Software V.18 Modem Standard Negotiation

Mika Kallijärvi

Luleå University of Technology
MSc Programmes in Engineering
Department of Computer Science and Electrical Engineering
Division of Signal Processing
Software V.18 modem standard negotiation

Mika Kallijärvi

Luleå University of Technology
Department of Computer Science and Electrical Engineering
Division of Signal Processing

May, 2005
During the recent years, the telecommunication infrastructure has been moving slowly from analog techniques towards IP (Internet Protocol) and other digital formats. Along with this change, a new concern, packet loss, has arisen among the earlier problems during text telephone communication. During voice transmission this is not a problem, but in text transmission it leads to bit errors and corrupted characters.

One solution to this problem is to change the analog text telephone tones to text, and send it with VoIP (Voice over IP). The characters sent in this fashion can be changed back to the same text telephone tones as transmitted or into some other text telephone standard, if needed. This decoding/encoding of analog- to digital signals is done in a gateway.

The purpose of this project is to investigate if it is possible to implement a V.18 modem in Java. This project is divided in two parts: detection of the incoming text telephone signals standard, and decoding of the incoming signals. The part considered in this master thesis is the signal detection part.

From the test results for the V.18 modem, we make the conclusion that it is possible to implement a V.18 modem in Java. The different V.18 sub standards, implemented in the modem, can be detected, and the project has resulted in an implementation of a V.18 modem in Java.
This present document is a master degree project by Mika Kallijärvi at Master of Science Program in Electrical Engineering at Luleå University of Technology. It was carried out at Omnitor AB in Luleå during the summer and autumn of 2004.

I would like to thank my supervisor Andreas Piirimets at Omnitor AB for help with Java and for feedback on this report, Gunnar Hellström at Omnitor AB, and my examiner Dr. James P. LeBlanc at Luleå University of Technology. I would also like to thank Johan Carlsson at Luleå University of Technology for helping me out with LATEX, he has also made the template used for this document. Last but not least, I would like to thank my family for supporting me in everything I do.

This work is part of an ongoing collaborative effort to promote mainstreaming and functional enhancements toward telecommunications access for all. The work was partially funded by the National Institute on Disability and Rehabilitation Research, US Dept of Education under Grant H133E040013 as part of a co-operation between the Telecommunication Access Rehabilitation Engineering Research Center of the University of Wisconsin -Trace Center joint with Gallaudet University, and Omnitor. The opinions herein are those of the authors and not necessarily those of the funding agencies.

Luleå, May 2005

Mika Kallijärvi
2.6 Signal energy .............................................. 16

CHAPTER 3: IMPLEMENTATION

3.1 General design ........................................... 17
3.2 Identification of frequency for incoming signal .............. 20
  3.2.1 Selection of band-pass filter .......................... 21
3.3 Identification of bit rate for incoming signal .................. 23
  3.3.1 Asynchronous demodulation of FSK modulated signal ....... 23
  3.3.2 Bit rate determination ............................ 24
3.4 Java implementation ...................................... 25
  3.4.1 Matlab implementation ............................... 26
  3.4.2 Java design in general .............................. 26
  3.4.3 Design of java classes ............................. 26
3.5 Test scheme .............................................. 28
  3.5.1 Test scheme for originating mode ....................... 28
  3.5.2 Test scheme for answering mode ....................... 31
3.6 Problems .............................................. 35
  3.6.1 JMF .............................................. 35
  3.6.2 V.8 negotiation signals ................................ 35
  3.6.3 Ans and ANSam ..................................... 36
  3.6.4 Transmission of Baudot and EDT ....................... 36

CHAPTER 4: RESULTS ........................................ 37

CHAPTER 5: CONCLUSIONS ...................................... 39

CHAPTER 6: FURTHER WORK .................................... 41

APPENDIX A: BLOCK-DIAGRAMS OVER AUTOMOD ORIGINATING AND ANSWERING MODES .............................. 43
CHAPTER 1

Introduction

1.1 Background

During the recent years, the telecommunication infrastructure has been moving slowly from analog techniques towards IP (Internet Protocol) and other digital formats. Along with this change, a new concern, packet loss, has arisen among the earlier problems like over compression and poor quality text telephone signal. During transmission of voice, these packet losses are not critical, and the gaps introduced can be smoothed over with techniques that provide a transition from the last good packet received to the next good packet received. When sending digital data, for example some text telephone standard tones, the techniques used to fix the packet losses can actually result in bit errors that leads to corrupted characters. Protocols like TCP/IP automatically checks for missing packets and request replacement packets if needed. This results in no packets lost but causes delays, which is not a problem in data transmission, but is unacceptable in speech transmission. Protocols included in VoIP (Voice over IP) will not try to replace lost packets or wait for delayed ones; instead the gaps will be smoother out.

Many different competing solutions, to this problem that is introduced in text telephone communication, are proposed by different standards groups. One solution, that is championed by Gunnar Helström at Omnitor AB, is to change the analog text telephone tones to text, T140 (an IP text standard), and send it with VoIP. The characters sent in this fashion can be changed back to the same text telephone tones as transmitted or into some other text telephone standard tones, if needed.

This approach solves also another major issue in the area of access to telecommunication by individuals who are deaf, which is text telephone migration. By changing the analog signals to digital format, it is possible for people with only computers, that do not have access to text telephone, to be able to communicate with people that use only
text telephones without having to have extra hardware. This way those who do not want to migrate from old analog techniques to new digital techniques, are still able to communicate with each other.

1.1.1 T-hybrid

The project T-hybrid is a research project cooperation between Omnitor AB and the University of Wisconsin. The purpose of the project is to make a gateway (between analog and digital telephony) for text telephones in IP telephony traffic, which decodes analog text telephone signals to pure text and vice versa. This is a new strategy that has yet never been tested.

1.2 Purpose

The purpose of this project is to investigate if it is possible to implement a V.18 modem in Java, that translates (encodes and decodes) RTP (Real Time Protocol) sound streams, carrying analog text telephone signals, to pure text (digital format), and vice versa.

1.3 Objectives

The V.18 modem shall be implemented as an independent (open source) java package (plug-in), so that it can be used in other context, in a communication application called Sipcon1. This application (Sipcon1) is developed at Omnitor AB, as a reference implementation for IP text telephony protocols, such as RFC4103.

A figure over the test environment for the project, including Sipcon1, Cisco ATA 188, and text telephone Textlink 9100, can be studied in Figure 1.1. The Cisco ATA 188 is used to encode the analog telephone audio signal from the text telephone (Textlink 9100) to G.711 μ-Law 8 bit compressed PCM (Pulse Code Modulation) samples, at a sampling rate of 8000 samples/second. In the computer, JMF (Java Media Framework) is used by Sipcon1 to decompress the incoming G.711 μ-Law encoded samples. The protocol used for transportation of the data is RTP. Information regarding JMF, RTP, and Sipcon1 can be found in [1].

This project, implementation of the V.18 modem, is split up into two different parts. One part of the project contains the communication fraction of the modem, that is detection (identification) of incoming text telephone signals and transformation of bit streams, containing text coded with a specific V.18 substandard (see [2] or 2.1 for all sub standards supported by V.18), to V.18 substandard specific tones. The other part of the project contains decoding of incoming signals, with V.18 substandard specific tones, to pure text, and encoding of outgoing text to bit streams, coded with a specific V.18 substandard.
The part considered in this report (master thesis) is the communication part of the V.18 modem. The objective is to implement a detector that detects what V.18 substandard is used for the text telephone communication session. Primarily the modem shall be able to detect V.21 and Baudot, which are two sub standards of V.18. The final goal is to be able to detect all the sub standards supported by V.18. The implementation objectives are divided into primary- and secondary objectives.

Information and objectives regarding the decoder/encoder part of the project can be acquired through Omnitor AB.

1.3.1 Primary objectives

- To implement V.21 and Baudot to the V.18 modem.

1.3.2 Secondary objectives

- To implement V.8/V.18(text telephone mode) communication signals to the V.18 modem.

- To implement EDT, Bell 103, DTMF, and V.23 to the V.18 modem.

1.4 Delimitations

- The decoder and encoder part of the modem shall not be implemented. This fraction of the V.18 modem is acquired through another master degree project at Omnitor AB.

- No consideration shall be taken to optimize the program code, in order to increase the performance of the program, and decrease the amount of clock cycles needed to run it.
1.5 Omnitor

Omnitor AB is a Swedish company that is a leading actor in the accessible information technology area. The company is a mixture of hearing, hearing-impaired and deaf people that have a common vision: Information technology and telecommunication shall be accessible nationally and internationally for people with disabilities.

The main area of work is deaf people’s communication, and since 1995 (when the company was founded) Omnitor has been involved in different developments and projects, such as:

- Establishing a quality assessment method for evaluating of the performance of sign language transmission through video communication
- Establishing an international standard for text telephony ITU-T V.18, compatible with all old national standards
- Establishing international standards for Total Conversation including video, text, and voice
- Developing a PC based Total Conversation Terminal for video, text, and voice conversation. It was the first user terminal to include harmonizing standards for communication in video, text, and voice.
2.1 ITU-T recommendation V.18 version 3

2.1.1 Overview

This recommendation describes modem procedures to automatically be able to communicate with different types of text telephones. The recommendation specifies signal analysis, signal transmission, and logic needed to determine what kind of text telephone is connected to the line. It also describes the required steps to be able to communicate with a standard, that is supported by DCE:s (Data Communication Equipment, e.g. modem, text telephone, etc.).

The recommendation also specifies transmission of identification signals, which makes it possible to determine when the connection is between two DCE:s, that support V.18. The modulation used for this purpose is V.21 modulation. The different text telephone standards, that are supported by this recommendation are: EDT, Baudot, DTMF, V.21, V.23, Bell 103 and different V.18-based devices.

The recommendation defines three different modes for the DCE:s operating in text telephone mode: originating-, answering-, and automoding mode. In originating mode the DCE transmits V.18 identification signals, and V.23 stimulations signals until a recognized text telephone signal is received, and a connection is established. In answering mode the DCE try to stimulate to connection by sending probing signals for different types of text telephones, while listening for valid text telephone signals. An automode monitor mode is also provided to be able to have a voice telephone on the same line as the text telephone.

The following three paragraphs below include some basic information regarding these different modes, for more information about the subject the reader is referred to [2].
2.1.2 Automoding originating

When the DCE is connected to the line, it shall not transmit any signals during the first second. After the silent period the DCE shall begin transmission of the V.18 identification signal CI. When three CI signals have been sent, the DCE shall be silent for 2 seconds and after that send XCI signal (for more information regarding how CI and XCI signals are defined see [2] and [3]). The identification sequence shall be sent until one of the signals below has been received, or a timer has expired. The identification signals transmission sequence looks in summary as follow:

<table>
<thead>
<tr>
<th>Signal</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silence</td>
<td>1 s</td>
</tr>
<tr>
<td>CI</td>
<td>400 ms</td>
</tr>
<tr>
<td>Silence</td>
<td>2 s</td>
</tr>
<tr>
<td>CI</td>
<td>400 ms</td>
</tr>
<tr>
<td>Silence</td>
<td>2 s</td>
</tr>
<tr>
<td>XCI</td>
<td>3 s</td>
</tr>
<tr>
<td>Silence</td>
<td>1 s</td>
</tr>
<tr>
<td>CI</td>
<td>400 ms</td>
</tr>
<tr>
<td>Silence</td>
<td>2 s</td>
</tr>
<tr>
<td>etc.</td>
<td></td>
</tr>
</tbody>
</table>

The DCE:s receiver shall be able to detect the following signals:

- modulated 2100 Hz ANSam as defined in [3];

- 2100 Hz ANS as defined in [4];

- 2225 Hz;

- 1300 Hz;

- 1650 Hz;

- 1400 or 1800 Hz;

- DTMF tones;

- 980 or 1180 Hz;

- 1270 Hz;

- 390 Hz;
2.1.3 Automodding answering

In answering mode the DCE:s receiver should, after connecting to the line, be able to detected the following signals:

- V.23 high-band signals;
- 1300 Hz;
- 1400 or 1800 Hz;
- DTMF tones;
- 980 or 1180 Hz;
- CI signal;
- 2100 Hz;
- modulated 2100 Hz ANSam as defined in [3];
- 1270 Hz;
- 2225 Hz;
- 1650 Hz;

2.1.4 Automodding monitor mode

This mode shall be implemented for the purpose of detection of text telephone connection attempts from voice mode, and for use in automatic voice/text answering systems. Automode functions similar to answering mode with minor modifications, see [3] for more specific information.

2.1.5 Probing mode

For nordic countries (Sweden, Finland, Iceland, Norway, and Denmark) the suggested probing sequence for automoding is:

send V.21 carrier
send DTMF buffered message
send Baudot buffered message
send EDT buffered message
send V.23 carrier
send Bell 103 carrier

Suggested probing sequences for other countries can be found in [2].
2.1.6 Baudot

The communication channel for Baudot is half-duplex (it is not possible to transmit data in both directions simultaneously). A carrier is sent 150 ms before the first character is transmitted. After transmission of a character, the receiver shall be disabled for 300 ms to mitigate false detection of echoes.

The modulation used is FSK modulation using 1400 Hz (±5%) for binary 1 and 1800 Hz (±5%) for binary 0. The nominal data signalling rate is 50 or 45.45 bits/s.

2.1.7 DTMF

The communication channel is half-duplex, and the receiver shall be disabled for 300 ms after transmission of a character, to mitigate false detection of echoes.

The DCE shall detect characters at least for 40 ms with a silence period of at least for 40 ms. The DTMF characters shall be transmitter at least for 70 ms with a silence period of at least 50 ms.

2.1.8 EDT

The communication channel used for EDT is half-duplex. A carrier is sent 300 ms before transmission of the first character. The receiver shall be disabled for 300 ms after transmission of a character, to mitigate false detection of echoes.

The modulation for EDT is FSK modulation, and the signal is sent using [5] low-band frequencies. The data signalling rate is 110 bits/s.

2.1.9 Bell 103

The communication channel for data transmission is full-duplex, which make it possible to transmit data in both directions simultaneously, at a data signalling rate of 300 bits/s or less.

The modulation used is a binary frequency shift modulation, resulting in a modulation rate equal to the data signalling rate.

The nominal mean frequency for channel 1 and channel 2 is 1170 Hz and 2125 Hz respectively. The frequency deviation is ±100 Hz, and the higher characteristic frequency (FA) \textsuperscript{1} in each channel corresponds to a binary 1, and the lower (FZ) corresponds to binary 0.

2.1.10 V.23 Videotex terminals

There are two main types of Videotex terminals in use for text telephone communication, Minitel and Prestel. The modulation used is asymmetric duplex, with a data signalling rate of

\textsuperscript{1}Channel 1: FA = 1270 Hz, FZ = 1070 Hz; Channel 2: FA = 2225 Hz, FZ = 2025 Hz
rate of 1200 bits/s for forward channel, and 75 bits/s for backward channel. For more information about Videotex terminals, the reader is referred to [2] and [6].

2.1.11 V.21

The communication channel for V.21 has a data signalling rate of 300 bits/s, and is full-duplex (data transmission is possible in both directions simultaneously).

The modulation used is FSK modulation with continuous carrier, according to [5] frequencies.

2.1.11.1 Channel selection

Existing text telephones use different methods to select the mode of operation (originate or answering). The two most common ways of doing this is:

1. The DCE starts in answering mode, and toggles randomly at intervals of 0.6 - 2.4 s between the two modes, until a carrier connection is reached.

2. The DCE uses stored information, and chooses mode depending on if it recently made a call or received a call.

2.1.12 V.18 Text telephone mode

The modulation in this mode shall be in accordance with [5] at 300 bits/s if no other modulation has been agreed on during the connection procedures, for more information see [3].

2.2 ITU-T recommendation V.21

2.2.1 Overview

This recommendation contains a description about how it is intended that a V.21 modem shall work. It also contains some demands that need to be fulfilled. The important demands are:

- The communication circuit is duplex, and there by it is possible to simultaneously transmit data in both directions, at the same time, with a data signalling rate of 300 bits/s or less.

  The modulation used is FSK modulation, that results in a modulation rate that is equal to the data signalling rate.

- The nominal mean frequency for channel 1 and channel 2 are 1080 and 1750 Hz, respectively. The frequency deviation is ±100 Hz, and the higher characteristic
frequency (FA) for both channels corresponds to a binary 0 (the lower (FZ) to a binary 1).

The characteristic frequencies\(^2\), measured after modulation output, may not differ more than ±6 Hz from the nominal values.

It is assumed that the telephone line can make the signal frequency drift by maximum ±6 Hz. This requires that the demodulation equipment shall tolerate drifts of ±12 Hz between the frequency received, and the nominal value.

- a) When both channels are used for simultaneous transmission of data in both directions, channel 1 is used to transmit the caller’s data and channel 2 is used to transmit in the opposite direction.

- b) When only one channel is used for data transmission, the other channel is used for transmission of control signals. In this scenario, channel 1 is used for transmission of data regardless of the direction the data is transmitted.

More information about V.21 can be found in [5].

2.3 ITU-T recommendation V.8

2.3.1 Overview

Many DCE:s can offer communication according to several different V-series modem recommendations. Means are needed to automatically determine the best available operation mode. This can be done by using modem handshake.

This recommendation defines initial modem handshake signals that shall be exchanged between DCE:s, to determine what V-series modem recommendation to use for communication, before the communication channel is established.

The signal that are used, use a common coding format. Each signal consists of a repeated bit sequence. A sequence consists of ten ONEs followed by ten bits for synchronization witch is subsequently followed by information bearing octet. An octet consists of eight information bearing bits, preceded by a start-bit (0) and followed by a stop-bit (1).

2.3.2 Definitions

- CI (call indicator signal): CI signal is transmitted from the calling DCE to indicate the general communication function. It is a V.8 alternative to the call tone CT (defined in [4]). The signal is sent with a regular ON/OFF cadence. The ON period shall not be less than three periods of the CI sequence, and not more than 2 s in duration. The OFF period shall not be less than 0.4 s.

\(^{2}\text{Channel 1: } \text{FA} = 1180 \text{ Hz and FZ} = 980 \text{ Hz; Channel 2: } \text{FA} = 1850 \text{ Hz and FZ} = 1650 \text{ Hz}\)
and not more than 2.0 s in duration. The data signalling rate is 300 bits/s, and the modulation used is V.21(L), the low-band channel defined in [5].

- **CM (call menu signal):** A signal that is sent from the calling DCE, to indicate what modulations modes are supported. The CM signal consist of repeated bit sequences, with a data signalling rate of 300 bits/s. The modulation used is the same as for CI.

- **JM (joint menu signal):** This signal is sent from the answering DCE, and indicates what modulations modes are jointly supported by both the calling-, and answering DCE. The JM signal consists of repeated bit sequences, with a data signalling rate of 300 bits/s. The modulation used is V.21(H), the high-band channel defined in [5].

- **CJ (CM terminator):** CJ is transmitted to confirm detection of JM, and also to indicate the end of CM signal. The signal consists of three consecutive octets, which only include zeros between the start- and stop-bits. The modulation used is the same as for CI and CM.

- **ANS:** Answering tone as defined in [4].

- **ANSam (modified answering tone):** This signal is consist of a sine wave at $2100 \pm 1$ Hz, with phase reversals at an interval of $450 \pm 25$ ms, amplitude-modulated by a sine wave at $15 \pm 0.1$ Hz. The modulated signals envelope shall range in amplitude between $0.8 \pm 0.01$ and $1.2 \pm 0.01$ times the signals average amplitude.

- **sigC:** A signal transmitted by a calling DCE, witch is specific to a V-series modem recommendation.

- **sigA:** A signal transmitted by an answering DCE, witch is specific to a V-series modem recommendation.

### 2.3.3 Data session start-up procedure

Figure 2.1 shows how the signals CI, ANSam, CM, and JM interact to establish a communication channel according to V-series modem recommendation.

#### 2.3.3.1 Start-up procedure in the call DCE

- After a silent period of 1 s, when connected to the line, the DCE shall initiate transmission of CI signal. At the same time, the DCE shall listen for ANS, ANSam, and sigA that are characteristic of an acceptable mode of modulation.
If a suitable sigA is detected, the calling modem shall continue according to the modulation that is indicated by sigA.

- After detection of ANS or ANSam, the transmission of CI is stopped.
- If ANS is detected, the DCE shall proceed according to [7] or [8].
- If ANSam is detected, the DCE shall enter a silent period of $T_e$ s in duration, before transmission of CM. The minimum value for $T_e$ is 0.5 s, but it should be preferably more than 1 s.
- When the silent period $T_e$ has ended, the DCE start transmission of CM and at the same time begins to listen for JM.
- After detection of at least two identical JM sequences, the CJ signal is sent. Before transmission of sigC, after the transmission of CJ, the DCE enters a silent period of $75 \pm 5$ ms, and after the silent period, the sigC is sent according to the agreed V-series modem recommendation.
- If JM shows zero for all modulation modes, the calling DCE may disconnect after transmission of CJ.

2.3.3.2 Start-up procedure in the answer DCE

- After connecting to the line, the answering DCE shall not transmit anything for a period of at least 0.2 s.
- If the CI signal is detected, the DCE shall begin transmission of ANSam.
• If a suitable sigC is detected during transmission of ANSam, the DCE shall stop the transmission of ANSam, and enter a silent period for $75 \pm 5$ ms followed by appropriate sigA, and continue according to the modulation that is indicated by sigC.

• If a minimum of two identical CM sequences are detected, the DCE shall start the transmission of JM. This signal is transmitted until the CJ signal is detected and all of its three octets are received. Detection of CJ is followed by a silent period of $75 \pm 5$ ms and after the silent period sigA is transmitted according to the V-series modem recommendation, agreed to during the modem handshake.

• If JM shows zero for all modulation modes, the answering DCE may disconnect after reception of CJ.

• ANSam is transmitted for $5 \pm 1$ s, in cases when either CM or suitable sigC is detected.

2.4 Digital filters

Different kind of filters can be used for noise suppression, to enhance specific frequency bands, to limit signal bandwidth, and to remove or suppress specific frequencies.

For digital filters the frequency range is always finite and limited to half of the sampling frequency. The transfer function for these filters is usually specified in the z-domain and can be studied in equation 2.1.

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \ldots + b_q z^{-q}}{1 + a_1 z^{-1} + \ldots + a_p z^{-p}}.$$  \hspace{1cm} (2.1)

There are two different basic digital filter types; FIR (Finite Impulse Response) and IIR (Infinite Impulse Response). The difference between FIR and IIR filters in the transfer function 2.1 is that for FIR filters $p = 0$ and for IIR filters $p \geq 1$. These two filter types can be used to design four basic filters; Low-pass, High-pass, Band-pass, and Band-stop filters. Both FIR and IIR filters have their advantages and disadvantages, more about this subject for FIR filters can be found in 2.4.1 and for IIR filters in 2.4.2.

Standard digital filter realization includes direct, parallel, and cascade realization. They all have the same number of delays, adders, and multipliers. The only difference between them is the behavior when implemented in finite word length.

For more information about digital filters, design, and realization the reader is referred to [9].

2.4.1 Advantages and disadvantages of FIR filters

Advantages:
- Linear phase
- Inherent stability
- Very flexible, almost any amplitude response can be realized.
- Low sensitivity to finite word length effects.

Disadvantages:

- Complex to implement, because of the high filter order needed for difficult filter properties.
- Long delays that are not acceptable in some cases.

### 2.4.2 Advantages and disadvantages of IIR filters

**Advantages:**

- Easy to design standard filters, because these are often transferred from analog filter design that is well developed.
- Low implementation complexity, compared to FIR filters.
- Relatively short delays.

**Disadvantages:**

- IIR filters do not have linear phase.
- Bad flexibility, non standard frequency responses are difficult to accomplish.
- Design techniques, besides analog filter design techniques, are complex to be developed and implemented.
- Theoretically stable filters can easy be non stable when filter coefficients are truncated to finite word length.

### 2.5 FSK modulation

FSK (Frequency Shift Keying) is a modulation that is used to send digital information, by changing the continuous carrier frequency.

When only two different frequencies are used to send binary data, the modulation is called Binary FSK. One of the frequencies corresponds to a binary 1 and the other frequency corresponds to a binary 0.

The FSK modulation is represented by equation 2.2.
2.5. FSK modulation

\[ s[n] = A \sin\left(\frac{2\pi(f_c \pm \Delta f)n}{f_s}\right), \]  
\[ (2.2) \]

where

- \( s[n] = \) the transmitted signal
- \( A = \) amplitude of the signal
- \( f_c = \) carrier (mean) frequency
- \( \Delta f = \) frequency deviation
- \( f_s = \) sampling frequency
- \( n = \) sample number

To simplify implementation of the FSK modulator, equation 2.2 can be rewritten as

\[ s[n] = A \sin(\phi[n]), \]  
\[ (2.3) \]

where \( \phi \) is given by

\[ \phi[n] = \frac{2\pi(f_c \pm \Delta f)n}{f_s}. \]  
\[ (2.4) \]

By defining

\[ f_b = f_c \pm \Delta f, \]  
\[ (2.5) \]

equation 2.4 can be simplified to

\[ \phi[n] = \frac{2\pi f_b n}{f_s}. \]  
\[ (2.6) \]

To get continuous phase, \( \phi \) is always depending on the previous value. This gives that equation 2.6 can be rewritten as

\[ \phi[n] = \phi[n - 1] + \Delta \phi_b \]  
\[ (2.7) \]

where \( \Delta \phi_b \) is defined as

\[ \Delta \phi_b = \frac{2\pi f_b}{f_s}. \]  
\[ (2.8) \]

By calculating \( \phi[n] \) modulo \( 2\pi \), \( \phi[n] \) will always be in the interval \([0, 2\pi]\).

For binary signals, equation 2.5 can only take two different values. For the case when the higher characteristic frequency corresponds to binary 1 and the lower to binary 0, the equation can be written as
for binary 1, and as

\[ f_{b=1} = f_c + \Delta f \]  \hspace{1cm} (2.10)

for binary 0. Inserting 2.9 and 2.10 in equation 2.8 results in

\[ \Delta \phi_0 = \frac{2\pi f_0}{f_s}, \]  \hspace{1cm} (2.11)

and

\[ \Delta \phi_1 = \frac{2\pi f_1}{f_s}. \]  \hspace{1cm} (2.12)

Now with help of equation 2.3, 2.7, 2.11, and 2.12 it is easy to implement a FSK modulator with continuous carrier.

### 2.6 Signal energy

For a continuous signal \( x(t) \), the total energy over the time interval \( t_1 \leq t \leq t_2 \) is given by

\[ E = \int_{t_1}^{t_2} |x(t)|^2. \]  \hspace{1cm} (2.13)

Similarly, the total energy for a discrete signal \( x[n] \), over a sample interval \( n_1 \leq n \leq n_2 \), is given by

\[ E = \sum_{n=n_1}^{n_2} |x[n]|^2. \]  \hspace{1cm} (2.14)
3.1 General design

The design of the signal detection part of the V.18 modem is based on the two block diagrams over the automode originating and automode answering procedures in [2]. The block diagram over the automode originating procedure takes care of the case when the modem is used as the calling part of the communication session, and the block diagram over the automode answering procedure is used to handle the case when the modem is used as the answering part of the communication session. These block diagrams can be studied in Appendix A along with the block diagram over the probing mode. The block diagrams are simplified according to the primary objectives (see section 1.3.1) and secondary objectives (see section 1.3.2). From the primary objectives V.21 and Baudot are kept in the schemes. From the secondary objectives, the V.8/V.18(text telephone mode) communication signals have the highest priority. Since EDT (from secondary objectives) is similar to Baudot and Bell 103 is similar to V.21 these are also kept in the diagrams. The rest of the supported V.18 standards in secondary objectives are removed from the diagrams for now, and will be implemented if there is still time for that when primary objectives and the high priority secondary objectives are met. This discussion can be summarized to the following working order:

1) V.21 and Baudot,
2) V.8/V.18(text telephone mode) communication signals,
3) EDT and Bell 103,

where the standards are listed according to the implementation priority.

The simplifications leads to the simplified block diagrams over the automode originat-
Implementation and automode answering modes, see Appendix A for the original block-diagrams, that can be studied in Figure 3.1 and Figure 3.2. Because of these simplifications the probing sequence from section 2.1.5 need also to be reworked. The new simplified probing sequence is:

send V.21 carrier  
send Baudot buffered message  
send EDT buffered message  
send Bell 103 carrier

The block-diagram over the simplified probing mode can be found in Figure 3.3.

*Text telephone mode

Figure 3.1: Block-diagram over the simplified start-up procedure in the originating V.18 DCE.
3.1. General design

Figure 3.2: Block-diagram over the simplified start-up procedure in the answering V.18 DCE.
From Figure 3.1 and Figure 3.2 it is possible to identify two problems, for the primary- and secondary objectives, which need to be solved in order to be able to identify the incoming signals and establish communication. These problems are; identification of incoming signals frequency and identification of the incoming signals bit rate. For the V.8/V.18 (text telephone mode) communication signals, the incoming signal need also to be decoded in order to identify the signal and get the information in it.

### 3.2 Identification of frequency for incoming signal

To determine the frequency of the incoming signal $s[n]$ (Figure 3.4), a simple technique including band-pass filters, and energy comparison, is used. A block diagram over this method can be studied in Figure 3.4.

First the incoming signal is passed through parallel connected/implemented band-pass filters that are toned for the different v-series modem standard frequencies. After this, the filtered signals energy is determined by using equation 2.14. In the last step of the process the energy of the signal is compared to a pre defined threshold value and the other calculated energy values.
The incoming signal has the same frequency as the center-frequency of the band-pass filter that gives the highest energy and at the same time has energy that is over the threshold value.

![Block-diagram over the frequency detection method.](image)

### 3.2.1 Selection of band-pass filter

#### 3.2.1.1 Demanded properties of the band-pass filters

The band-pass filters used to filter the incoming signal shall fulfil as many as possible of the following demands. The filter

- shall be stable even after truncation of the filter coefficients to finite word length,
- shall have low implementation complexity for fast and effective implementation,
- shall have short delay,
- shall have low pass-band ripple,
- shall have fast transition from pass-band to stop-band,
- shall attenuate the signal at least $20\, dB$ in stop-bands.

From the demands above and the advantages and disadvantages of FIR and IIR filters in section 2.4.1 respective section 2.4.2, it is easy to draw the conclusion that the filter type to be used for implementation of the band-pass filters is IIR. The fact that IIR filters do not have linear phase is not a problem, since the filtered signal is only used to get the signal energy at a specific frequency range of the signal.
3.2.1.2 Different IIR filter design methods

There are many different methods to use for IIR filter design. In Figure 3.5 it is possible to study the differences between three basic filter designs; Butterworth, Chebyshev type 1, and Chebyshev type 2 bandpass-filters. The filters are of fourth order with a center-frequency at 1400 Hz. The stop-band edges for the filters are at 1300 Hz respectively at 1500 Hz, with stop-band attenuation of -20 dB. The sampling frequency used for these filters is 8000 Hz.

![Filter Performance](image)

*Figure 3.5: Magnitude in dB over filter performance for fourth order Butterworth, Chebyshev type 1, and Chebyshev type 2 band-pass filters.*

3.2.1.3 Conclusion

As it is possible to see from Figure 3.5, Chebyshev type 2 has the best filter properties of these three basic filter design methods when the filter order is of fourth order.

The attenuation outside the pass-band is lower for both Butterworth and Chebyshev type 1. But the pass-band ripple is too big for Chebyshev type 1 and Butterworth has much wider pass-band than Chebyshev type 2. This is not good since in the worst
3.3. Identification of bit rate for incoming signal

To determine the bit rate of the incoming signal, a method that requires demodulation of a part of the signal is used. The method is described as a block-diagram in Figure 3.6.

As it is possible to see in Figure 3.6, the signal is first decoded. After that the signal is made to a bit stream that only consists of ones and zeros with help of a threshold operation. The amount of zeros and ones in a row is counted to get the number of samples per bit. This value, samples per bit, is compared against pre-known values of samples per bit for different bit rates. In order for this method to work, it is required that the bit rates that the signal can have are known, and the samples per bit for the bit rates are determined.

3.3.1 Asynchronous demodulation of FSK modulated signal

Asynchronous demodulation is used to demodulate the incoming signal. First the incoming signal $s[n]$ (Figure 3.6) is delayed by $k$ samples. The delay $k$, shall be less than the number of samples per bit. The incoming signal $s[n]$ is multiplied by the delayed signal $s[n-k]$, resulting in the signal $v[n]$ (equation 3.1).

$$v[n] = s[n]s[n-k]$$ (3.1)
The resulting signal \( v[n] \) is filtered by a low-pass filter to eliminate harmonic frequencies that arise during the multiplication.

To demonstrate how this work in practice, assume that the time samples \( n \) and \( n - k \) belongs to the same bit, corresponding to the frequency \( f_0 \). This gives \( s[n] \) (equation 3.2), \( s[n - k] \) (equation 3.3), and the product \( v[n] \) (equation 3.4).

\[
\begin{align*}
  s[n] &= A \sin \left( \frac{2\pi f_0 n}{f_s} \right) \quad (3.2) \\
  s[n - k] &= A \sin \left( \frac{2\pi f_0 (n - k)}{f_s} \right) \quad (3.3) \\
  v[n] &= s[n]s[n - k] = A^2 \sin \left( \frac{2\pi f_0 n}{f_s} \right) \sin \left( \frac{2\pi f_0 (n - k)}{f_s} \right) \\
  &= \cdots = \frac{A^2}{2} \left[ \cos \left( \frac{2\pi f_0 k}{f_s} \right) - \cos \left( \frac{4\pi f_0 n - 2\pi f_0 k}{f_s} \right) \right] \quad (3.4)
\end{align*}
\]

The first term in equation 3.4 only depends on \( k \) and is independent of time as long as \( n \) and \( n - k \) belongs to the same bit. The second term in equation 3.4 is eliminated by the low-pass filter. After the low-pass filter operation, the signal \( w[n] \) (Figure 3.6) corresponds to the constant in equation 3.5 or equation 3.6, presumed that \( n \) and \( n - k \) belongs to the same bit, depending on the received bit.

\[
\begin{align*}
  w[n] &= \cos \left( \frac{2\pi f_0 k}{f_s} \right) \quad (3.5) \\
  w[n] &= \cos \left( \frac{2\pi f_1 k}{f_s} \right) \quad (3.6)
\end{align*}
\]

To maximize the effectiveness of the demodulation, \( k \) shall be selected so that it maximizes the function \( d(k) \) in equation 3.7.

\[
d(k) = \left| \cos \left( \frac{2\pi f_0 k}{f_s} \right) - \cos \left( \frac{2\pi f_1 k}{f_s} \right) \right| \quad (3.7)
\]

In Figure 3.7 it is possible to study the data signal and the signals \( s[n] \), \( v[n] \), and \( w[n] \), that arise during the demodulation. More information regarding this demodulation technique can be found in [10].

### 3.3.2 Bit rate determination

To determine the bit rate, the signal \( w[n] \) (Figure 3.6) is thresholded to get a signal, \( m[n] \) (Figure 3.6), that only consists of two values, ones and zeros. In the next step the number of ones and zeros in a row is calculated, and the resulting samples per bit values is compared with already known samples per bit values that corresponds to specific bit rates. This way it is possible to determine the bit rate of the incoming signal.
3.4 Java implementation

The programming language used is Java, as mentioned in sections 1.2 and 1.3, and was specified by Omnitor AB. The Java code follows the Java programming code conventions, as specified by Sun Microsystems in [11]. JavaDoc and low-level comments have been added to the code, in order to make the code more readable (easier to understand), and to facilitate further development of the code.

Like mentioned in section 1.3, the modem shall be implemented as an independent plug in for Sipcon1. The communication window in Sipcon1 is a java class called BufferIO, which include functions to get incoming packets from JMF and to send outgoing packets to JMF. This class, BufferIO, is a part of the plug in package, se.omnitor.tipcon1.so1fV18, where the java code for the modem will be included.

In order to have good survey over the implementation, the implementation of the modem is divided into four parts as follow:

**Part 1:** Implementation of automode originating fraction of the modem, to handle the cases when the modem is used to make a call.

**Part 2:** Implementation of automode answering fraction of the modem, to handle the
cases when the modem is used to answer a call.

**Part 3:** Implementation of the tone generator fraction of the modem, to encode bit streams coded with specific V.18 substandard to tones for that specific substandard.

**Part 4:** Implementation of the probing mode, which is a part of the automode answering fraction of the modem.

All the parts, except the fourth part, was carried out at the same time, in the sense that when something was implemented in one part, the same thing was implemented to the other two parts also, before something new was implemented.

### 3.4.1 Matlab implementation

All the functions needed are first implemented and tested in Matlab. This because Matlab offer better and easier debug environment than Java. Matlab is also used to design the filters and to calculate the filter coefficients needed for the filters, and to verify that the filters stay stable after truncating the filter coefficients.

### 3.4.2 Java design in general

In Figure 3.8 it is possible to study the Java design over the V.18 modem. When BufferIO is run it initiates the SoftV18modem class, with a parameter that tells if the modem is supposed to act as the originating (making a call) part or as the answering (receiving a call) part. According to this information the SoftV18modem class initiates a thread class, either AutoModOrg or AutoModAns. AutoModOrg is run if the modem is needed in the originating mode and AutoModAns is run if the modem is needed in the answering mode. These two thread classes make use of SoftV18Sender, V18Func, Fc, and SendSignal classes.

The class SoftV18Sender is used to send and make data packets for BufferIO in V.18 substandard specific tones. When the V.18 substandard negotiation is completed in AutoModOrg or AutoModAns the control is forwarded to SoftV18, which is a java class of the decoding/encoding project part of the V.18 modem. More specific information about the different classes can be read in section 3.4.3.

### 3.4.3 Design of java classes

- **SoftV18Modem:** This class is the main class of the V.18 modem for negotiation of the V.18 substandard for text telephone session. SoftV18Modem initiates the thread class, with information from BufferIO, AutoModOrg if the modem is used in originating mode, at the same time the SoftV18Sender class is initiated to start transmission of CI signal. If the modem is used in answering mode the SoftV18Modem initiates the thread class AutoModAns.
3.4. Java implementation

Figure 3.8: The Java design of the V.18 modem with all the classes included.

- **AutoModOrg**: This class is used to process the incoming packets in order to try to determine the V.18 substandard that is used for the communication session. The design of this class follows the block diagram in Figure 3.1, including the transmission of the CI signal, which is already initiated in *SoftV18modem*, in order to try accomplish V.8 negotiation of the V.18 substandard to be used for the text telephone communication session. This thread is run until a communication session is made or the thread is terminated because a timer has expired or if it ends in a state that will end the process.

- **AutoModAns**: This class is designed according to block diagrams in Figure 3.2 and Figure 3.3. The class process the incoming packets and tries to determine the V.18 substandard, like AutoModOrg. The difference between AutoModOrg and AutoModAns is that AutoModAns do not send any negotiation signals. If no substandard is recognized in a specific time, the thread goes into probing mode and tries to stimulate the other part to communication. This thread is also run until a communication session is reached or the thread is terminated in a state that ends the process or a timer expires and leads to termination.

- **SoftV18Sender**: This class is used to encode the bit sequences that are coded with a specific V.18 substandard to tones according to that specific sub standards specification. The class is first initiated to some specific V.18 substandard,
then every time when JMF requests a packet via *BufferIO* the packet is put together and forwarded via *BufferIO* to JMF. This class have a bit queue that is read every time a packet is requested. From this queue the packet is constructed by encoding a bit at a time from the queue until the packet have appropriate size. In the scenarios when there are no more bits to encode in the queue, the rest of the packet is filled with carrier bits or zeros until the right packet size is reached.

- **V18Func:** All the public functions needed by the different java classes in the implementation of the V.18 modem are included in this class.

- **SendSignals:** This class is used to store needed predefined signals as bit sequences, for example buffered messages for Baudot and EDT, that are needed in the probing mode state of *AutoModAns*, and different V.8 negotiation signals. The signals are fetched with help of functions that returns the bit sequence for the demanded signal.

- **Fc:** This class is used to store all the needed filter coefficients. The coefficients are stored as public vectors that are declared to be constants.

### 3.5 Test scheme

The test scheme of the software V.18 modem is divided into an own test scheme for both originating- and answering mode. This gives better survey over the testing of the modem.

#### 3.5.1 Test scheme for originating mode

The tests included in the test scheme for the originating mode of the software V.18 modem are defined in this section. For these tests, the text telephone Textlink 9100 is the answering part and the implemented V.18 modem is the calling part.

##### 3.5.1.1 V.21 detection test

- **Identifier:** ORG-V21
- **Purpose:** To verify that V.21 frequency detection works.
- **Method:** The calling part shall be in V.18 mode, transmit CI signal and listen for all possible incoming signals, and the answering part shall be in V.21 mode.
- **Pass criteria:** The calling part shall be able to identify the incoming V.21 signal, sent from the answering part, and communication session shall be established with V.21 as the used substandard.
3.5. Test scheme

3.5.1.2 Baudot detection test
Identifier: ORG-BD
Purpose: To verify that Baudot frequency detection and bit rate detection works.
Method: The calling part shall be in V.18 mode, and the answering part shall be in Baudot mode, transmitting Baudot coded characters.
Pass criteria: The calling part shall be able to detect the incoming Baudot signal, determine the bit rate of the signal to 45.45 bits/second or to 50 bits/second, and to establish a communication session with Baudot as the used substandard.

3.5.1.3 EDT detection test
Identifier: ORG-EDT
Purpose: To verify that EDT frequency detection and bit rate detection works.
Method: The answering part shall be in EDT mode and transmit EDT coded characters. The calling part shall be in V.18 mode.
Pass criteria: The calling part shall be able to detect the incoming EDT signal, and end up in a communication session with EDT as the used standard.

3.5.1.4 CI signal transmission test
Identifier: ORG-CI
Purpose: To verify that transmission of V.8 negotiation signals work.
Method: Both the calling and the answering part shall be in V.18 mode.
Pass criteria: Since the answering part goes into probing mode, see section 3.6.1, before the transmission of the CI signal is started, the calling part shall end in a communication session with V.21, Baudot, EDT, or Bell 103 as the used standard.

3.5.1.5 Monitor 2100 V.21 detection test
Identifier: ORG-21-V21
Purpose: To verify that V.21 frequency detection works in Monitor 2100 (see section 3.6.3).
Method: The calling part shall be forced to Monitor 2100 state and only listen for V.21 frequencies. The answering part shall be in V.18 probing mode.
Pass criteria: The calling part shall detect the incoming V.21 signal and establish a communication session with V.21 as the used substandard.
3.5.1.6 Monitor 2100 Bell 103 detection test
Identifier: ORG-21-B103
Purpose: To verify that Bell 103 frequency detection works in Monitor 2100 (see section 3.6.3).
Method: The calling part shall be forced to Monitor 2100 state and only listen for Bell 103 frequencies. The answering part shall be in V.18 probing mode.
Pass criteria: The calling part shall detect the incoming Bell 103 signal and end up in a communication session with Bell 103 as the used substandard.

3.5.1.7 Monitor 2100 Baudot detection test
Identifier: ORG-21-BD
Purpose: To verify that Baudot frequency detection and bit rate detection works in Monitor 2100 (see section 3.6.3).
Method: The calling part shall be forced to Monitor 2100 state and only listen for Baudot frequencies. The answering part shall be in V.18 probing mode.
Pass criteria: The calling part shall detect the incoming Baudot signal and its bit rate 45.45 bits/second or 50 bits/second, and establish a communication session with Baudot as the used substandard.

3.5.1.8 Monitor 2100 EDT detection test
Identifier: ORG-21-EDT
Purpose: To verify that EDT frequency detection and bit rate detection works in Monitor 2100 (see section 3.6.3).
Method: The calling part shall be forced to Monitor 2100 state and only listen for EDT frequencies. The answering part shall be in V.18 probing mode.
Pass criteria: The calling part shall detect the incoming EDT signal and end up in a communication session with EDT as the used substandard.

3.5.1.9 Monitor 1 timer test
Identifier: ORG-T-1
Purpose: To verify that the timer in Monitor 1 (Figure 3.1) works.
Method: The calling part shall be in Monitor 1 state with all signal detection disabled. The answering part shall be in Baudot or EDT mode with no signal transmission.
Pass criteria: After the timer has expired the calling part shall be terminated.
3.5. Test scheme

3.5.1.10 Monitor 3 timer test

Identifier: ORG-T-3
Purpose: To verify that the timer in Monitor 3 (Figure 3.1) works.
Method: The calling part shall be forced to Monitor 3 state and wait for timer to expire. The answering part shall be in Baudot or EDT mode with no signal transmission.
Pass criteria: After the timer has expired the calling part shall restart to Monitor 1 (Figure 3.1).

3.5.1.11 Monitor 2100 timer test

Identifier: ORG-T-21E
Purpose: To verify that the timer in Monitor 2100 (see section 3.6.3) works.
Method: The calling part shall be forced to Monitor 2100 state, with all frequency detection disabled. The answering part shall be in Baudot or EDT mode with no signal transmission.
Pass criteria: After the timer has expired the calling part shall be terminated.

3.5.2 Test scheme for answering mode

The tests included in the test scheme for the answering mode, including the probing mode, of the software V.18 modem are defined in this section. For these tests, the text telephone Textlink 9100 is the calling part and the implemented V.18 modem is the answering part.

3.5.2.1 V.21 detection test

Identifier: ANS-V21
Purpose: To verify that V.21 frequency detection works.
Method: The calling part shall be in V.21 mode, and transmit V.21 carrier. The answering part shall be in V.18 mode and listen for incoming signals.
Pass criteria: The answering part shall be able to identify the incoming V.21 signal and end up in a communication session with V.21 as the used substandard.
3.5.2.2 Baudot detection test

Identifier: ANS-BD
Purpose: To verify that Baudot frequency detection and bit rate detection works.
Method: The calling part shall be in Baudot mode, and transmit Baudot coded characters. The answering part shall be in V.18 mode and listen for incoming signals.
Pass criteria: The answering part shall be able to identify the incoming signal as Baudot, to determine the bit rate to 45.45 bits/second or 50 bits/second, and end up in a communication session with Baudot as the used substandard.

3.5.2.3 EDT detection test

Identifier: ANS-EDT
Purpose: To verify that EDT frequency detection and bit rate detection works.
Method: The calling part shall be in EDT mode, and transmit EDT coded characters. The answering part shall be in V.18 mode and listen for incoming signals.
Pass criteria: The answering part shall be able to detect the EDT signal and end up in a communication session with EDT as the used substandard.

3.5.2.4 CI signal detection test

Identifier: ANS-CI
Purpose: To verify that detection of the V.8 negotiation signal CI works.
Method: The calling part shall be in V.18 mode and transmit CI signal. The answering part shall also be in V.18 mode.
Pass criteria: Since no V.8 negotiation is implemented, see section 3.6.2, the answering part shall end up in probing mode, and via this mode end up in a communication session with either V.21, Baudot, EDT, or Bell 103 as the used substandard.
3.5. Test scheme 33

3.5.2.5 2100Hz signals detection test

Identifier: ANS-2100
Purpose: To verify that detection of signals at 2100Hz works.
Method: For this test the calling part shall be forced into probing mode and the answering part shall be forced to only listen for 2100Hz signals, see section 3.6.3.
Pass criteria: According to section 3.6.3 the answering part shall end up in probing mode, and via this mode end up in a communication session with either V.21, Baudot, EDT, or Bell 103 as the used substandard.

3.5.2.6 Probing mode termination test

Identifier: ANS-P-PM01
Purpose: To verify that the application is terminated after 3 rounds of the probing sequence.
Method: The answering part shall be forced to probing mode (or to V.18 mode and wait for a timer in monitor A (Figure 3.2) to expire in order to get into probing mode), and the calling part shall be in EDT or Baudot mode. Transmission of EDT or Baudot, depending of mode in the calling part, buffered messages shall be disabled during this test.
Pass criteria: The answering part shall be terminated after 3 full probing sequences.

3.5.2.7 Probing mode transmission of V.21 test

Identifier: ANS-P-V21
Purpose: To verify that transmission of V.21 carrier works in probing mode.
Method: The calling part shall be in V.18 mode and the answering part shall be forced into probing mode, with no detection of signals at 980Hz (because the calling part transmit CI signal in V.18 mode), and only transmission of V.21 carrier enabled in the probing sequence.
Pass criteria: The answering part shall be able to stimulate the calling part to communication with V.21.
3.5.2.8 Probing mode transmission of Baudot test
Identifier: ANS-P-BD
Purpose: To verify that transmission of the buffered Baudot message works in probing mode.
Method: The calling part shall be in Baudot mode and the answering part shall be forced into probing mode, with only transmission of buffered Baudot message enabled in the probing sequence.
Pass criteria: The answering part shall be able to stimulate the calling part to communication with Baudot.

3.5.2.9 Probing mode transmission of EDT test
Identifier: ANS-P-EDT
Purpose: To verify that transmission of the buffered EDT message works in probing mode.
Method: The calling part shall be in EDT mode and the answering part shall be forced into probing mode, with only transmission of buffered EDT message enabled in the probing sequence.
Pass criteria: The answering part shall be able to stimulate the calling part to communication with EDT.

3.5.2.10 Probing mode transmission of Bell 103 test
Identifier: ANS-P-B103
Purpose: To verify that transmission of Bell 103 carrier works in probing mode.
Method: The calling part shall be in V.18 mode and the answering part shall be forced into probing mode, with no detection of signals at 980Hz (because the calling part transmit CI signal in V.18 mode), and only transmission of Bell 103 carrier enabled in the probing sequence.
Pass criteria: The answering part shall be able to stimulate the calling part to communication with Bell 103.

3.5.2.11 Monitor E timer test
Identifier: ANS-T-E
Purpose: To verify that the timer in Monitor E (Figure 3.2) works.
Method: The answering part shall be forced to Monitor E, and the calling part shall be in EDT or Baudot mode with no signal transmission.
Pass criteria: After the timer has expired the answering part shall restart to Monitor A (Figure 3.2).
3.6. Problems

3.6.1 JMF

There were two major problems that could directly be related to JMF. The first problem encountered in this project was that the signals transmitted from the Java class `SoftV18Sender` did not get recognized by the text telephone, Textlink 9100. By recording the transmitted signal at the text telephone, it was discovered that the signal were basically just noise. With this information in hand, the packet was examined just before it was sent to JMF. This test showed that the transmitted signal was correct, before the JMF, and after the JMF process the signal was destroyed. Eventually this problem was solved, after that a lot of time and effort was spent on debugging the different Java classes involved in the process of initiating and delivering packets to JMF. It was discovered that when the JMF was initiated, one parameter was wrong. One number had to be changed from 8 to 16 in the initiation phase of the JMF in order to get it work properly.

The second problem was encountered during the transmission of the V.8 negotiation signal, CI. At start-up in the originating mode, the modem start transmissions of CI in order to try negotiate the V.18 substandard to be used during the communication session. The answering part shall answer by transmission of ANSam (Figure 2.1). This signal was never received; instead the text telephone sent signals according to the probing sequence in section 2.1.5. By taking the time between start-up of the modem, in originating mode, and the beginning of CI transmission it was discovered that JMF needs, in the average case, 5.5 seconds in start-up time before it can send packets including the CI signal. Because of this delay, the answering text telephone gets into probing mode before it has a chance to detect the CI signal.

No solution was found in order to get rid of this unwanted delay. By reading different forums on the internet, about this subject, it was discovered that other projects also suffered from the same problem, some with much worse start-up delays.

3.6.2 V.8 negotiation signals

Like mentioned in section 3.6.1, the answering text telephone gets into probing mode before it can recognize the incoming CI signal. According to [2] this should not be
a problem, since the text telephone should listen for the CI signal, and other V.8 and V.18 (text telephone mode) negotiation signals, in the probing mode. Still it fails to detect CI and CM signals, which are transmitted. One possibility is that the text telephone identifies the signal, but the signal vanishes before it is accepted as a valid detection, and the text telephone remains in probing mode. Other possibilities are that the signal gets distorted in some critical way during the transmission or that no V.8 negotiation signal detection is implemented in the probing mode of the text telephone. Most likely the problem is that the signal gets distorted, and as a result of this the signal is not recognized by the text telephone.

Because V.8 negotiation is a secondary objective it was dropped in favor for the primary objectives. Instead, if in the answering mode of the modem, CI signal is detected in Monitor R (Figure 3.2), the probing mode is entered and the calling part is stimulated to connection by a V.18 substandard included in the probing sequence.

3.6.3 Ans and ANSam

ANS and ANSam are both signals at the frequency 2100 Hz. The difference between them is that ANS is a pure sinusoid while ANSam is amplitude modulated (see section 2.3.2 or [3]).

When the incoming signals frequency is detected to be 2100 Hz in the originating mode or in the answering mode, it has to be determined if it is ANS or ANSam that is received. Some effort was put in to try distinguishing ANS and ANSam, but since no V.8 or V.18 (text telephone mode) negotiation signals are implemented, this was not the main concern and was dropped.

If 2100 Hz signal is detected in answering mode, the problem is solved by going to probing mode in order to stimulate the calling text telephone to a communication session with a V.18 substandard from the probing sequence. In originating mode, the modem is forwarded to Monitor 2100 (Figure 3.1) and continues to listen for the different V.18 substandards implemented.

3.6.4 Transmission of Baudot and EDT

The transmitted Baudot and EDT characters turns out corrupted at the receiving text telephone. The packets that are transmitted looks like they are suppose to do, before they are sent to JMF. This implies that the characters gets distorted somewhere along the path from JMF to the text telephone. It is also possible that the implementation of the procedure that remakes the characters to tones is incorrect. But no obvious errors were found when the implementation was examined. This is a problem that was not solved, and needs to be solved in order to be able to communicate in Baudot and EDT. The text telephone is though able to detect these sub standards, but not to decode the characters correctly.
The implementation of the V.18 software modem is tested according to the tests for originating- and answering mode in section 3.5.1 respectively section 3.5.2. The test results for the originating mode can be seen in Table 1, and for the answering mode, the test results can be found in Table 2.

Overall, the test results are satisfying, both for originating- and answering mode. For the originating mode (Table 1) all test are passed with no wrong detections. In the case for answering mode (Table 2), all except two of the tests are passed with no wrong detections.

The test ANS-EDT, see section 3.5.2.3, detects the substandard EDT eight times, CI signal one time, and V.21 ones. The CI signal detection is a result from that the bit rate determination procedure (see section 3.2) fails to determine the correct bit rate for the incoming signal. This can occur when the part of the incoming signal, which is used for the bit rate determination, is difficult to interpret. The detection of the V.21 signal comes from that the part of the incoming EDT signal, that is used in Monitor E (see Figure 3.2) to distinguish between the different possible V.18 sub standards, have uncommonly much frequency information at 980 Hz. This leads to that the EDT signal is wrongly detected as a V.21 signal instead.

During the test ANS-P-V21, see section 3.5.2.7, the substandard V.21 was detected nine times and Bell 103 was detected once. This was though not a wrong detection, for some strange reason the text telephone transmitted Bell 103 instead of V.21 as it is supposed to. The only logical explanation to this is that there is a bug in the text telephone, which leaded to transmission of Bell 103, and ended in a so-called wrong detection as a result.

For the tests, for both originating- and answering mode, that contain detection of Baudot, only the bit rate 45.45 bits/s was detected. This because the text telephone seems to be able to only transmit at this bit rate (at start-up), it is not possible to change mode so that it can transmit Baudot in a bit rate of 50 bits/s instead.
### Table 1: Test results for the originating mode of the implemented software V.18 modem.

<table>
<thead>
<tr>
<th>Test identifier</th>
<th>Criteria passed</th>
<th>Wrong detection in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORG-V21</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ORG-BD</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ORG-EDT</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ORG-CI</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ORG-21-V21</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ORG-21-B103</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ORG-21-BD</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ORG-21-EDT</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ORG-T-1</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ORG-T-3</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ORG-T-21E</td>
<td>10 / 10</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 2: Test results for the answering mode (including the probing mode) of the implemented software V.18 modem.

<table>
<thead>
<tr>
<th>Test identifier</th>
<th>Criteria passed</th>
<th>Wrong detection in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANS-V21</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ANS-BD</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ANS-EDT</td>
<td>8 / 10</td>
<td>20</td>
</tr>
<tr>
<td>ANS-CI</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ANS-2100</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ANS-P-PM01</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ANS-P-V21</td>
<td>9 / 10</td>
<td>10</td>
</tr>
<tr>
<td>ANS-P-BD</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ANS-P-EDT</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ANS-P-B103</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ANS-T-E</td>
<td>10 / 10</td>
<td>0</td>
</tr>
<tr>
<td>ANS-T-R</td>
<td>10 / 10</td>
<td>0</td>
</tr>
</tbody>
</table>
The purpose of this project was to investigate if it is possible to make a V.18 modem in Java. By studying the results for the originating mode and answering mode in Table 1 and Table 2, it can be said that it is possible to implement a V.18 software modem in Java. The different V.18 sub standards are detected, and the project has resulted in an implementation of a V.18 modem in Java.

The primary objectives are met in the sense that both V.21 and Baudot communication sessions can be achieved. Transmission of V.21 (and Bell 103) carrier signals works fine, and about 95% of the transmitted characters turns out like they are suppose to. This makes the transmitted messages readable and understandable. The Baudot and EDT transmitted characters turn out corrupted (see section 3.6.4), and messages sent with these sub standards can not be read or understood.

At the beginning of the project, one concern was if Java is suitable for real time communication, and it can be said that Java and JMF need much computer power. In order to be able to have multiple communication sessions running at the same time in the gateway, the implementation shall be well optimized. One way to (possible) reduce the clock cycles needed to run the application, is to code the implementation in the programming language C instead of Java. Implementing an application for the RTP communication, that is used instead of JMF, could solve the second problem mentioned in section 3.6.1 and reduce the amount of needed clock cycles.
• Solve the problem with corrupted Baudot and EDT characters (see section 3.6.4).

• Implement an algorithm that can distinguish between ANS and ANSam, in order to make V.8 / V.18(text telephone mode) negotiation signal communication possible.

• Implement detection and transmission of V.8 / V.18(text telephone mode) negotiation signals.

• Implement the rest of the V.18 modem standards, which are not implemented yet.

• Optimize the code for the implementation, in order to reduce the clock cycles needed to run the application.
APPENDIX A

Block-diagrams over Automod originating and answering modes

The block diagram over the automode originating mode procedures can be studied in Figure A.1. The block diagram over the procedures for automode answering mode can be found in Figure A.2, and the block diagram over the probing mode procedure can be seen in Figure A.3. All these three block diagrams are from [2].
Figure A.1: Block-diagram over the start-up procedure in the originating V.18 DCE.
Figure A.2: Block-diagram over the start-up procedure in the answering V.18 DCE.
Transmit ANSam for 1 s.
Stop transmission for 75 ms.
Start 1650 Hz transmission.
Start timer.

Transmit stored message in Baudot, EDT or DTMF.
Start timer.

Transmit 2100 Hz for 1 s.
Stop transmission for 75 ms.
Start 2225 Hz transmission.
Start timer.

Transmit ANSam for 1 s.
Stop transmission for 75 ms.
Start 1300 Hz transmission.
Start timer.

Monitor A

Tm

Next probe

V.23 mode

Monitor A and 390 Hz

Tc

Next probe

Figure A.3: Block-diagram over the automode probing mode.
REFERENCES


