there vehicles connected to two nets we have counted these twice, i.e. we have to subtract $1 \times 4$ for the vehicles between company and battalion level and $4 \times 6$ for all vehicles between platoon and company level to obtain the 160 vehicles. Together with the 768 soldiers (which is 96 squads times 8) we have thus 928 participants in the total radio-network divided in 125 separate nets.

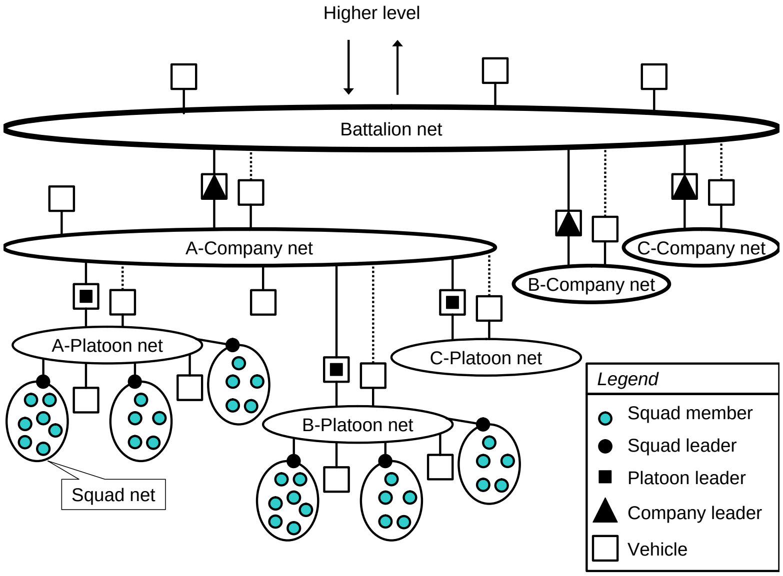


Figure 3.1: Radio net at the land force

### 3.2.2 Data types

As already described in 3.1 we distinguish between two different types of data which should be exchanged in the radio-network, namely:

- **Planning support information** (or command & control information): comprehensive information about the mission, which is generated at a higher level (for example a base station). It is also possible this information contains location and actual information from all attendant army units (in one message).
- **Situational awareness information** (or actual information): this provides the user information about his own position as well as position of others (own, opponent, unknown, etc.) and other information that represents the actual situation in the operating area. The main contributer will be the geographical position information (composed of longitude and latitude coordinates received from a GPS) of all attending units.

Planning support information is not sent often: there are large intervals between two consecutive updates of information. One can think of intervals of half-an-hour or an hour. Furthermore, this information is mostly only sent to leaders of groups, as they need to give orders to

| Net | Underlying nets | Number of vehicles/soldiers |
|---|---|---|
| Battalion | 4 company nets | 4 vehicles (each one connected with a company) |
| Company | 6 platoon nets | 22 vehicles (1 connected with the battalion,<br>6 connected with a platoon,<br>15 single vehicles) |
| Platoon | 4 squad nets | 4 vehicles (1 connected with the company,<br>besides, each connected with a squad) |
| Squad | - | 8 soldiers |

Table 3.1: Distribution of radio nets and vehicles/soldiers in our reference model

their team members. Other characteristics of this information is that the size of the message is relatively large and the direction of the information stream through the network is from top towards bottom.

The common characteristic of situational awareness information is that all (or most) of the units are able to sent information, i.e. they are all able to act as publisher. Some types of situational awareness information are sent more occasionally (like detections of threats or positions of enemies) and some types are sent very often (like position information of own units). The size of these messages are relatively small and as everybody in the network is able to act as publisher and subscriber on the information, information streams are possible in all directions. Note that it is preferred to receive up-to-date information as often as possible. Especially position information, because as attending units (vehicles and men) are moving through the area of responsibility, it is likely to spot them on their current position. In this context one can think of position updates with intervals of 10 seconds to 1 minute.

### 3.2.3 Behaviour of participants

We assume the behaviour of subscribers acting as on/off sources on each specific type of information. Subscribers switch between active periods and inactive periods according the following possible events during the mission:

- The possibility of switching their equipment on and off. While off, the subscriber is inactive, while on, the subscriber is active. A main reason why this event is taken into account is to save battery.
- Due to movement in the area of responsibility radios can become out of reach. During the moment the subscriber is out of reach it is inactive and during the moment it is in reach of the radio-network it is active.

- As third there is the possibility for clients to change their activity 'by hand'; by indicating that they want to receive information or not.

## 3.3 Mapping

For the derivation of our model, we will focus on the situational awareness information only. This is because this type of information is most frequently sent and loads the radio-network most. Now, observe that the amount of messages that are generated for a type of situational awareness information in the radio-network, depends on the position of the publisher, the number of active subscribers in the network and the positions of the subscribers in the network (e.g. all close together or as dispersed as possible). Considering this observation we will derive a model in which we are able to vary these parameters.

In paragraph 3.2.1 we described the topology of the radio-network and gave the number of radio-nets and army units per single net. For modeling we distinguish each radio-net separately, therefore we will indicate each single net as an index $(i, j)$. Here $i$ stands for the level and $j$ a position number depending on the number of nets in a level (numbered from left to right, just as in the situation of the $m$-ary tree with $k$ levels). Table 3.2 shows the possible indexes for each level $i = 0, ..., 3$ (where 0 is the highest level: the battalion, etc.) and the number of army units $N_i$ in each single net in level $i$. These numbers are derived from table 3.1. Note that we assume no army units in the battalion-net. We assume that the 4 vehicles that form the connection between battalion-net and the 4 company-nets are located in their own company-net. This because we assume that these vehicles that contain two radios only use the radio in the company-net to indicate their interest. Naturally, communication between these vehicles or other army units in different company-nets will go through the battalion-net. Figure 3.2 shows the connections between the separate nets and the complete radio-network for our reference model.

| **Level** $i$ | **Number of nets** | **Index** $(i, j)$ | $N_i$ |
|---|---|---|---|
| 0 (battalion) | 1 | $(0, 1)$ | $N_0 = 0$ |
| 1 (company) | 4 | $(1, 1), ..., (1, 4)$ | $N_1 = 16$ |
| 2 (platoon) | 24 | $(2, 1), ..., (2, 24)$ | $N_2 = 4$ |
| 3 (squad) | 96 | $(3, 1), ..., (3, 96)$ | $N_3 = 8$ |

Table 3.2: Index numbering of radio-nets and number of participants in each net

We will consider the situation where one squad is acting as a publisher of information, containing for example a message with all positions of the squad members. We will assume this is the squad in radio-net $(3, 1)$ and assume they are always active and publish the message

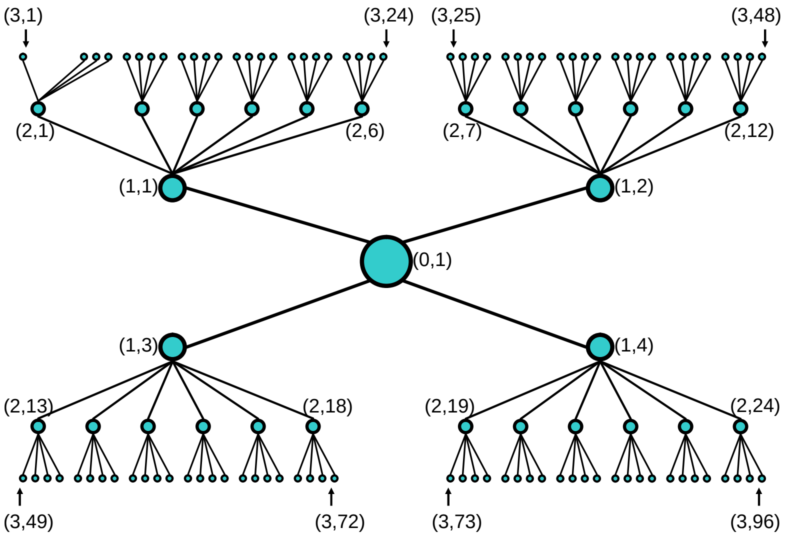


Figure 3.2: The radio-network of one battalion (till squad level)

frequently at a certain rate $\omega$. Furthermore, we assume that all members in the total battalion radio-network are able to indicate their interest in this information and act like on/off sources: a member in radio-net $(i,j)$ is inactive (off) during a period which is exponentially distributed with intensity $\lambda_{i,j}$ and is active (on) during an exponentially distributed period with intensity $\mu_{i,j}$. While on, the member acts like a subscriber and will indicate its interests by sending (re-)subscriptions towards radio-net $(3,1)$ (the re-subscriptions are sent according a refresh-period $r$). Let the stochastic process $X_{i,j}$ indicate the number of active subscribers at radio-net $(i,j)$. For each net $(i,j)$ we assume the steady state distribution of having $n$ active subscribers as the Binomial probability distribution according:

$$P(X_{i,j}=n)=\binom{N_i}{n}\left[\frac{\lambda_{i,j}}{\lambda_{i,j}+\mu_{i,j}}\right]^n\left[\frac{\mu_{i,j}}{\lambda_{i,j}+\mu_{i,j}}\right]^{N_i-n} \tag{3.1}$$

In Chapter 2 in paragraph 2.3.5 we showed how it is possible to derive the expressions we need, to calculate the amount of traffic in case of identity-based routing for a finite population and a steady state approach. Also here we need to derive the expression for $A_{ibr}$ which consists of state dependent traffic and toggle traffic. Both these are expressed in terms of $x$, $d$ and $t_r$. In the appendices C.1 and C.2 we derived these expressions for the radio-network where the publisher is located in radio-net $(3,1)$. In these expressions we used the term $p_{i,j} = P(X_{i,j} = 0)$ to indicate the proportion of time there are no active subscribers in radio-net $(i, j)$. So, the amount of traffic caused by identity-based routing is:

$$A_{ibr} = A_{sdt} + A_t \tag{3.2}$$

With $A_{sdt}$ from expression C.2 and $A_t$ from expression C.3. The amount of traffic caused by flooding is:

$$A_f = 124\omega \tag{3.3}$$

As there are 124 links in the radio-network. Observe that expression 3.2 depends on the parameters $\lambda_{i,j}$, $\mu_{i,j}$, $N_i$, $\omega$ and $r$. From the Binomial distribution it is derived that the expected number of active subscribers in a radio-net $(i, j)$ is expressed as: $N_i \frac{\lambda_{i,j}}{\lambda_{i,j}+\mu_{i,j}}$. By using this expression, the total expected number of active subscribers in the complete radio-network (denoted by $\Psi$) is given by:

$$\Psi = \sum_{(i,j)\neq(3,1)} N_i \frac{\lambda_{i,j}}{\lambda_{i,j} + \mu_{i,j}} \tag{3.4}$$

I.e., the sum of all expected active subscribers in each single radio-net (except the radio-net $(3, 1)$). By varying the arrival rate of subscribers $\lambda_{i,j}$ and fix the other parameters, we are able to compare flooding with identity-based routing on base of $\Psi$, the expected number of active subscribers in the complete network.

In next section we will show the results for different situations by varying the other parameters. At first we will assume that the interests of subscribers is uniformly distributed over the radio-network. This means that every unit anywhere in the network is equally interested in the information of the single squad. For this situation we will compare flooding with identity-based routing for (1) various notification rates, (2) various on-times of subscribers and (3) various refresh-periods. Next we will distinguish the situation in which the interest of subscribers is locally higher (closer to the publisher) than subscribers further away from the publisher. In this situation (4) we will assume various arrival-rates at distinct radio-nets.

## 3.4 Results

As discussed in paragraph 3.2.2, situational awareness information and especially position information is sent quite often. Therefore we will focus on a minute-scale for expressing the parameters.

We will not vary the number of participants $N_i$ in the nets (derived in table 3.2). However we will vary all other parameters. At first we will assume that interest in the information of the squad in radio-net $(3,1)$ is uniformly distributed. This means that we assume $\lambda_{i,j} = \lambda \ \ \forall i, j$ and $\mu_{i,j} = \mu \quad \forall i, j$. On base of these settings we will compare flooding with identity-based routing on the expected number of active subscribers (by varying $\lambda$) for:

1. Various notification rates (expressed by $\omega$)
2. Various activity periods of subscribers (expressed by $\mu$)
3. Various refresh-periods (expressed by $r$)

Finally we will consider the situation in which local interest is higher in comparison with the interest of participants further away from the squad-net $(3,1)$. So the final comparison will be made for:

4. Various arrival rates of subscribers in distinct radio-nets (expressed by $\lambda_{i,j}$)

### 3.4.1 Various notification rates

Let $r = 1/6$, i.e. subscriptions are refreshed every 10 seconds (1/6 minute) and let $\mu = 1$ denoting an average on-time of subscribers of 1 minute. In figure 3.3 we plotted the functions $A_f$ and $A_{ibr}$ on the average number of active subscribers $\Psi$ for varying notification rates $\omega$ (denoting the number of notifications sent per minute). We derived the varies values for $\Psi$ by varying $\lambda$ and calculated it from expression 3.4. In the figures in next paragraphs, we use this similar derivation.

Remember that there are a possible of 928 participants in the complete radio-network and as we assumed that one single squad is acting as publisher, the 8 members of this squad do not count with the number of possible subscribers. Hence, there are a possible 920 subscribers in the radio-network.

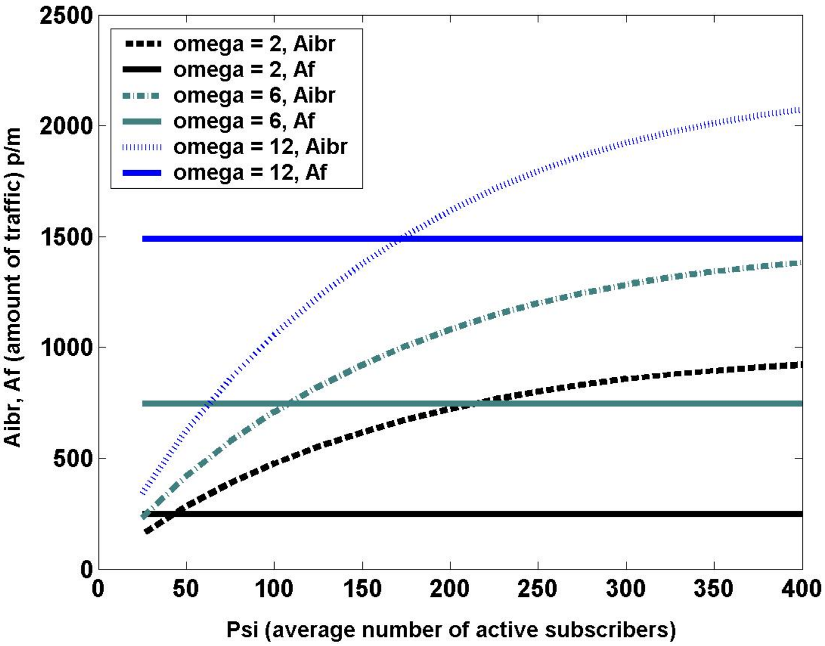


Figure 3.3: Flooding compared with identity-based routing for different $\omega$

From the figure, we see that when the notification rate raises, naturally the message traffic raises. However, the identity-based routing scheme crosses the flooding line at a higher $\Psi$. So for a higher notification rate identity-based routing improves in comparison with flooding. Well, with identity-based routing, the links are occupied with forwarding notifications for a proportion of the time and thus when the notification rate raises, the impact on identity-based routing is less than on flooding. This explains why the lines cross at higher $\Psi$.

Note that we varied $\lambda$ and expressed the graph on $\Psi$ from function 3.4. It is also interesting to see at what arrival rate the lines are crossing. So by solving the expression 3.4 for $\lambda$ we are able to express the arrival rate in terms of $\Psi$:

$$\Psi = \sum_{(i,j)\neq(3,1)} N_i \frac{\lambda_{i,j}}{\lambda_{i,j} + \mu_{i,j}} = \frac{\lambda}{\lambda + \mu} \sum_{(i,j)\neq(3,1)} N_i \quad \Leftrightarrow \quad \lambda = \frac{\Psi\mu}{\sum_{(i,j)\neq(3,1)} N_i - \Psi} = \frac{\Psi\mu}{920 - \Psi}$$

In figure 3.3 we see that identity-based routing crosses the flooding line at resp. $\Psi \approx 40$, $\Psi \approx 110$ and $\Psi \approx 170$ for resp. $\omega = 2$, $\omega = 6$ and $\omega = 12$. The corresponding arrival rates are resp. $\lambda \approx 0.045$, $\lambda \approx 0.136$ and $\lambda \approx 0.227$ and denote an average inactive-time of subscribers as resp. 22.0, 7.4 and 4.4 minutes. These are quite high in comparison with the average activity time of 1 minute.

In the following three paragraphs, were we vary $\mu$, $r$ and the distribution of the interests of subscribers we will use as fixed notification-rate $\omega = 6$. In this situation $\Psi \approx 110$ and as there are a possible of 920 subscribers, we say that the break-even-point where flooding and identity-based routing are equal, lies around an activity of 12% of the subscribers.

### 3.4.2 Various activity periods of subscribers

In this paragraph we will assume a fixed notification rate of $\omega = 6$ and a fixed refresh-period of $r = 1/6$. Figure 3.4 shows the results for $A_f$ and $A_{ibr}$ for various values of $\mu$ depending on $\Psi$.

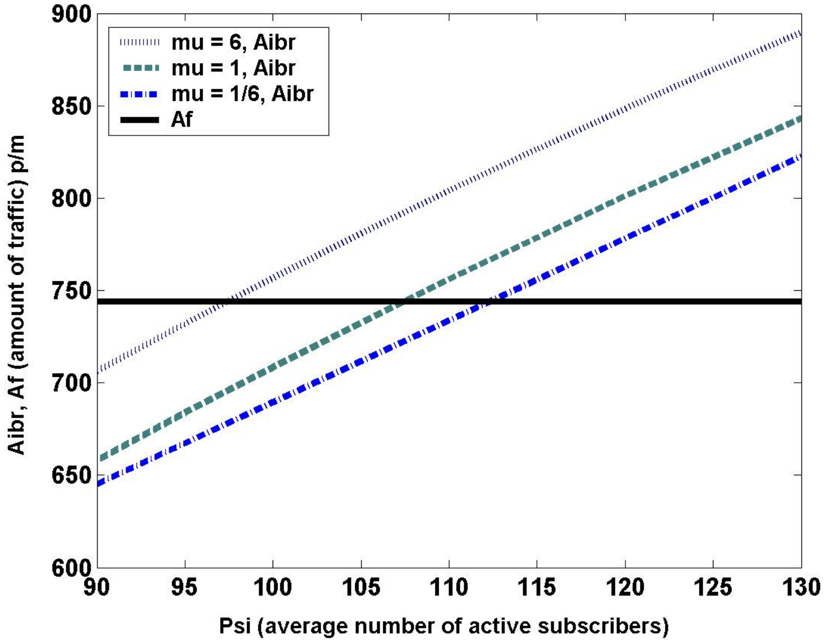


Figure 3.4: Flooding compared with identity-based routing for various $\mu$

In this figure we see that when $\mu$ decreases, identity-based routing improves when compared with flooding. This can be explained as there will be more toggle-traffic when $\mu$ is higher and less toggle-traffic when $\mu$ is smaller. However, the differences are not large. The Identity-based routing line crosses the flooding line at resp. $\Psi \approx 97$, $\Psi \approx 107$ and $\Psi \approx 112$ for resp. $\mu = 6$, $\mu = 1$ and $\mu = 1/6$. The corresponding arrival rates are resp. $\lambda \approx 0.707$, $\lambda \approx 0.132$ and $\lambda \approx 0.023$ and denote an average inactive-time of subscribers as resp. 1.4, 7.6 and 43.5 minutes.

### 3.4.3 Various refresh-periods

Here we consider a fixed $\mu = 1/6$ (i.e., an average active time of subscribers of 6 minutes) and a fixed $\omega = 6$ (i.e., on average 6 notifications per minute). Figure 3.5 shows the results for $A_f$ and $A_{ibr}$ for various values of $r$ depending on $\Psi$.

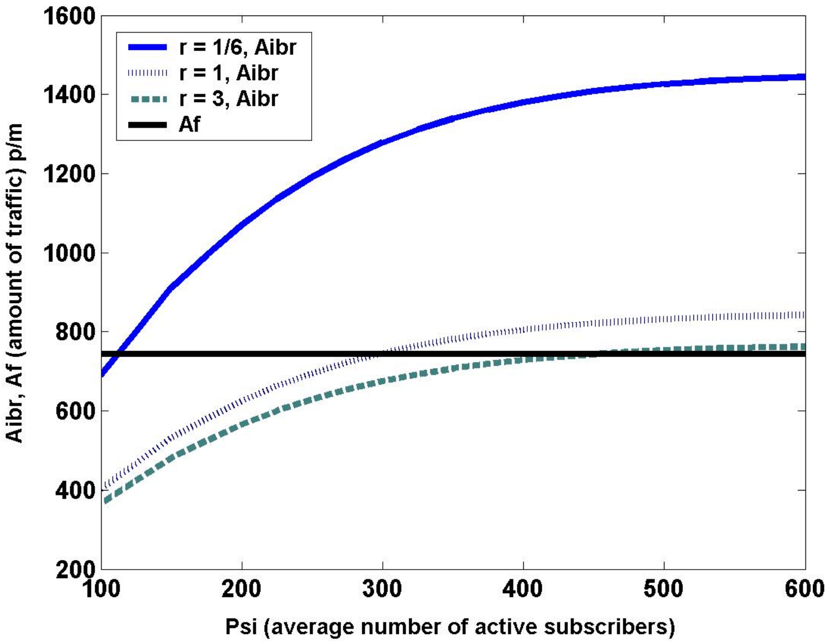


Figure 3.5: Flooding compared with identity-based routing for different $r$

We see that when refresh-periods become larger, this has a quite large impact on identity-based routing compared with flooding. We also see that the difference between flooding and identity-based routing becomes smaller. The break-even-points lie around $\Psi \approx 110$, $\Psi \approx 290$ and $\Psi \approx 460$ respectively. Expressed in a percentage compared to the 920 possible subscribers, these are resp. 12%, 32% and 50%. In conclusion, the longer the refresh-periods, the less control-traffic and the closer the ideal line is reached by identity-based routing.

### 3.4.4 Various arrival rates of subscribers in distinct radio-nets

For locality we assume three distinct behaviour among the subscribers on base of the distance from the publisher (the squad). We assume local more interest, on an average distance an average interest and further away low interest in the information of the squad. We divide these three groups over the nets as:

- Local: the members in the platoon net of the squad and the other squads attached to this platoon. In figure 3.2, this are $(2,1); (3,2); (3,3)$ and $(3,4)$. We assume a uniform arrival rate $\lambda_L$ for these nets.
- Average distance: the members in the company net $(1,1)$ and all other members below this company net, platoon $(2,2), ..., (2,6)$ and squads $(3,5), ..., (3,24)$. We assume a uniform arrival rate $\lambda_A$ for these nets.
- Far distance: all remaining nets contributing the radio-network. Here, we assume a uniform arrival rate $\lambda_F$.

Let, $\lambda_L = \lambda$, $\lambda_A = \frac{\lambda}{q}$ and $\lambda_F = \frac{\lambda}{q^2}$. By varying parameter $q$ we are able to express the local interest stronger. For fixed $\mu = 1$, $\omega = 6$ and $r = 1/6$ we compared $A_f$ with $A_{ibr}$ for $q = 1$ (uniform), $q = 10$ and $q = 100$. Figure 3.6 shows the results.

When $q$ is higher, identity-based routing improves in comparison with flooding. This is intuitively correct as a higher $q$ indicates more local interest in comparison with interest from units further away.

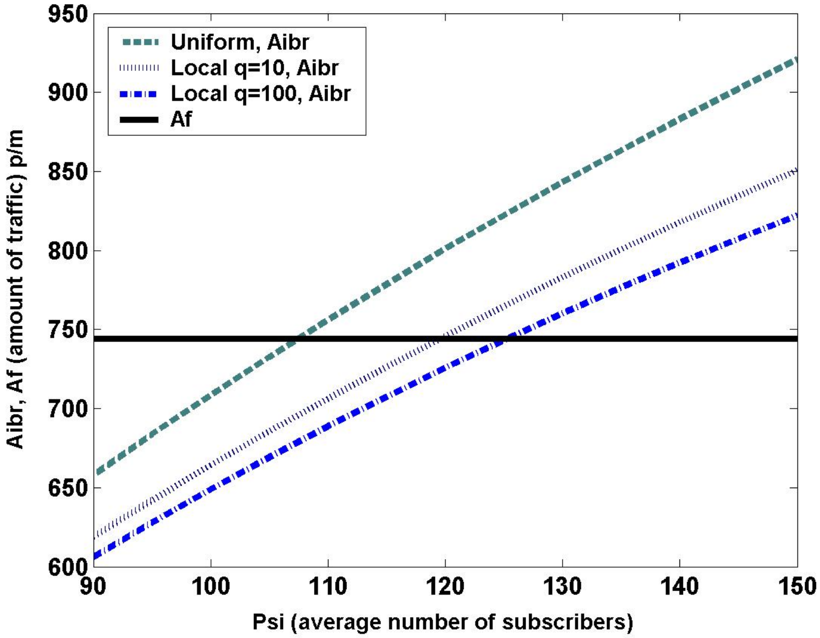


Figure 3.6: Flooding compared with identity-based routing for different uniform and local behaviour

# Chapter 4

# Content-based routing in CAFES

CAFES, currently under development at Thales Land & Joint Systems, is a system which supports Voice over IP (VoIP) communication between clients over an IP (Internet Protocol) based network. The communication includes telephony, instant messaging, presence information, multimedia distribution and multimedia conferencing. The functionality for establishing connections and forwarding and routing of messages in a network is integrated in so-called CAFES-routers, which are defined in the application layer. These routers are connected with each other through a multicast network and clients are attached to the routers following a unicast connection. The topology of a CAFES system is shown in figure 4.1.

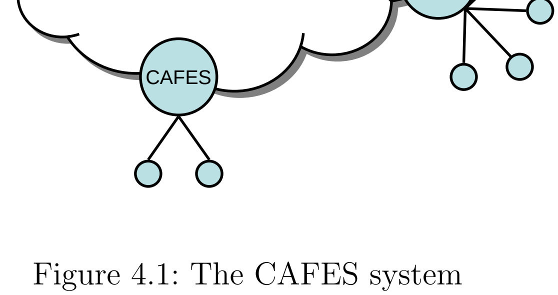


Figure 4.1: The CAFES system

One of the features the CAFES system needs to support is presence information. This type of information comprehends the status of a client (e.g. available, unavailable, busy, out to lunch, etc.). The same as in for example MSN, a client is able to pass on its status, so that other clients who are interested in that status can be aware of it. The availability of presence information is important in the CAFES system, as clients then know which of the users they can contact.

Since the CAFES system will be implemented in a highly dynamical environment, where for example connections are lost so now and then and (groups of) clients join or leave the system, it is proposed to use the publish/subscribe paradigm and the content-based routing mechanism to control the presence information. In a project which started together with this project, Ayuba [1] implemented this in CAFES. However, there was a decision to be made on how messages should be forwarded over the network, so that message traffic is minimal. By using our model, we were able to answer this question.

In the remainder of this chapter, at first we will formulate the problem. Next, in section 4.2 the mapping with the model we derived in Chapter 2 is given and finally in section 4.3 an example and results are shown.

## 4.1 Problem formulation

Just as in the general research objective, the main concern is to minimise the amount of messages flowing through the multicast network. On base of this performance indicator we will compare a flooding mechanism with a more sophisticated routing mechanism. Note that the multicast network works the same as a radio-network, i.e. if a CAFES-router sends a message into the network, all participating CAFES-routers will receive this message. The clients that participate the network are able to act as publisher and subscriber, i.e. they send notifications with their presence information towards the local broker they are connected with and subscriptions indicating the interest of receiving presence information from others. As we are only interested in the amount of messages that are flowing through the multicast network, again we assume the traffic between client and local broker is free. Now, we distinguish two different ways to implement a content-based routing mechanism over the multicast network, namely by flooding notifications and the more sophisticated routing by flooding subscriptions (where notifications will only be sent if there is an active subscription):

- **Notification flooding**. All notifications sent from clients to their local CAFES-routers are forwarded into the multicast-network. Each CAFES-router contains the subscriptions of its local clients and thus will forward the notification to the interested subscribers locally.

- **Subscription flooding**. All subscriptions send from clients to their local CAFES-routers are forwarded into the multicast-network. As a consequence each CAFES-router contains the subscriptions of all clients in the network and at the moment a CAFES-router receives a notification, it will only be forwarded into the network if there is interest in this information.

Not surprisingly, the question we want to answer is: in which situation is notifications flooding better and in which situations is subscription flooding better. The situation depends on the number of clients, the number of CAFES-routers and the behaviour of clients. In the next section we will derive the model concerning this situation from our work in Chapter 2.

## 4.2 Mapping

In this section we will derive a model that fits the message traffic that occurs in a CAFES system, depending on the behaviour of clients. This model will be based on the models we derived in Chapter 2, however there are some slight differences in the situation of CAFES. For a clear understanding we will therefore first describe the assumptions we made, before deriving the model.

In the environment where the CAFES system should be implemented, there will be a finite number of users able to participate in the network. Therefore we will assume a finite number of possible clients at each CAFES-router. Each client is either active or inactive and is able to act as publisher or subscriber. This means that a client can be both publisher as subscriber at the same time, or either one of the two separately. If a client acts as publisher it frequently sends a notification containing its up-to-date presence information. When subscriber, it frequently resends its subscription to refresh the routing tables at the CAFES-routers. We assume that each client contains unique information, so each message (either notification or subscription) is unique for one client. In the model we assume that the process of subscribing on the information of a single client (acting as publisher) by all other clients (acting as subscribers) is independent for all distinct clients (acting as publishers). Furthermore we also assume the processes of being active as publisher or as subscriber independent of each other.

Let $N$ denote the number of CAFES-routers that are connected with each other through the multicast network. At each CAFES-router $C_i$ we assume a finite population of clients $c_i$ that are possible to act as publisher and/or subscriber. Similar as the derivation in section 2.5, we assume that clients act as follows:

- A client at broker $i$ acts as a subscriber on a publisher $h$ at broker $j$, during a period that is exponential distributed with parameter $1/\mu_{i,j^h}$. And is not acting as a subscriber on the same publisher during a period that is exponential distributed with parameter $1/\lambda_{i,j^h}$. The stochastic variable $X_{i,j^h}$ indicates the number of active subscribers at broker $i$ on a publisher $h$ at broker $j$. The corresponding steady state distribution (from 2.39) of having $n$ active subscribers equals:

$$p_{X_n,i,j^h} = P(X_{i,j^h} = n) = \binom{c_i}{n} \left[\frac{\lambda_{i,j^h}}{\lambda_{i,j^h} + \mu_{i,j^h}}\right]^n \left[\frac{\mu_{i,j^h}}{\lambda_{i,j^h} + \mu_{i,j^h}}\right]^{c_i - n}$$

- A client $h$ at broker $j$ which acts as a publisher is inactive during a period which is exponentially distributed with intensity $\gamma_{j^h}$ and is active during an exponentially distributed period with intensity $\nu_{j^h}$. The steady state distribution for the activity state $Y_{j^h}$ (a stochastic variable with $Y_{j^h} \in \{0, 1\}$) is given by (from 2.41):

$$p_{Y_0,j^h} = \frac{\nu_{j^h}}{\gamma_{j^h} + \nu_{j^h}}, \quad p_{Y_1,j^h} = \frac{\gamma_{j^h}}{\gamma_{j^h} + \nu_{j^h}}$$

Next we will define the expressions for calculating the amount of messages while notifications are flooded or subscriptions are flooded. Again we can use the derivations from section 2.5. However, in the CAFES system we are not dealing with different links, but with one multicast network where one message is received by all. This slightly changes the derivation as the costs for sending one message now just equals one. Observe that with flooding of notifications there well be merely data-traffic in the network, while by flooding of subscriptions there will be both data-traffic as control-traffic. Considering this observations we will show the expressions and explain how these are derived.

The amount of messages considering *flooding notifications* ($A_{NF}$) equals:

$$A_{NF} = \sum_{j=1}^{N} \sum_{h=1}^{c_j} \omega_{j^k} p_{Y_1,j^h} \tag{4.1}$$

This is, the sum of the notification rate $\omega_{j^h}$ (for a client acting as publisher $h$ at router $j$) multiplied with the proportion of time the client is active as publisher for all $h$ client at every CAFES-router $j = 1, ..., N$.

For the amount of messages caused by *flooding subscriptions* ($A_{SF}$) we consider both control-traffic ($A_{SF,ct}$) as data-traffic ($A_{SF,dt}$) separately:

$$A_{SF,ct} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{h=1}^{c_j} \left( \frac{1}{r_{s,j^h}} (1 - p_{X_0,i,j^h}) + 2\lambda_{i,j^h} p_{X_0,i,j^h} \right) \tag{4.2}$$

The left part of this expression equals the state dependent traffic that originates when at least one subscriber is active (that is why it is summed from $i = 1$ to $N$) on the presence information of publisher $h$ at CAFES-router $j$ and depends on the refresh-period $r_{s,j^h}$. The right part denotes the toggle traffic, which consists of the toggle rate multiplied with 1, as the cost for propagating a message is just 1 unit of traffic in the multicast network. Both parts have to be summed for all possible clients (acting as publisher) to account for all control-traffic.

$$A_{SF,dt} = \sum_{j=1}^{N} \sum_{h=1}^{c_j} \omega_{j^k} p_{Y_1,j^h} \left( 1 - \prod_{i=1}^{j-1} p_{X_0,i,j^h} \prod_{i=j+1}^{N} p_{X_0,i,j^h} \right) \tag{4.3}$$

The data-traffic that originates depends on the rate of notifying $\omega_{j^h}$ multiplied with the proportion of time the client is active as publisher and multiplied with the proportion of time there is at least one subscriber active at a CAFES-router that is not the router $j$ where the publisher is sending from (as this is free traffic). Also this summed for all possible clients.

In the next section we will give the results when both content-based routing mechanisms are compared in an exemplary setting.

## 4.3 Results, conclusions and discussion

For a comparison between flooding notifications and flooding subscriptions we took the following values for the parameters in the expressions derived in previous section:

| | |
|---|---|
| $\omega_{j^k} = 1 \ \ \forall j, h$ | On average 1 notification per minute |
| $r_{s,j^h} = 2 \ \ \forall j, h$ | A refresh-period of 2 minutes |
| $\lambda_{i,j^h} = 1/60 \ \ \forall i, j, h$ | Client as subscriber is on average 60 minutes inactive |
| $\gamma_{j^h} = 1/30 \ \ \forall j, h$ | Client as publisher is on average 30 minutes active |

Figure 4.2 shows the results where we compared both routing schemes by subtracting them from each other, $A_{NF} - A_{SF}$, and varying both $1/\nu$ (the average on-time of a client as publisher) and $1/\mu$ (the average on-time of a client as subscriber), where $\nu_{j^h} = \nu \ \ \forall j, h$ and $\mu_{i,j^h} = \mu \ \ \forall i, j, h$. The area in which the 3d-curve is above 0 means that notification flooding is worse than subscription flooding and below 0 otherwise around. So only when clients are quite active as publisher and very inactive as subscriber flooding of subscriptions would be better. This is just what intuition tells us.

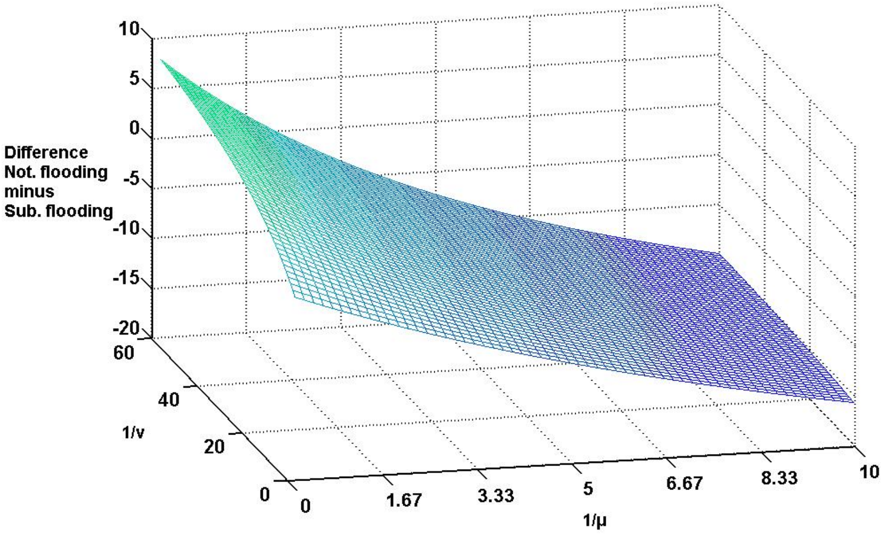


Figure 4.2: Flooding of notifications compared with flooding of subscriptions on activity period of publishers and subscribers

If we raise the value for $\omega$, the curve will lift in the left upper corner, which means that flooding subscriptions becomes slightly better in more situations. On the opposite, if we lower the value of $\omega$, the curve will drop in the left upper corner, i.e. flooding notifications will be advantaged in even more situations. The same can be applied for a varying $r$. If $r$ raises, the right part of the curve raises, which means that the difference between flooding subscriptions and flooding notifications becomes smaller (flooding subscriptions slightly better) and if $r$ drops, the differences grows and flooding notifications becomes better.

Remember we considered presence information as this information is intended to be implemented in the CAFES system. On base of the figure 4.2 it can be said that flooding notifications will be the best as it is assumed that clients acting as subscribers will be much longer active than say around 1 or 2 minutes in comparison with an inactive time of 60 minutes.

Possible extensions to the model can be made by the arguments and proposals we did in Chapter 2. For example the use of advertisements or the situation where more multicast networks are coupled through different CAFES-routers.

# Chapter 5

# Conclusions and recommendations

In the first section of this chapter the conclusions on this project are described. Next, in the second section the main recommendations for future research are given.

## 5.1 Conclusions

This research focused on answering one of the problems when considering to implement a publish/subscribe system in a networked environment. In more detail, we focused on the content-based routing schemes proposed to use in these systems and formulated the main research question as: *in what environments is a more sophisticated filtering algorithm superior to a simple flooding algorithm when total traffic has to be minimised?* For answering this question we derived an analytical model. At first we considered (1) a static situation, next we considered (2) a dynamic situation. Finally we used the model in the environment of (3) a radio-network and (4) CAFES-network. Considering these 4 parts, we derive the conclusions separately.

**1. Static situation**
In this situation an algorithm was derived which indicated what brokers in a regular hierarchical tree network ($m$-ary tree with $k$ levels) would be concerned with forwarding of both subscriptions and notifications. We assumed one publisher at the root of the network and subscribers at the bottom of the network. From this we were able to find a solution for the amount of messages caused by identity-based routing for three possible distributions of subscribers and compared these with flooding: a worst-case distribution, a best-case distribution and a homogeneous distribution. From the results in an exemplary setting we were able to

conclude the following. At lower activity then 30% of the subscribers, identity-based routing is always preferred. At higher activity then 48%, flooding is always preferred. However as it is likely that subscribers are dispersed a little better than the worst-case, e.g. according a homogeneous distribution, the break-even-point lies around 37% activity. Also, we were able to conclude that when subscribers are closer together the identity-based routing algorithm improves when compared with flooding.

**2. Dynamic situation**

In the next steps we extended the model by assuming the system in a more dynamic situation by considering the subscribers behaviour changing in time and model this by using queueing systems. We used the derivations by Jaeger and Mühl [6] as basis for our model and derived the following extensions:

- Instead of a steady state approach we considered a transient state approach were subscribers behaviour is possible to change in time.
- Jaeger and Mühl assumed uniform arrival and departure processes at each end-broker were subscribers are able to act. Besides, they assumed an infinite population of possible subscribers. We extended this by assuming different arrival and departure processes at each end-broker and also gave the model for a finite population of possible subscribers.
- We abandoned the regular $m$-ary tree with $k$ levels network and proposed a model for any tree network were subscribers and publishers are possible to locate at any broker.
- We extended the behaviour of publishers be assuming them acting as on/off sources and indicate their presence with the use of advertisements.
- Finally we proposed an idea how it is possible to add covering-based routing to the model and made some remarks on the modeling of roaming clients.

The most important parameters of the model include the arrival rate of subscribers $\lambda$, the departure rate $\mu$, the notification rate $\omega$, the re-subscription period $r$, the number of brokers $N$ and parameters for defining the network (e.g. $m$ and $k$ or other index-numbers).

In a comparison of a varying (non-homogeneous) arrival rate with a homogeneous arrival rate with respect to identity-based routing we found out that the non-homogeneous arrival process performs better for lower arrival rates, but becomes worse for higher arrival rates. This is caused by the impact of the toggle traffic. Furthermore we derived that more variation in the arrival process (to a certain point) is worse for identity-based routing. However, when the variation becomes larger and larger the difference between the non-homogeneous and homogeneous arrival processes converges to 0.

**3. Radio-network**
We derived a reference model for the radio-network to compare flooding with identity-based routing. We varied the parameters we described above within a range that is likely to appear in real when considering the dispersion of situational awareness information in the radio-network. In an exemplary setting we found that the break-even-point for the activity of subscribers lies around 12%. When more than 12% of the subscribers are active, flooding is preferred, when less than 12%, identity-based routing is preferred. By varying the parameters, $\omega$, $\mu$, $r$ and the distribution of the interest of subscribers (i.e. locally distinct interest than farther away from the publisher) this break-even-point will become higher when:

- The notification $\omega$ rate is higher.
- The period of being active as subscriber becomes longer (so $\mu$ lower).
- The re-subscription periods $r$ becomes longer.
- When locally the interest is higher than further away in the network.

**4. CAFES-network**
In the CAFES environment, we compared a flooding of subscriptions with a flooding of notifications. From the results we found that only when clients are quite active as publisher and very inactive as subscriber flooding of subscriptions would be better. In all other situations flooding of notifications is preferred. When considering the presence information that is proposed to use in the CAFES-environment, flooding of notifications will be best as it is assumed that the activity of clients is rather higher than quite low.

## 5.2 Recommendations

For modeling the content-based routing algorithms acting in a realistic environment we took a lot of assumptions to simplify reality. In this section we discuss what possible other options can be studied in the light of the research objective. For most of these, the model we derived can be used as a basic model.

- As performance indicator we took the amount of messages flowing through the network to indicate the bandwidth usage. We calculated this amount be considering that each message had a unit-size 1 and that the costs for flowing over 1 link exactly equals 1 unit of traffic. Furthermore we assumed that there was no delay when messages where propagated over the links. However, message sizes vary, lengths of links vary and delay occurs when messages are forwarded over the links. All these parameters have influence on the use of the bandwidth for both flooding as more sophisticated algorithms. So especially in networks where there are large differences in lengths of links, sizes of messages, delays and available bandwidth, it is interesting to investigate the impact of these differences when flooding is compared with more sophisticated routing algorithms.

- In this report we mainly focused on the identity-based routing algorithm as the more sophisticated algorithm and compared this with flooding. However, as discussed in the introduction, there are more different routing mechanisms, such as covering-based routing or merging-based routing. To give some insight, in paragraph 2.5.3 we proposed an idea to model covering-based routing within our model. For further research it is interesting to compare these different sophisticated routing schemes with each other. Also this can be done on base of the amount of messages flowing through the network. However, as these routing schemes make use of the routing tables, it is also interesting to compare them on base of the sizes of routing tables: in principle the smaller these are, the better the performance of the system.

- A third observation for a possible extension of the model is the use of buffers at certain points in the network. An example of the use of buffers is that information can be saved (to a certain degree) so clients who become active can be immediately notified without having to wait a moment in time a publisher notifies the (newest) information. In systems with roaming clients this can be a useful implementation, as the notifications that are sent to a broker at which a client was connected can be buffered and re-routed to the new broker the roaming client becomes connected. Depending on the roaming behaviour of clients its interesting to investigate the impact of re-routing traffic by using for example the identity-based routing algorithm.

- In our model we assumed that each publisher contained distinct information. It is also interesting to investigate the amount of messages that are flowing through the network when more publishers contain the same information.

# Bibliography

[1] Ayuba, K.C., *CAFES and Content-Based Routing*, Presence implementatie, Thales Communications BV, Huizen, 2007 (in Dutch).

[2] Baldoni, R., Beraldi, R., Tucci Piergiovanni, S. & Virgillito, A., On the modelling of publish/subscribe communication systems, *Concurrency Computat.: Pract. Exper.*, Volume 17, pages 14711495, 2005.

[3] Carzaniga, A., Rosenblum, D.S. & Wolf, A.L., Design and Evaluation of a Wide-Area Event Notification Service, *ACM Transactions on Computer Systems*, Volume 19, No. 3, pages 332- 383, 2001.

[4] Eugster, P.T., Felber, P.A., Guerraoui, R. & Kermarrec A., The Many Faces of Publish/Subscribe, *ACM Computing Surveys*, Volume 35, No. 2, pages 114-131, 2003.

[5] Fiege, L., Zeidler, A., Gartner, F.C. & Handurukande, S.B., Dealing with Uncertainty in Mobile Publish/Subscribe Middleware, *Proceedings of 1st International Workshop on Middleware for Pervasive and Ad-hoc Computing*, pages 60-67, 2003.

[6] Jaeger, M.A., & Mühl, G., Stochastic analysis and comparison of self-stabilizing routing algorithms for publish/subscribe systems, *The 13th IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2005)*, pages 471-479, Atlanta, Georgia, IEEE Press, 2005.

[7] Kurose, J.F. & Ross, K.W., *Computer Networking: A Top-Down Approach Featuring the Internet*, Addison-Wesley, Second edition, 2003.

[8] Massey, W.A., & Whitt, W., Networks of infinite-server queues with non-stationairy Poisson input, *Queueing Systems 13*, pages 183-250, 1993.

[9] Mühl, G., *Large-Scale Content-Based Publish/Subscribe Systems*, PhD thesis, Darmstadt University of Technology, 2002.

[10] Mühl, G., Fiege, L., Gärtner, F.C. & Buchmann, A., Evaluating Advanced Routing Algorithms for Content-Based Publish/Subscribe Systems, *The 10th IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2002)*, pages 167-176, IEEE Press, 2002.

[11] Mühl, G., Jaeger, M.A., Herrmann, K., Weis, T., Fiege, L. & Ulbrich, A., Self-stabilizing publish/subscribe systems: Algorithms and evaluation, *Proceedings of the European Conference on Parallel Computing (EuroPar 2005)*, Lisboa, Portugal, Springer, 2005.

[12] Segall, B. & Arnold, D., Elvin has left the building: A publish/subscribe notification service with quenching, *Proceedings of the Australian UNIX and Open System User Group Conference (AUUG 1997)*, Brisbane, Australia, 1997.

[13] Valk, K., de & Schilder, S.J.D., *Content-based routing*, Werking en toepasbaarheid, Thales Communications BV, Huizen, 2006 (in Dutch).

# Appendix A

# List of symbols

Most of the symbols that are used in the report.

| **Symbol** | **Description** |
|---|---|
| $a$ | Frequency of repeating in non-homogenous arrival function |
| $a(\mathbf{n})$ | Function indicating the amount of messages per time unit that originate in a state $\mathbf{n}$ |
| $A_f$ | The amount of messages caused by flooding per time-unit |
| $A_{ibr}$ | The amount of messages caused by identity-based routing per time-unit |
| $A_{sdt}$ | The amount of messages caused by state dependent traffic per time-unit |
| $A_t$ | The amount of messages caused by toggle traffic per time-unit |
| $b$ | Maximal amplitude of the wave in non-homogenous arrival function |
| $b(\mathbf{n}, \mathbf{n}')$ | Function indicating the amount of messages caused by a toggle |
| $B_{i,j}$ | Indicates if a broker at position $(i, j)$ its routing table is occupied or not |
| $\mathbf{B}$ | The matrix of all $B_{i,j}$ |
| $c$ | Average arrival rate used in non-homogenous arrival function |
| $d$ | Function indicating the average path-length an (un)subscription needs to travel |

| | |
|---|---|
| $k$ | Parameter indicating the depth of a regular hierarchical tree network |
| $l$ | Total number of links in a network |
| $L(\mathbf{B})$, $L(\mathbf{n})$ | Function that expresses the number of links that are used |
| $m$ | Parameter indicating the width of a regular hierarchical tree network |
| $n_i$ | State indicating the number of active clients at broker $i$ |
| $\mathbf{n}$ | State-vector of length $N$, denoting the number of active clients at each broker |
| $N$ | Number of brokers with possible subscribers |
| $N_i$ | Maximum number of possible clients at a broker $i$ |
| $P(X = n) = p_n$ | Probability distribution for a stochastic process $X$ of being in state $n$ |
| $q(\mathbf{n}, \mathbf{n}')$ | Transition rate |
| $r$ | Refresh-period of re-subscriptions |
| $S$ | Number of active subscribers |
| $t_r$ | Function indicating the toggle rate |
| $T$ | Length of integration period |
| $x$ | Function that expresses the proportion of time all links in the a network have to be traveled |
| $\lambda$ | Arrival rate |
| $\mu$ | Departure rate |
| $\omega$ | Notification rate |
| $\rho$ | Load of the system |

# Appendix B

# Algorithm for assigning the routing brokers

```
ALGORITHM ASSIGNING ROUTING BROKERS($B_{k-1,j}$)
initialize $i = k - 2$
start
Step 1. Is ($i \geq 0$)? If yes ⇒ step 2, if no ⇒ stop
Step 2. $j = 1$ ⇒ step 3
Step 3. Is ($j \leq m^i$)? If yes ⇒ step 4, if no ⇒ step 8
Step 4. $sum = 0$, $h = jm - m + 1$ ⇒ step 5
Step 5. Is ($h \leq jm$)? If yes ⇒ step 6a, if no ⇒ step 6b
Step 6a. $sum = sum + B_{i+1,h}$, $h = h + 1$ ⇒ step 5
Step 6b. Is ($sum > 0$)? If yes $B_{i,j} = 1$, if no $B_{i,j} = 0$, ⇒ step 7
Step 7. $j = j + 1$ ⇒ step 3
Step 8. $i = i - 1$ ⇒ step 1
```

There are three loops in this algorithm. The outer loop ($i$) goes from $k-2$ to 0 and considers each level of routing brokers in the three. The second loop ($j$) runs through all routing brokers in a level $i$ one by one and in the inner loop $h$ all children of the routing brokers at position $(i, j)$ are checked. If there is at least one of them active, the routing broker itself will be assigned active.

# Appendix C

# The radio-network: expressions for $x$, $d$ and $t_r$

On the basis of figure C.1 in the following two sections expressions for $x$, $d$ and $t_r$ are derived.

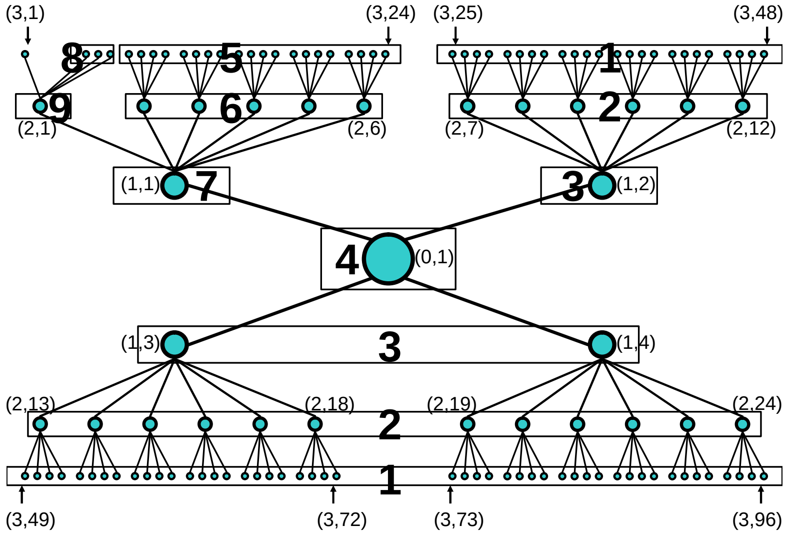


Figure C.1: Overview of the radio-network for the reference model in chapter 3

## C.1 Expression $x$

As we can see in the figure, we distinguish between 9 distinct positions of all nets when compared with the squad-net at $(3,1)$. We numbered them from 1 to 9. Number 1, for example, indicates all squads nets that are most far away from squad net $(3,1)$. For each distinct position with respect to the squad-net we are able to express the sum of the proportion of time the brokers are occupied with a routing entry from the steady state distribution 3.1, where $p_{i,j} = P(X_{i,j} = 0)$:

$$x_1 = \sum_{j_3=25}^{96} (1 - p_{3,j_3})$$

$$x_2 = \sum_{j_2=7}^{24} (1 - p_{2,j_2}) \prod_{j_3=24+4(j_2-6)-4+1}^{24+4(j_2-6)} (1 - p_{3,j_3})$$

$$x_3 = \sum_{j_1=2}^{4} (1 - p_{1,j_1}) \prod_{j_2=6+6(j_1-1)-6+1}^{6+6(j_1-1)} (1 - p_{2,j_2}) \prod_{j_3=24+24(j_1-1)-24+1}^{24+24(j_1-1)} (1 - p_{3,j_3})$$

$$x_4 = (1 - p_{0,1}) \prod_{j_1=2}^{4} (1 - p_{1,j_1}) \prod_{j_2=7}^{24} (1 - p_{2,j_2}) \prod_{j_3=25}^{96} (1 - p_{3,j_3})$$

$$x_5 = \sum_{j_3=5}^{24} (1 - p_{3,j_3})$$

$$x_6 = \sum_{j_2=2}^{6} (1 - p_{2,j_2}) \prod_{j_3=4+4(j_2-1)-4+1}^{4+4(j_2-1)} (1 - p_{3,j_3})$$

$$x_7 = (1 - p_{1,1}) \prod_{j_2=2}^{6} (1 - p_{2,j_2}) \prod_{j_3=5}^{24} (1 - p_{3,j_3}) x_4$$

$$x_8 = \sum_{j_3=2}^{4} (1 - p_{3,j_3})$$

$$x_9 = (1 - p_{2,1}) \prod_{j_3=2}^{4} (1 - p_{3,j_3}) x_7$$

$$x = \sum_{h=1}^{9} x_h \tag{C.1}$$

$$A_{sdt} = \left(\omega + \frac{1}{r}\right) x \tag{C.2}$$

## C.2 Expressions $d$ and $t_r$

Also for the toggle traffic we distinguish the 9 different positions with respect to the squad-net at $(3,1)$. For each of them we will multiply $d$, the average path-length with $t_r$ the toggle rate for all possible nets for each of the $h = 1, ..., 9$ positions in expressions $dt_h$:

$$\begin{aligned}
dt_1 = 2\sum_{i=25}^{96} \Bigg[ & 1 + p_{2,\lceil \frac{i}{4} \rceil} \prod_{j_3=4\left(\lceil \frac{i}{4} \rceil - 1\right)+1}^{i-1} p_{3,j_3} \prod_{j_3=i+1}^{4\lceil \frac{i}{4} \rceil} p_{3,j_3} + \\
& p_{1,\lceil \frac{i}{24} \rceil} \prod_{j_2=6\left(\lceil \frac{i}{24} \rceil - 1\right)+1}^{6\lceil \frac{i}{24} \rceil} p_{2,j_2} \prod_{j_3=24\left(\lceil \frac{i}{24} \rceil - 1\right)+1}^{i-1} p_{3,j_3} \prod_{j_3=i+1}^{24\lceil \frac{i}{24} \rceil} p_{3,j_3} + \\
& p_{0,1} \prod_{j_1=2}^{4} p_{1,j_1} \prod_{j_2=7}^{24} p_{2,j_2} \prod_{j_3=25}^{i-1} p_{3,j_3} \prod_{j_3=i+1}^{96} p_{3,j_3} + p_{0,1} \prod_{j_1=1}^{4} p_{1,j_1} \prod_{j_2=2}^{24} p_{2,j_2} \prod_{j_3=5}^{i-1} p_{3,j_3} \prod_{j_3=i+1}^{96} p_{3,j_3} + \\
& p_{0,1} \prod_{j_1=1}^{4} p_{1,j_1} \prod_{j_2=1}^{24} p_{2,j_2} \prod_{j_3=2}^{i-1} p_{3,j_3} \prod_{j_3=i+1}^{96} p_{3,j_3} \Bigg] \lambda_{3,i} p_{3,i}
\end{aligned}$$

$$\begin{aligned}
dt_2 = 2\sum_{i=7}^{24} \Bigg[ & \prod_{j_3=4(i-1)+1}^{4i} p_{3,j_3} + p_{1,\lceil \frac{i}{6} \rceil} \prod_{j_2=6\left(\lceil \frac{i}{6} \rceil - 1\right)+1}^{i-1} p_{2,j_2} \prod_{j_2=i+1}^{6\lceil \frac{i}{6} \rceil} p_{2,j_2} \prod_{j_3=24\left(\lceil \frac{i}{6} \rceil - 1\right)+1}^{24\lceil \frac{i}{6} \rceil} p_{3,j_3} + \\
& p_{0,1} \prod_{j_1=2}^{4} p_{1,j_1} \prod_{j_2=7}^{i-1} p_{2,j_2} \prod_{j_2=i+1}^{24} p_{2,j_2} \prod_{j_3=25}^{96} p_{3,j_3} + p_{0,1} \prod_{j_1=1}^{4} p_{1,j_1} \prod_{j_2=2}^{i-1} p_{2,j_2} \prod_{j_2=i+1}^{24} p_{2,j_2} \prod_{j_3=5}^{96} p_{3,j_3} + \\
& p_{0,1} \prod_{j_1=1}^{4} p_{1,j_1} \prod_{j_2=1}^{i-1} p_{2,j_2} \prod_{j_2=i+1}^{24} p_{2,j_2} \prod_{j_3=2}^{96} p_{3,j_3} \Bigg] \lambda_{2,i} p_{2,i}
\end{aligned}$$

$$d_3 = 2\sum_{i=2}^{4} \Bigg[ \prod_{j_2=6(i-1)+1}^{6i} p_{2,j_2} \prod_{j_3=24(i-1)+1}^{24i} p_{3,j_3} + p_{0,1} \prod_{j_1=2}^{i-1} p_{1,j_1} \prod_{j_1=i+1}^{4} p_{1,j_1} \prod_{j_2=7}^{24} p_{2,j_2} \prod_{j_3=25}^{96} p_{3,j_3} +$$

$$\left. p_{0,1}\prod_{j_1=1}^{i-1}p_{1,j_1}\prod_{j_1=i+1}^{4}p_{1,j_1}\prod_{j_2=2}^{24}p_{2,j_2}\prod_{j_3=5}^{96}p_{3,j_3}+p_{0,1}\prod_{j_1=1}^{i-1}p_{1,j_1}\prod_{j_1=i+1}^{4}p_{1,j_1}\prod_{j_2=1}^{24}p_{2,j_2}\prod_{j_3=2}^{96}p_{3,j_3}\right]\lambda_{1,i}p_{1,i}$$

$$\begin{aligned} dt_4 = 2\Bigg[&\prod_{j_1=2}^{4}p_{1,j_1}\prod_{j_2=7}^{24}p_{2,j_2}\prod_{j_3=25}^{96}p_{3,j_3}+\prod_{j_1=1}^{4}p_{1,j_1}\prod_{j_2=2}^{24}p_{2,j_2}\prod_{j_3=5}^{96}p_{3,j_3}+\\ &\prod_{j_1=1}^{4}p_{1,j_1}\prod_{j_2=1}^{24}p_{2,j_2}\prod_{j_3=2}^{96}p_{3,j_3}\Bigg]\lambda_{0,1}p_{0,1}\end{aligned}$$

$$dt_5 = 2\left[p_{0,1}\prod_{j_1=2}^{4}p_{1,j_1}\prod_{j_2=2}^{24}p_{2,j_2}\prod_{j_3=5}^{96}p_{3,j_3}+p_{0,1}\prod_{j_1=2}^{4}p_{1,j_1}\prod_{j_2=1}^{24}p_{2,j_2}\prod_{j_3=2}^{96}p_{3,j_3}\right]\lambda_{1,1}p_{1,1}$$

$$dt_6 = 2\left[p_{0,1}\prod_{j_1=1}^{4}p_{1,j_1}\prod_{j_2=2}^{24}p_{2,j_2}\prod_{j_3=2}^{96}p_{3,j_3}\right]\lambda_{2,1}p_{2,1}$$

$$\begin{aligned} dt_7 = 2\sum_{i=5}^{23}\Bigg[&1+p_{2,\lceil\frac{i}{4}\rceil}\prod_{j_3=4\left(\lceil\frac{i}{4}\rceil-1\right)+1}^{i-1}p_{3,j_3}\prod_{j_3=i+1}^{4\lceil\frac{i}{4}\rceil}p_{3,j_3}+\\ &p_{0,1}\prod_{j_1=2}^{4}p_{1,j_1}\prod_{j_2=2}^{24}p_{2,j_2}\prod_{j_3=5}^{i-1}p_{3,j_3}\prod_{j_3=i+1}^{96}p_{3,j_3}+p_{0,1}\prod_{j_1=1}^{4}p_{1,j_1}\prod_{j_2=1}^{24}p_{2,j_2}\prod_{j_3=2}^{i-1}p_{3,j_3}\prod_{j_3=i+1}^{96}p_{3,j_3}\Bigg]\lambda_{3,i}p_{3,i}\end{aligned}$$

$$\begin{aligned} dt_8 = 2\sum_{i=2}^{6}\Bigg[&\prod_{j_3=4(i-1)+1}^{4i}p_{3,j_3}+p_{0,1}\prod_{j_1=1}^{4}p_{1,j_1}\prod_{j_2=2}^{i-1}p_{2,j_2}\prod_{j_2=i+1}^{24}p_{2,j_2}\prod_{j_3=5}^{96}p_{3,j_3}+\\ &p_{0,1}\prod_{j_1=1}^{4}p_{1,j_1}\prod_{j_2=1}^{i-1}p_{2,j_2}\prod_{j_2=i+1}^{24}p_{2,j_2}\prod_{j_3=2}^{96}p_{3,j_3}\Bigg]\lambda_{2,i}p_{2,i}\end{aligned}$$

$$dt_9 = 2\sum_{i=2}^{4}\left[1+p_{0,1}\prod_{j_1=1}^{4}p_{1,j_1}\prod_{j_2=1}^{24}p_{2,j_2}\prod_{j_3=2}^{i-1}p_{3,j_3}\prod_{j_3=i+1}^{96}p_{3,j_3}\right]\lambda_{3,i}p_{3,i}$$

$$A_t = \sum_{h=1}^{9}dt_h \tag{C.3}$$