Switched multi-hop EDF networks –
the influence of offsets on real-time performance

Master’s Thesis in Computer System Engineering

Sha Mao Xuan & Xie Jun & Xu Xiao Lin
Switched multi-hop EDF networks –
The influence of offsets on real-time performance

Master Thesis in Computer System Engineering

School of Information Science, Computer and Electrical Engineering
Halmstad University
Box 823, S-301 18 Halmstad, Sweden

September 2011
Description of cover page picture/figure: a multi-hop network
Preface

We are really grateful to all the people who helped us during study period in Halmstad University, Sweden. Especially we would like to thank our supervisor Mattias, who patient and kind enough to help us on our way. It would not be possible to finish this thesis without his help. Also, we would like thank our head supervisor, Tony - thank you for providing us with useful information and discussions on the topic. We also want to thank our parents and friends – your help allowed us to pass all the difficulties during our hard work on the way to achieving the master degree.

Sha Mao Xuan & Xu Xiao Lin & Xie Jun
Halmstad University, September 2011
Abstract

In computer science, real-time research is an interesting topic. Nowadays real-time applications are close to us in our daily life. Skype, MSN, satellite communication, automation car and Ethernet are all things related to the real-time field. Many of our computer systems are also real-time, such as RT-Linux, Windows CE. In other words, we live in a “real-time” world. However, not everyone knows much about its existence. Hence, we chose this thesis in order to take a knowledge journey in the real-time field. For an average reader, we hope to provide some basic knowledge about real-time. For a computer science student, we will try to provide a discussion on switched multi-hop network with offsets, and the influence of offsets on real-time network performance. We try to prove that offsets provide networks of high predictability and utilization because offsets adjust packet’s sending time. A packet’s sending time is the time when a sender/router starts to transmit a date packet. Packets are sent one after the other. Therefore, we need to lower the time interval between one packet and another. Hence, in our network model, network performance is more predictable and effective. There might be some things left to discuss in future, so we would like to receive any advice and also suggestions for future discussions.
Switched multi-hop EDF networks --- the influence of offsets on real-time performance
Contents

PREFACE .............................................................................................................................. 1
ABSTRACT ............................................................................................................................ 2
CONTENTS ........................................................................................................................... 4
LIST OF FIGURES ............................................................................................................... 6
LIST OF TABLES .................................................................................................................. 7
1 INTRODUCTION ................................................................................................................. 8
2 PROBLEM STATEMENT ..................................................................................................... 11
3 RELATED WORK ............................................................................................................... 12
  3.1 REAL-TIME SYSTEM ................................................................................................. 12
  3.2 ETHERNET ................................................................................................................ 22
  3.3 NETWORK TOPOLOGY .............................................................................................. 28
  3.4 REAL-TIME CHANNELS ............................................................................................ 32
4 APPROACH ...................................................................................................................... 35
  4.1 PERIOD TRAFFIC VERSUS NON-PERIOD TRAFFIC ANALYSIS ............................... 35
  4.2 DEADLINE PARTITIONING ....................................................................................... 36
  4.3 EXPERIMENT ANALYSIS ......................................................................................... 37
  4.4 NETWORK TOPOLOGY .............................................................................................. 37
  4.5 ALGORITHM ............................................................................................................... 38
5 EXPERIMENT RESULTS ................................................................................................. 41
6 CONCLUSIONS ................................................................................................................. 48
7 REFERENCES .................................................................................................................... 49
List of Figures

Figure 1.1. Simple network model................................................................. 9
Figure 3.1. (a) CBR, (b) VBR(on/off) and (c) VBR........................................ 17
Figure 3.2. (a) linear bus topology, (b) star topology and (c) mesh topology........ 29
Figure 3.3. Tree topology.............................................................................. 30
Figure 3.4. Full binary tree ........................................................................... 31
Figure 3.5. set up of the real-time channel.................................................... 33
Figure 4.1. Traffic model parameters.............................................................. 35
Figure 4.2. Deadline partitioning.................................................................. 36
Figure 4.3. An upward buffer link is blue link and a downward buffer link is red link... 38
Figure 4.4. The flow of transmitting packet in switch .................................. 39
Figure 4.5. Delay in multi-hop network.......................................................... 40
Figure 5.1. offset defers deadline ................................................................. 42
Figure 5.2. Comparison of maximum delay for the experiments.................... 42
Figure 5.3. Comparison of average delay for the experiments........................ 44
Figure 5.4. Comparison of 64 nodes deadline miss ratio for the experiments..... 45
Figure 5.5. Comparison of 128 nodes deadline miss ratio for the experiments... 46
List of Tables

Table 3.1. QoS requirement........................................................................................................21
Table 3.2. QoS priority level .........................................................................................................21
Table 5.1. Comparison of maximum delay for the experiments.........................................................43
Table 5.2. Comparison of average delay for the experiments............................................................44
Table 5.3. Comparison of average delay for the experiments............................................................46
1 Introduction

With the rapid development of the Internet technology, it is obvious that the internet became a popular trend very quickly, but at the same time has grown to be more complex during the last ten years [1] [12]. In our daily life, real-time communication became more and more common to all. There are thousands of real-time software applications that we use every day. For example, real-time on-line chat applications, multi-video conferencing applications and on-line games. These applications have the same requirement; video and audio data should be shown to both parties (the sender and the receiver) on time. In other words, packets from a sender need to arrive at a receiver on time. It is acceptable to have a short video or audio delay between the sender and the receiver. From the perspective of a service provider, we have to offer the cheapest price as well as guarantee qualitative real-time communication. In the last decade, scientists and network experts have already begun to research the real-time system field. Based upon their achievements, we have attempted to research the influence of packet offsets on multi-hop network performance. A packet offset is used to delay a packet’s transmit time and deadline. For example, packet A begins to transmit at time t1 and packet B also begin to transmit at time t1. Both packet A and B need to arrive at the same receiver. Moreover, they have same characteristics of data. By adding offset o1 to packet B, packet B will start to transmit at time t1+ o1. Hence, packet A and B will send to the receiver one after another. In this case, we eliminate collisions, which occur during this real-time communication phase. This research focuses on the question of how to handle real-time traffic by having different offsets. Since our thesis focuses on soft real-time systems, especially on a real-time network such as Ethernet, we will start our discussion by introducing a simple real-time communication model.
Figure 1.1. Simple network model

Figure 1.1 shows three nodes: A, B and C - connected with a switch with linkages a, b and c. The real-time channel is established between nodes A, C and nodes B, C. Nodes A and B have the same global time, which is t. Links a, b and c have the same link length and bandwidth. All of them have the same packet generation rate, packet size, operating system delay and other parameters. At same time nodes A and B have established a real-time channel with node C. There are two channels in this model - Channel AC and Channel BC. They start to generate real-time traffic to node C simultaneously. Packets from both A and B nodes will pass through the switch; they will be stored and forwarded. The queuing method used is the EDF algorithm. The packets from A and B have the same packet size, period and deadline, so the packets’ period and deadline are equal.

In this case, packets from node A and B will arrive at switch at same time. Also, a receiver/switch will randomly choose the packet from the input buffer to the output queue, because they have all the same characterizes. Therefore, it is possible for the switch to choose only one side packet, for example channel AC’s packets, to pass through. However, to chose only channel AC’s packet to pass through, will affect
Switched multi-hop EDF networks --- the influence of offsets on real-time performance

channel BC’s performance. As such, the packet lost and retransmission rate might rise in this case, since two real-time channel’s packets affect each other in the switch’s buffer. Hence, we plan to use offsets to adjust transmitting start time of node A and B.

We organized the thesis as follows: In section 2, we discuss the problem we need to test by adding time offset in a soft real-time system. In section 3, we introduce basic real-time system knowledge and other related topic about this thesis. In section 4, we discuss the assumptions, definitions and algorithms which the offset account in the calculation. In section 5, simulation flow chart and experiments is discussed. In section 6, we described the conclusion. In section 7, we attached thereferences.
2 Problem Statement

In this research, we program a MATLAB real time network simulation to test the influence of packet offsets in multi-hop networks’ performance, which is packet average delay, maximum packet delay and end-to-end delay. We research into the soft real-time packet-level communication domain. In soft real-time packet-level communication, the application does not need to guarantee the packets reach the destination on time. Offset is the key in this research. We assume that offset provides a predictable network, because offset offers a more accurately prediction on where and when a packet in the network will exist. Hence, we hope the network analysis tool or conclusion could be useful for future network improvement.

In order to prove this assumption, we use simulation, mainly because simulation is cost-effective to get a result in a short time. We try to keep the simulation simple but reasonable. We construct a real-time tree topology network. The network includes switches, end-nodes and links. Each switch has several output buffer queues. When the switch receives packet, it will transmit packet into different output buffer according to its destination. Switch receives the packet from the source node and sends to destination node according to the real time scheduling algorithm. We use the EDF scheduling algorithm to decide the sending order of the packets in the queue. EDF scheduling algorithm is implemented in all end-nodes and switches. The end-node generates packet in a fixed rate. Each packet will send to the destination at the generation time. The packet order is adjusted by the offset and we arrange the packets’ interval time as tightly as possible. In this case, we improve the network performance.
3 Related work

This thesis focuses on switched multi-hop EDF networks and the influence of offsets on real-time performance. Hence, we will discuss five key words from the title - switched networks, multi-hop, EDF, offsets and real-time. Switch network can be also called a packet switch network or switching packet network. This technology is used as the opposite to circuit switching. Multi-hop means the number of nodes should be more than one. EDF is a form of scheduling policy, which is used to handle packets in the end-nodes and switches. Offset is used to affect the packet’s response time and deadline during packet transmission; it is the main parameter which we will examine.

Real-time system is a system which needs time constraints for both communication parties. If we analyze the five key words listed above, we also construct Ethernet-based real-time communication - it can be categorized into two main sub-categories: Ethernet and real-time systems. These two areas will be the main research directions in our thesis.

In order to provide a deeper understanding of this thesis area to our readers, we see the need to explain these two areas separately in the part that describes the related work. It will not only help in understanding of the topic, but also will give valuable insights to the reader, who wants to expand his/her own knowledge base. Based upon this, we will talk about real-time systems and Ethernet in the following two subsections.

3.1 Real-time system

The term “real-time” represents systems that are ‘fast enough’. Real-time systems can accomplish a system function in the appointed time and deliver synchronous or
asynchronous responses. Time restriction is the main parameter of real-time systems. Each task or event has to be finished in the limited time allocate. However, the need for a task to be finished in a limited time does not mean that it is to be finished as soon as possible. For example, in a packet-switched network, the job or packet sending procedure should finish within a certain time period; otherwise the packet will arrive to the switchestoo faster, which causes the switch to fail at handling the arrived packets. If the number of packets arriving at the switch exceeds its handling capability of packets, the excess packets will be dropped.

In a packet-switched network, there usually is a soft real-time system, where a packet is usually organized as a sequence of data, and having only one or two packets dropped will not cause serious damage to the whole system, application or environment. Assuming such a circumstance, in a system that is a video, audio or chat application, losing a packet will only cause a frame stop, which means that the picture or sound will be stopped for a short period of time. It will not affect the whole conversation between parties, since even if people miss a frame or second of sound, they can still continue with their conversation. In contrast to the packet-switched network example, another real-time system example that is automation control area; a packet that misses its deadline might cause a serious problem, such as a car accident, and could possibly cause injuries and the death of people.

By comparing packet-switched networks with automation controls, it becomes clear that real-time systems are actually defined by their intended usage and different purpose; users usually have different requirements for the real-time systems. In a packet-switched network, the user needs packets to arrive on time, but packets can arrive earlier or later. Early arrived may create a problem, known as “buffer overflow”, which depends on buffer size and the number of packets arriving at the same time. Late arrives will cause picture or sound loss; however, in automation control, real-time systems, packets have to arrive to the node just on time. Thus, it can
clearly see that a real-time system is better at delivering packets, and it guarantees their arrived to be on time.

To differentiate between two types of real-time systems, they are divided into hard real-time and soft real-time types. A packet-switched network is a soft real-time system. An automation control system is a hard real-time system. We will separately describe these types of real-time systems in order to understand the difference between hard real-time systems and soft real-time systems.

**Hard real-time system and industrial communications**

Hard real-time systems (another name is immediate real-time systems) have strict time deadlines, strong guarantees of QOS [14] [16]. The system correctness is not only present in logical correctness, but also in time correctness. The key point for hard real-time systems is whether the signal arrives at the end note on time or not. If a packet missed deadline, which occurred in hard real-time system, then critical failure must be displayed.

Five examples of hard real-time systems are shown below:

- Car engine control system
- Medical system (e.g., heart pace makers)
- Industrial controllers
- Embedded systems (e.g., robotics)
- Nuclear Power plants

We can find hard real-time system implementations not only in software applications, but also in hardware applications. Time in these five examples of systems is the most important criteria. A packet missing a deadline in such systems will cause serious consequences. In car engine control systems, missing a deadline can cause the breakage of a car engine. It may cause a car accident if the vehicle is being
Switched multi-hop EDF networks --- the influence of offsets on real-time performance

driven at the time [15]. Hence, such a system’s predictability is much more important than its performance [13].

Soft real-time system and industrial communications

Soft real-time systems are the key topic we are going to discuss in this thesis [16], since soft real-time systems present such systems as:

- Live video-audio conference applications
- Air flight control plan for commercial systems
- Automatic teller machine
- Online games

In such systems, it is acceptable to miss a deadline, since the system could continue to operate, even if they missed a deadline. For example, live video-audio conferences can tolerate late arrives, but QOS will be decreased. (Communicating parties can accept incoherence of the picture and/or sound, since even one or two words are missing, it won’t affect the whole conversation.)

Real-time traffic

Real-time traffic is a packet stream which has a time constraint property [1]. It is a data-oriented service. With the rapid development of networks, they have high-speed fiber and a diverse set of services. The best-effort-delivery network is not enough for our research anymore [12]. Hence, we need a predictable “quality-of-service” network.

➢ Best-effort versus guarantee performance

Best-effort-delivery and guarantee performance delivery are always discussed in the area of real-time networks. In real-time systems, best-effort-delivery networks present the setup where a message always tries to reach the destination node as soon as possible [12] [14] [15]. However, in this case, network utilization is not considered, as well as buffer space, end-to-end delay and so on. If we do not take this network quality requirement into account, a best-effort-delivery network could not provide a
network suitable for the current requirements of the research. Those who have some idea of computer networks surely know that today they are far more complex than ten years ago [16]. Higher network speeds, new types of network architecture, new connection methods, and mixed network environments have been introduced. A network that is only providing a fast enough speed is not good enough anymore. A network also needs to be aware of the limitation of the routers’ buffer space. For example, fast elaboration of packets that exceed the networks’ capability will result in dropping the following packets because the buffer space will potentially be full. Hence, there is no available data space to store packets. Thus, the destination node has to request these dropped messages to be sent again. However, retransmitting messages will cause message delays; in this case, if this is a hard real-time system, the system might fail. If it is a soft real-time system, it could still running, but we are certain that message delays caused by retransmissions will cause a decrease in the performance of the system. Now, we can see that there is a need to have some sort of performance guarantee in different kinds of networks.

Networks with guarantee performance try to meet different requirements of QoS. A network’s requirement of QoS is defined by the user or application. For example, in the case of video conferencing and online gaming, if we want to maintain a smooth video conference, the message delay must not be too long. From a user’s perspective, one or two seconds delay might be acceptable. Comparing this with online gaming experience, delays could be a little bit longer. For instance, in the game, if you want to send a message to another player, the message might arrive significantly later, for instance, 1 or 2 minutes. So, from a user’s perspective, even 10 minutes or more minutes is acceptable. Hence, our conclusion is that applications or users define the requirements for the kind of performance needed. Let us show a simple example of what we are going to guarantee. As we mentioned above, sending messages quicker than the network capability will cause packet loss. If we use protocols to set a channel,
(we will talk about channels a bit later), during the establishment phase, we will set the sending rate, thus the node will send packets in a reasonable rate.

➢ Traffic Model

In [12], the author suggests three types of traffic models which are defined by different kinds of applications. These two main types of traffic are the Constant Bit Rate model (CBR) and the Variable Bit Rate (VBR). Variable bit rate model can be sub-categorized as an on/off source model and periodic with variable packet model. Three types of model are shown below:

![Traffic Models Diagram](image)

**Figure 3.1. (a) CBR, (b) VBR(on/off) and (c) VBR**

Figure 3.1: (a) presents applications which have regular intervals, such as radar systems. If the data input is not continuous, then this type of model represents a hard real-time system model. In this type of model, data has to be received continuously, and if data is missing only for a short period of time, the whole system (e.g. radar system) could not make the right calculation. Figure 3.1 (b) presents these applications like voice application. Because of human speech, the talk will be stop for a short while. This phase we call a silence phase. Figure 3.1 (c) presents video traffic. When we decode and encode video frames, sometimes the packet varies in size because the decoding rate or encoding rate is not always a constant value. Video applications also could be presented as a CBR model, if we adjust the applications’ encode and decode rate. In other words, we control the encoding and decoding rate to keep it in balance.

We can make the traffic to become a CBR model
Packet Scheduling

In this paper, we assume each node is connected by several incoming and outgoing links. Each out port has one or more ongoing queues [13] [17]. We use a queuing algorithm to decide which queue is served. In this paper, the queuing algorithm will be presented by the EDF algorithm. However, the algorithm can also be implemented by other scheduling algorithms. Scheduling algorithms can be categorized to priority and no-priority base scheduling types. Priority scheduling can be subcategorized to dynamic and static scheduling [14]. For example, an EDF algorithm is a dynamic scheduling algorithm, while a rate monotonic algorithm is a static scheduling algorithm. Dynamic scheduling is a preemptive scheduling algorithm; static scheduling is a non-preemptive scheduling algorithm. In this paper, the message is split to several packets. During the run-time scheduling phase, a shorter deadline message can interrupt a longer deadline message [15]. But since the message is considered on a packet level, if a packet is being transmitted, but it is not finished yet, preemptive scheduling will cause packet loss, thus we will wait until this packet finishes its transmission and then the next packet gets the network resource. In this thesis we use the EDF algorithm to schedule the packets, hence we will introduce the EDF algorithm below and another two famous algorithms as well, such as FCFS and Rate monotonic algorithm.

EDF algorithm is a preemptive dynamic scheduling technique. It sets earliest deadlines to higher priority. Priority could be changed dynamically, because a new packet arriving at the switch will affect the deadline order. FCFS scheduling is also a common scheduling algorithm used in commercial routers. It is also studied by another partner group in their FCFS-related thesis. FCFS uses time to decide priority. The message that comes earlier gets higher priority. Rate-monotonic scheduling is a static-priority scheduling. Priority is decided by the job, and shorter jobs have a higher priority for execution.
The difference between dynamic scheduling and static scheduling is that with dynamic scheduling packet priority could be changed when a new packet arrives. However, in the static scheduling, the priority is assigned before the task starts running. Also, Liu and Layland have proved that EDF algorithm is the optimal preemptive scheduling method. Hence, in our thesis, we will use the EDF algorithm to perform our research.

- **Policing**
  Policing is used to monitor real-time traffic. If the traffic does not behave as guaranteed, then policing will adjust its sending rate or stop the sending procedure [1] [2]. For example, in the real-time network, if packet bit rate is larger than it is promised to be, it may cause buffer overflow. By using policing we can work against the malicious end system [4].

- **Error, Flow Control and Buffer Management**
  Error, flow control and buffer management are commonly used in real-time communication. Errors are used to check whether the transmitted data is wrong or not [1] [2] [4]. If it is, then we need to retransmit the wrong packets. However, this mechanism may cause a great increase of delay for the network. Because retransmitting a packet does not only bring the cost of resources in the network, such as bandwidth, it also affects the previous packets or new packets. Once the error mechanism is used, and transmission was made, it is impossible to avoid the great increase in latency. Flow control is used to control the end node traffic, when it becomes over-transmitted. For example let us take a case when a real-time communication channel is established between two parties. First, they agree on the QOS, promised by both parties, and then the real-time communication starts. If one end node is going to send data that is larger than the agreed size, the flow control mechanism will be used in the receiver to control the sender, which tries to send more
data. Typically, we can find this happening in video transmitting when the decode rate is not equal to the encode rate between the sender and the receiver. It is also used to control the buffer overflow. Buffer management is used to manage the buffer during real-time communication.

➢ Admission Control

Admission control is a switch mechanism. It is used to calculate network resources [1] [2] [4]. The main purpose is to meet QOS requirements. If a network could be used for a new channel, the resources will be reserved for the new channel.

➢ Quality of Service

Quality of Service (QOS) is widely used in real-time communication. It is used to reserve network resources, such as throughput, rather than providing different levels of service quality [1] [2] [4]. Real-time communication networks use QOS to guarantee the specific bit rate, delay, jitter and other criteria during the call establishment phase. In contrast to the best-effort network that provides the network throughput as fast as possible, real-time communication networks make the network more predictable by using QOS mechanisms. Real-time applications, such as VoIP, prefer a more predictable network because resource from predictable networks can be effectively used. People always mix QOS with high level performance of networks. However, QOS can be viewed from several key performance metrics, such as throughput, number of dropped packets, delay (latency), jitter and error. These metrics are chosen under human and technical factors. Stability of service, availability, delays, user information isconsidered as human factors. Reliability, scalability, effectiveness, maintainability and grade of service are considered as technical factors.
Switched multi-hop EDF networks --- the influence of offsets on real-time performance

<table>
<thead>
<tr>
<th>Application name</th>
<th>Require metric</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Streaming multimedia</td>
<td>Guaranteed throughput</td>
<td>Ensure minim level of quality</td>
</tr>
<tr>
<td>IP telephony/Voice Over IP</td>
<td>Strict limit on jitter and delay</td>
<td>Clearly communication</td>
</tr>
<tr>
<td>Video teleconferencing</td>
<td>Low jitter and latency</td>
<td>Fluent video</td>
</tr>
</tbody>
</table>

Intable 3.1, we can see that video and audio both require limits on jitter and delay. However, the requirement varies, since IP telephony requires strict limits, video teleconferencing has lower requirements on the level of jitter and latency. QOS’s metrics are chosen by different real-time applications and are used differently. The difference is also used to create different QOS priority levels, thus during per-packet processing, packets can enter different queues according to different priority levels.

The table of QOS priority levels is shown below:

<table>
<thead>
<tr>
<th>Priority Level</th>
<th>Traffic Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (lowest)</td>
<td>Best Effort</td>
</tr>
<tr>
<td>1</td>
<td>Background</td>
</tr>
<tr>
<td>2</td>
<td>Standard (Spare)</td>
</tr>
<tr>
<td>3</td>
<td>Excellent Load (Business Critical)</td>
</tr>
<tr>
<td>4</td>
<td>Controlled Load (Streaming Multimedia)</td>
</tr>
<tr>
<td>5</td>
<td>Voice and Video (Interactive Media and Voice) [less than 100ms latency and jitter]</td>
</tr>
<tr>
<td>6</td>
<td>Layer 3 Network Control Reserved Traffic [less than 10ms latency and jitter]</td>
</tr>
<tr>
<td>7</td>
<td>Layer 2 Network Control Reserved Traffic [lowest latency and jitter]</td>
</tr>
</tbody>
</table>

Table 3.2 shows QoS priority level. It is defined in 8 levels. From the top to the bottom, the quality of service is increasing. In level 0, the network performance is defined as a best effort network. Level 3 and 4 are defined as business critical and streaming multimedia levels respectively. In levels 5 to 7, network performance differs by the amount of time of latency and jitter. The different levels are used in packet processing in order to separate different levels of real-time traffic. We will talk about this in the subsection devoted to packet processing.
3.2 Ethernet

With the development of computer and network technologies, the technology control area has been under huge changes. Now Ethernet is one of the most widely used technologies and most common communication protocol standards of local area networks (LAN). Ethernet is a kind of LAN technology, it is always located in a building and the connection distance of the equipment is short. In the past, the longest Ethernet cable between the devices was only several hundred meters long, since it is not used to connect geographically dispersed locations. But with the current rapid development of the technology, the Ethernet connection distance is upgraded and today people have been able to create Ethernet networks several kilometers away. Such fields as industrial automation and process control areas have more and more Ethernet applications used in them. Ethernet was invented in 1973, and at first it was not used for real-time communication. The advantage of Ethernet is its popularity, low cost and high performance and all these are still good for real-time applications.

➢ The transmission mode of the Ethernet

There are two kinds of transmission types of Ethernet. One is the half-duplex, another is full-duplex. Traditional shared LAN works in the half duplex mode. At the same time, the data can only be transferred in a single direction. When two directional transmissions of data are needed, a conflict will occur. This will reduce the efficiency of Ethernet.

Full-duplex transmission is used for point-to-point transmissions; it supports two directional transfers, because it uses two separate twisted pair lines. In our thesis, we will use an Ethernet mode with full duplex – it needs two independent CPUs, one to control the uplink of the transmission, another will be the control for the downlink of the transmission.
The principle of the Ethernet

Ethernet uses Carrier Sense Multiple Access method with Collision Detection (CDMA/CD). Ethernet is a broadcast network.

The Ethernet works as follows

When a host is ready to transmit data in the Ethernet, it follows these steps:

- Monitor whether there is a signal transmitting. If a signal is transmitting, it shows that the channel is in busy state and continues to monitor until the channel is free.
- If there is no signal - then transmit the data.
- When data is being transmitted, it has to keep monitoring. If collision occurs, step1 needs to be executed after a while.
- It succeeds if there is no conflict.

Shared Ethernet

In the early stages of Ethernet history, multiple nodes of Ethernet share the same transmission medium, which is called shared Ethernet - it uses broadcast communication between nodes. Shared Ethernet uses Carrier Sense Multiple Access/Collision Detection (CSMA/CD) technology to prevent conflicts - the CSMA/CD method is the one that the sender suspends when a collision was detected, it will then send it again after a random delay until success is achieved. Since delay time is random and cannot be known in advance, the uncertainty in the time of response in shared Ethernet is the main limitation for the real-time application.

Collision:

With Ethers, a conflict will occur when two data frames post messages to the physical transmission media at the same time. When a conflict occurs, the data on the physical segment will not be effective anymore.
Switched multi-hop EDF networks --- the influence of offsets on real-time performance

- Collision domain:
  Every node can receive the frame that was sent by another node in the same collision domain.

- The factors influence the conflict produce
  Conflict is an important factor, which affects the performance of Ethernet, since if conflicts occur in more than 40% of the cases, the efficiency will be decreased. There are many factors that produce conflicts. For example: the more nodes are present in the same collision domain, the more conflicts occur. In addition, such things as data packet length (the maximum Ethernet frame length is 1518 bytes), the diameter of the network and other factors also affect the appearance of conflicts. Therefore, when the size of the Ethernet is increasing, measures must be taken to control the spread of conflicts. The usual approach is to use bridges and switches to divide the network in order to divide a large number of collision domains into several smaller ones.

- The disadvantage of the shared Ethernet
  Since all the nodes are connected to the same collision domains, irrespective of where the frame comes from or where will it go, all nodes can receive it. With the increase of the number of nodes, a larger number of conflicts will cause network performance to decrease rapidly.

**Switched Ethernet**

Switched Ethernet eliminates the collision problem of the CSMA/CD mechanism. With switched Ethernet, the switch determines to which port of the switch the data frame should be sent to, according to the MAC address from the received data. Because the transmissions of frames between ports shield each other, this can eliminate the collision problem of the traditional Ethernet.

Compared with Shared Ethernet, the advantages of the Switched Ethernet are:

- Reduced conflict rate: the switch eliminates the number of collisions in each
port (each port is a collision domain), it also avoids conflicts from spreading.

- Improved Bandwidth: Each node of the switch can use the whole bandwidth, rather than each node sharing the bandwidth. It guarantees bandwidth for each node.

Each port of the Switched Ethernet connects to the host directly and generally works in a full duplex mode. The Switched Ethernet is largely supported by the automation industry because of its high bandwidth in real-time communication. Industrial Switched Ethernet is used for real-time data transmission in complex industrial environments.

There are three kinds of swap modes of the Switched Ethernet. They are cut through, store and forward, and fragment free. Store and forward is one of most widely used technologies in the area of computer networks. The store and forward method is one in which the data frame is completely stored and checked before it is transferred to the destination. If everything is correct, it is then sent to the destination. It can eliminate the presence of faulty data and enhance the utilization of bandwidth.

In industrial environments, Switched Ethernet can be used to decrease cost, as well as increase efficiency. Moreover, it increases network bandwidth and provides network determinism for industrial control applications.

**Ethernet for the industrial automation**

Ethernet technology is inexpensive and stable. As one of the most popular communication networks, Ethernet technology provides high speed communications, varied hardware and software products, a wide range of applications and mature support technology. In recent years, with the development of network technology, Ethernet enters into a control area. Thus new types of Ethernet control network technologies emerged. Setting up an open and transparent communication protocol becomes a tendency. This is mainly for the reason that industrial automation systems are developed for distribution and intelligent control. Ethernet technology enters into
the industrial control field, its technical advantages are clear: interconnection can be
easily achieved. It forms an integrated enterprise-wide control of open networks. The
cost of hardware and software is low, and the communication rate is high.

**EtherCat**

EtherCAT is an open real-time network communication protocol. It was research and
development by the Beck off Automation GmbH. EtherCAT sets a new standard for
real-time performance and flexible topology. At the same time, it even decreases the
use cost of the profibus.

Based on Ethernet technology, automation has a lot of advantages. It is low cost
and open; it can perform remote control and easily integrate with the management.
These advantages have been used for successful automation. By using EtherCAT
technology, we do not need to receive Ethernet data packets at each connection node
and then decode and copy the data as data is processed. When the frames go through
each device, the EtherCAT can read the most important data for this device from the
station control unit. EtherCAT protocol is optimized for processing data; it is sent to
Ethernet frames directly or compressed into UDP/IP data packets. Ethernet frames
may contain several EtherCAT packets; each packet dedicated to a specific storage
area, and the storage area can be up to 4GB of logical process size image. EtherCAT
network performance has reached a new level. The update time of 1000 distributed
I/O of data it only needs $30\mu s$, it includes the sub-datagram. This kind of network
performance advantage is obvious in small controllers which have medium computing
ability. EtherCAT support linear, tree and star topologies. Actually EtherCAT
supports almost all of topology types. Therefore, the bus-shaped structure of fieldbus
can be used for Ethernet.


**Profibus**

Profibus is an industry data bus and developed rapidly in recent years. It is an all digital, bidirectional and multi-communication system of connection between the intelligent field devices and automation systems. The advantages of the profibus are shown below.

- A pair of twisted-pair cables can connect to a number of control equipment; do it is convenient for reducing installation costs.
- Reducing the maintenance cost.
- Improvement of system reliability.
- It provides flexible service for users.
- Continue to develop and improve the low-speed field bus.

**ATM**

ATM stands for Asynchronous Transfer Mode. It is a cell-based packet switching and multiplexing mode, which is designed for a variety of general business connection-oriented transfer modes. It is suitable for LAN and WAN, which have high-speed data transfer rates and support many types of data such as voice, data, fax, and real-time video, CD quality audio and video communication.

- The principle of the ATM
  
  When the transmitting end wants to communicate with the receiving end, it will send a control signal about applying for the connection through the network. The receiving end receives this control signal and agrees to establish a connection - then a virtual circuit will be established.
3.3 Network topology

With the network development, network management in network environment is more and more important. As a good network management system, the first step is to get acquainted with the network’s topology, and then it can configure the network device, performance test and fault diagnosis effectively.

Network topology means that computers and other devices are connected through some methods. There are two kinds of network topology.

- Logical topology: displays connections between two network devices, based on the IP address of network devices.
- Physical topology: it is the real physical connection of networks. It includes routers and router connections, router and switch connections, switches and switch connections.

Network topology shows the network server, the configuration of work station and the connections between each other. The main topologies of networks are known as the linear bus topology, star topology, tree topology and mesh topology.
Switche multi-hop EDF networks --- the influence of offsets on real-time performance

**Linear bus topology, star topology and mesh topology**

As shown in the diagram below, this is linear bus topology. From the figure, there is only one cable connecting all devices and this cable is named the backbone.

![Diagram showing linear bus topology, star topology, and mesh topology]

Figure 3.2. (a) linear bus topology, (b) star topology and (c) mesh topology

Figure 3.2 (a) show linear bus topology. The advantages and disadvantages of the linear bus topology are shown below.

- **Advantages:** It is easy to install. Since each node is sharing a bus as a data path, the channel utilization is high.
- **Disadvantages:** Because of the channel sharing, it’s not suitable for connecting too many nodes. The fault of the bus will cause the breakage of the whole network.

Figure 3.2 (b) shows star topology. The star topology network has a central node, the other nodes are connected to the central node directly; it is also called a centralized network.

- **Advantages:** It is the simple structure; ease of managing, control is simple, easy to build, network latency small, and low transmission error.
- **Disadvantages:** It is high cost, low reliability, and resource sharing capabilities are poor.

Figure 3.2 (c) shows mesh topology. This topology refers to interconnection of nodes by transmission lines; each node has to be connected to at least two other nodes.
Switched multi-hop EDF networks --- the influence of offsets on real-time performance

- **Advantages:** Network reliability is high. There are two or more paths between arbitrary two nodes in the network. If one path broken, it can use another path to send the information to the destination. The network can be formed into various shapes, using a variety of communication channels, different data rates. It’s easy to share the resource in such a network. It can choose the best path to the destination, so the transmission delay is low.

- **Disadvantages:** The network topology is complex – it is hard to control. Connection lines are expensive – it is not easy to expand.

- The linear bus topology, star topology, mesh topology are also very good network topology. Different topology have their different own pros and cons. We can use these topologies as our network topology in the future work.

**Tree topology**

We chose the tree topology as our network topology model. It has evolved from the linear bus topology. The tree topology can look like a combination of the linear bus topology and the star topology, and it is suitable for broadcasting.

In the tree topology, the root node (the highest node in the topology) has no parent node. The other nodes only have one parent node. The leaf node (the lowest node in the topology) has no child nodes. Each node might have one or many child nodes except the leaf node.

![Figure 3.3. Tree topology](image-url)
Figure 3.3 shows the advantage of the tree topology that is easy to expand, which makes fault isolation easier. The tree topology also has its disadvantages. It has too much dependence on the root node; if the root node fails, then the whole network will be broken.

There is a special tree topology which is named the full binary tree. In this topology, each node only has two child nodes (left child node and right child node) except the leaf node. The number of the node is \(2^N - 1\) (N is level). The example is shown in figure 3.4.

![Figure 3.4. Full binary tree](image)

Figure 3.4 shows the packet will go through the parent node from any node to any node. Each node has its own number. The route map between the senders to the receivers can be known in advance. There is a route map stored in each switch. Switches can use the route map to decide packet’s path in order to choose which output queue to forward packets to. In our thesis, we use full binary tree as our network topology. Each node can look like a switch or node in the network. The tree topology is suitable for the LAN and it is used widely.
In the tree topology, it’s convenient to change the network topology easily, which is very important to our thesis, as can simulate a different number of switches to test the offset.

### 3.4 Real-time channels

First, we will discuss the mechanisms needed to support real-time communication. Real-time communication is described below [2] [3]: first, the end-node sends a channel request to the router, then the router calculates the link utilization and decides whether this request can be accepted or not. After the real-time channel is accepted, the end-nodes agree on the same performance metrics. During run-time, we use policing, packet scheduling, error control, flow control and buffer management to adjust and control network performance [5] [18].

To model packet deadlines, Ferrari and Verma proposed “real-time channels” [3]. A real-time channel is a logical link between two nodes. It is a unidirectional virtual circuit which is established for application level messages in a multi-hop network and guarantees timely delivery of messages [3] [16]. When an application needs to deliver a set of packets with stringent timing constraints it sends a request of establishing a real-time channel to the multi-hop network [3] [17].

The steps for requesting the establishment of a real-time channel as seen in figure 3.5 is as follows:

1. **Step1:**
   Node A wants to send a message to node C through two switches. In this case we should select a route for the channel first, the route between these two nodes consists of links 1, 2, and 3. Once the channel is established, all the packets will be sent from the source to destination through this route.
Step 2:
The real-time switch calculates the resource demand required from the end node. Also, it uses a simple schedulability test to check whether the channel deadline can be guaranteed or not. It checks whether the requested resources are available to establish a real-time channel. These resources include link bandwidth, buffer space, and packet processing capability. The main purpose for this test is to make sure that the new channel will not affect existing channels. If the channel deadline can be guaranteed, the channel will be accepted and the real-time switch will forward the real-time traffic. [5]

Step 3:
If the schedulability test is passed, the logical real-time channel can be set up between the source and destination. If not, it rejects the real-time channel guaranteeing that other existing real-time channels are not influenced. In our thesis, we assume that if a packet misses its deadline, it will be discarded in the switch.

The setup of real-time channels is the basic element for real-time communication since it leads to a certain real-time guarantee between two end-nodes. The real-time switch calculates the resource which is possible to be used for the new real-time
Switted multi-hop EDF networks --- the influence of offsets on real-time performance

channel. Also it ensures that the exiting channel is not affected by the new channel. Because the exiting channel is already using the link bandwidth, real-time switch buffer space, and CPU processing capability, in order to make sure that the performance is being guaranteed, the set up real-time channel phase is a key phase in the whole the real-time communication. During this set up channel step, the two calling parties will decide guarantee performance and how to arrange the real-time switch resource. The two end-nodes will also decide the sending rate and so on, so all the network performance is calculated or guaranteed before it really starts to transmit. Real-time uses regulation control to protect the end node from over-sending the data to the real-time switch, thus it protects the packet from being lost because the number of packets is over the switch’s buffer space. Actually, we require from the end node to declare its traffic characteristics. Here we have a list of some basic parameters that are to be considered during the set up of the real-time channel phase.

- The minimum packet interval on the channel/ packet sending rate
- The maximum packet size
- The maximum service time in the switch for the channel’s packets
- The maximum packet loss rate
- Delay bound

These are only the basic parameters that we drew here. However, during a real real-time channel establishment phase, these parameters are not enough. Other tests also need to be calculated, such as the deterministic test, statistical test and delay bound test. All these tests have to be passed by the entire node along the real-time channel node. If at least one test failed in the intermediate node, the channel won’t be set up. If all the tests are completed successfully, the real-time channel can be set up, and then it is time to consider the way we are going to schedule the packets.
4 Approach

4.1 Period traffic versus Non-period traffic analysis

In the traffic model, we introduce three types of traffic models which are researched in [12]. These three types of traffic modes present three categories of applications. CBR is a hard real-time system application. VBR is a soft real-time application. They can be categorized as an on/off model and variable packet size model. They present only audio and audio-and-video soft real-time systems respectively. In the Figure 4.1 below, we characterize three types of traffic model parameters:

![Figure 4.1. Traffic model parameters](image)

Figure 4.1 introduce a common real-time analysis model. The difference in this picture is that each period is a deadline for first packets and also is a start time for following packets. Execution time is the transmit time in each link and switch. Execution time in the link is a constant and each link will be equal. However, execution time in the switch needs to count the waiting time in the buffer. The time that a packet needs to wait depends on the other packet, which arrives via the same route. It depends on their priority and it will be selected by the EDF algorithm, thus the execution time for a packet in the router is a sum of the transmission time and the waiting time. Relative deadlines are equal to each period and are decided by the
sending rate of the node. This should not exceed the guaranteed sending rate. It should not exceed the guaranteed sending rate otherwise the intermediate node’s buffer will overflow.

- **Arrived time** $a$: is the time at which a message or a packet arrives to any node in the network.
- **Response time** $R$: is the time which a message or a packet needs to wait in order to transmit in the network, which is executed by the CPU.
- **Worst case execution time/Computation time** $c$: is the maximum length of time the packet transmits in the queue or link.
- **Start time** $s$: is the time at which a message starts its processing.
- **Finish time** $f$: is the time at which a message finishes its processing.
- **Period** $p$: is the time at which the end node transmits a new packet.
- **Relative deadline** $d$: is the time at which a process stops its execution. We assume that it represents an end-to-end deadline in our research.
- **Absolute deadline** $D$: is the time at which a message stops its execution even if it misses its deadline.

### 4.2 Deadline partitioning

![Figure 4.2. Deadline partitioning](image-url)
Figure 4.2 shows the node a sends a packet to node b, node a’s application defined absolute deadline $D$ for the packet, and this deadline is separate to $T1$, $T2$, $T3$ and $T4$. During the call establishment phase, we need to calculate whether sum of $T1$, $T2$, $T3$ and $T4$ is equal to $D$. If the result is larger than $D$, then it means that the delay bound test has failed and the call could not be established.

Deadline partitioning [10] divides the user specified deadline by the number of hops or end-to-end connections [14] [15]. The formula is shown below.

$$D \text{ (absolute deadline) = T1 \text{ (end to end deadline) + T2 + T3 + T4}}$$

In each end-to-end deadline test, schedulability test will also be made. The entire test has to pass, and if successful the call will be established.

### 4.3 Experiment analysis

The goal of the experiment is to research the soft real-time performance. We will work with a multi-hop network with EDF - this is a 100$Mb/s$ full duplex network. The number of nodes in this switched network is set to 64 nodes and 128 nodes. We will set the same period, deadline and packet size. They are 2000 $\mu$s, 2000 $\mu$s and 1630 Bytes. We set the offset to 500$\mu$s and 1000$\mu$s. We run the simulation 100 times and set 10 time points every time. At each time point we will record data: packet average delay, maximum packet delay, and the deadline-miss ratio, and then use this data to compare the average delay, maximum delay and the deadline-miss ratio.

### 4.4 Network topology

We decided to adopt a binary tree topology as the experiments topology, where each node has two or more leave nodes. We will set the depth to 6 or to 7 - this means that the number of nodes is 64 and 128. The buffer in the switches of this topology will be
of two types - one is an upward buffer, another one is a downward buffer. An example
buffer is shown in figure 4.3.

![Diagram of a multi-hop network with upward and downward buffers](image)

**Figure 4.3.** An upward buffer link is blue link and a downward buffer link is red link

In the upward buffer, the packet delay only depends on packets from lower
layers in the tree. The packet delay in a downward buffer depends on the packets from
upward buffers and downward buffers in the tree, as seen in Figure 4.3.

### 4.5 Algorithm

A multi-hop network stores and forwards packets and uses an EDF algorithm to examine a packet from the source node to the destination node and how big the delay will be in the worst-cases. We assume that the network is collision-free and that all links are of the full-duplex type. The buffer in the switch is assumed to be big enough to avoid buffer overflow.
When a packet transmits into a switch, it comes into buffer first through the interface. Then it enters into queue and is sequenced with other packets according to their deadline. After queuing up, it is conveyed to the output port and is delivered to the input interface of the next switch through link. Then it does this process again. Figure 4.4 shows the flow of transmitting packet into a switch.

![Figure 4.4. The flow of transmitting packet in switch](image)

In the transmission process, from the source node to the destination node, we will assume some parameters:

- $X$ = switch delay
- $P$ = delay for link
- $Tr$ = transmission delay for each hop
- $\Delta$ = offset
- $T$ = local time
- $C$ = queue delay

When a packet passes through the switch, there is a queuing delay ($X$). The switch will check whether any packets will miss their deadline. If the packet will miss its deadline, the switch will automatically add an offset to the packet. The offset can only be added for each packet once in the same switch. If the packet does not miss its deadline, the switch will not add the offset to the packet. There is a propagation delay.
(P) and a transmission delay for each hop (Tr). After packets arrive in the queue, there is a queuing delay (C). Supposing that the number of switches is n, then the packet’s arrived time is: \((n+1) \cdot (Tr + P) + n \cdot (X + C + A)\). As shown in the figure 4.5.

![Figure 4.5. Delay in multi-hop network](image)
5 Experiment results

In this section, we want to get the network’s performance by testing different offsets and assumption parameters - How offsets influence the network performance in different networks.

The experiments are repeated with the same packets’ period, deadline, packet’s size, and simulation time under different network topologies - different number of leaf nodes and offsets. The packets’ period, and deadline are 2000 μs. The packets’ size is 1630 bytes. We set simulation running time to 100000 μs.

After adding offset, each packet’s response time is increased. Each packet’s delay, from source node to destination node, also is increased. If a packet misses its deadline, the simulator will add an offset to the packet. In the simulator, this means that a packet was waiting in a switch’s input queue at that moment. In the switch’s input queue, a packet’s offset is included in its queuing delay, because a packet’s queuing delay is always longer than the offset which we add. A packet’s offset does not defer its execution time, but it defers its deadline. For example, in figure 5.1, packet B will run after packet A finished. The shadow part denotes packet B’s execution time. After adding offset, the queuing delay is decreased but B’s execution time is not changed. Hence, the offset is used to defer packet B’s deadline. We conjecture that the packets’ deadline miss ratio will decreased.
In addition, the offset defers the packet’s release time. When we calculate the average delay and maximum delay, only arrived packets are calculated in the results. So the average delay and maximum delay might be decreased.

Figure 5.2 represents the comparison of maximum delay between 64 nodes set and 128 nodes set. There are 6 curves are shown on the figure. The data analysis for each curve is explained in the Table 5.1 below.
Table 5.1. Comparison of maximum delay for the experiments

<table>
<thead>
<tr>
<th>Maximum</th>
<th>offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>0 μs</td>
</tr>
<tr>
<td>64 nodes</td>
<td>6225 μs</td>
</tr>
<tr>
<td>128 nodes</td>
<td>9525 μs</td>
</tr>
</tbody>
</table>

In Table 5.1, 6 results are explained. There are 64 nodes with 0 offset, 64 nodes with 500 offset, 64 nodes with 1000 offset, 128 nodes with 0 offset, 128 nodes with 500 offset and 128 nodes with 1000 offset.

In Table 5.1, the 64 nodes set, compared with the case without offset, the maximum delay decreases from 6225 μs to 5975 μs. The decrease of the maximum delay is 4%. When offset is 1000 μs, the decreases of maximum delay is 3.75%.

In Table 5.2, the 128 nodes set, compared with the cases without offset, the maximum delay decreases from 9525 μs to 8325 μs. The decrease of the maximum delay is 12.6%. When offset is 1000 μs, the decreases of maximum delay is 11.29%.
Figure 5.3. Comparison of average delay for the experiments

Figure 5.3 represents the comparison of average delay between 64 nodes set and 128 nodes set. There are 6 curves are shown on the figure. The date analysis for each curve is explained in the Table 5.2 below.

Table 5.2. Comparison of average delay for the experiments

<table>
<thead>
<tr>
<th>Average Delay</th>
<th>offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 nodes</td>
<td>0 μs</td>
</tr>
<tr>
<td>2714 μs</td>
<td>2578 μs</td>
</tr>
<tr>
<td>128 nodes</td>
<td>4355 μs</td>
</tr>
</tbody>
</table>

In Table 5.1, 6 results are explained. There are 64 nodes with 0 offset, 64 nodes with 500 offset, 64 nodes with 1000 offset, 128 nodes with 0 offset, 128 nodes with 500 offset and 128 nodes with 1000 offset.
In Table 5.2, the 64 nodes set, compared with the case without offset, the average delay of the case decreases from 2714μs to 2569 μs. The decrease of the average delay is 5%. When offset is 1000μs, the decreases of average delay is 5.34%.

In Table 5.2, the 128 nodes set, compared with the cases without offset, the average delay of the case decreases from 4355 to 3378. The increase of the average delay is 0.53%. When offset is 1000μs, the decreases of average delay is 20.39%.

![Figure 5.4. Comparison of 64 nodes deadline miss ratio for the experiments](image)
Switched multi-hop EDF networks --- the influence of offsets on real-time performance

Figure 5.5. Comparison of 128 nodes deadline miss ratio for the experiments

Figure 5.4 represents the comparison of deadline miss rate for the 64 nodes set experiment. Figure 5.5 represents the comparison of deadline miss rate for the 128 nodes set experiment. There are 3 curves shown on each figure. The data analysis for each curve is explained in the Table 5.3

Table 5.3. Comparison of average delay for the experiments

<table>
<thead>
<tr>
<th>Deadline</th>
<th>offset</th>
<th>0 μs</th>
<th>500 μs</th>
<th>1000 μs</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 nodes</td>
<td>1.2%</td>
<td>0.116%</td>
<td>0.153%</td>
<td></td>
</tr>
<tr>
<td>128 nodes</td>
<td>30.9%</td>
<td>1.384%</td>
<td>1.375%</td>
<td></td>
</tr>
</tbody>
</table>

In Table 5.3, 6 results are explained. There are 64 nodes with 0 offset, 64 nodes with 500 offset, 64 nodes with 1000 offset, 128 nodes with 0 offset, 128 nodes with 500 offset and 128 nodes with 1000 offset.
In Table 5.3, the 64 nodes set, compared with the case without offset, the deadline miss ratio decreases from 1.2% to 0.116%. The decrease of the deadline miss ratio is 1.08%. When offset is 1000μs, the decreases of deadline miss ratio is 1.05%.

In Table 5.3, the 128 nodes set, compared with the cases without offset, the deadline miss ratio decreases from 30.9% to 1.384%. The decrease of the deadline miss ratio is 29.52%. When offset is 1000μs, the decreases of average delay, maximum delay and deadline miss ratio is 20.39%, 11.29% and 29.53%.

The results of experiment accord with our conjecture. Within a period of time, the average delay, maximum delay and deadline miss ratio decrease.
6 Conclusions

This research’s main topic is the influence of offsets on the performance of a real-time network by simulating an experiment on a packet level, where the offsets are allowed to be used in each switch. The most important parameters recorded are the average delay and maximum delay that were affected by the offsets. We used an EDF scheduling algorithm to calculate the priority of each packet, and then the switches are fixed. The values of parameters that were set by μs in the experiment are similar to a real life scenario. We have used various different values for offset and the number of switches in order to run the simulation that was tested off-line. With the help of the simulation, we can see the results for maximum delay and average delay, as well as the deadline-miss ratio. We tested our setup with a different number of nodes, 64 nodes or 128 nodes. We have compared networks that had different offsets, or without offsets. The different values of the offsets are equal to 0 μs, a quarter of the period and a half of the period. The result of the simulation showed that offsets and the period affect the average and maximum delays, as well as deadline-miss ratio. We have set an equal offset to each packet and after adding the offset, the average delay, maximum delay and deadline-miss ratio have decreased. The larger the offset is; the lower the average delay is. At the same time if the offset is larger, the maximum delay is larger as well. Concerning the deadline-miss ratio, the effect with a smaller offset being place is better than with larger one correspondingly. So in a network of a different kind, it is important to select an appropriate value for the offset since it will affect the delay and deadline-miss rate considerably.

There are still many things need to be done in future. We can build a network of a different topology to test the average and the maximum delay. For example, one can test a star topology, ring topology or others type.
7 References


Switched multi-hop EDF networks --- the influence of offsets on real-time performance


