FFI-rapport 2007/01472

# Latency and jitter for different RS-232 converter technologies

Magne Mandt

Forsvarets forskningsinstitutt/Norwegian Defence Research Establishment (FFI)

4 June 2007

FFI-rapport 2007/01472

103502

ISBN 978-82-464-1218-4

## Keywords

Datakommunikasjon

Dataoverføring

## Approved by

| | |
|---|---|
| Stig Lødøen | Project manager |
| Nils Størkersen | Director of Research |
| John-Mikal Størdal | Director |

## Sammendrag

Denne rapporten presenterer resultatene av forskjellige tester av dataoverføringstiden ved bruk av RS-232 konvertere. Det er testet en RS-232 til USB konverter og to forskjellige RS-232 til Ethernet konvertere. Hensikten med testene har vært å få et inntrykk av eventuelle svakheter ved disse teknologiene når de anvendes på tidskritiske data. Testene er ikke ment som en produktsammenligning.

RS-232 til USB konverteren er omtrent like rask som en fysisk serielinje og innfører ikke noen signifikant økning av variasjonen i overføringstid. Den kan dermed trygt brukes for tidskritiske data.

De testede RS-232 til Ethernet konverterne er av multiport typen, med henholdsvis fire og seksten serieporter. De er blitt testet med høy og lav belastning på disse portene, og med høy og lav belastning på Ethernettet (LANet). Det viser seg at de er forholdsvis upåvirket av trafikken på LANet, men begge er følsomme for belastning på serieportene. Når mange porter er aktive samtidig, øker både gjennomsnittlig forsinkelse og variasjonen i forsinkelse, og forsinkelsen vil kunne bli høyere enn det som kan aksepteres. Begge de testede konverterne er også følsomme for konfigurasjonsparametre.

Erfaringene med RS-232 til Ethernet konverterne tilsier at dersom slike konvertere skal brukes til tidskritiske data, så bør de aktuelle produktene testes i den konfigurasjonen som er tenkt brukt og med representativ last på portene for å sikre at de virkelig holder krav til overføringstid.

## English summary

This report present the results of different tests of data transmission time when using RS-232 converters. One RS-232 to USB and two different RS-232 to Ethernet converters have been tested. The purpose of the tests has been to gain an insight into how well these technologies are suited with regards to transmission of time critical data. The tests are not meant to be a product comparison.

The RS-232 to USB converter is almost as fast as a real serial line, and does not increase the variation in transmission time significantly. It can safely be used for time critical data.

The RS-232 to Ethernet converters are of the multiport type, with respectively four and sixteen serial ports. They have been tested with high and low load on these ports, and with high and low load on the Ethernet (LAN). The results show that they are mostly unaffected by traffic on the LAN, but both are sensitive to the load on the serial ports. Both average transmission time and the variation in transmission time increases when many ports are active at the same time, and the delay may become greater than acceptable. In addition, both converters are sensitive to their configuration parameters.

The Ethernet converters's sensitivity to serial port load and configuration leads to the following recommendation: If serial to Ethernet converters are to be used to transmit time critical data, tests should be performed on the products selected for installation using representative loads on the serial ports and the LAN, and using the same configuration as they will have in the finished installation.

# Contents

# 1 Introduction

This report investigates different techniques for distributing time critical data in the updated Ula class submarine combat system. The data originates on serial lines from different sensors. Navigation data are used as a test case, both because these are known to be time critical, and because their format is already known (legacy). The navigation data is present as binary data on RS-485 serial lines from the inertial navigation system, and must be distributed to several different users. This report investigates the transmission time aspects of different distribution techniques.

The total transmission time can be described by three parameters: Nominal transfer time, average latency and jitter. The nominal transfer time is the time it takes to transmit a given message at a given bit rate, average latency is the average transmission time consisting of the nominal time and the average delays caused by hardware and software, and jitter is the variation around the average delay. The nominal transfer time is the theoretical minimum time needed to transmit a message, and can not be reduced for a given message length and baud rate.

Several technologies for serial data distribution are available and will be investigated in this report. The results will depend on the specific product used, and may therefore not hold true for other products using the same distribution principle. However, the results should give an indication of where problems are likely to occur.

The tests are of an exploratory nature and are not meant to be a product evaluation. Therefore, no product comparisons will be made.

The rest of this report is organized as follows: Chapter 2 gives an overview of test conditions and test procedures, chapter 3 to chapter 5 gives the results for the various technologies, and chapter 6 gives a conclusion.

# 2 Test conditions and procedures

All tests were done on a PC with a 2.78 GHz Pentium Pro processor running Windows XP Service Pack 2. The same PC was both transmitter and receiver, to make sure that there were no problems with time synchronization. Windows XP is not a real time operating system, and will introduce some jitter. The test program was the only program running on the computer during the test. The tests used RS-232 serial lines in stead of RS-485, as the PC did not have RS-485 ports. The difference between RS-232 and RS-485 is electrical (RS-485 uses differential signals); this should not affect the delays. All standard tests ran for 10 minutes.

## 2.1 Local Area Network

The Local Area Network (LAN) was a one Gb/s network. It was kept isolated from other traffic, so that the only deterministic load was that commanded as part of the tests. Due to other activity in the laboratory, the physical configuration of the LAN changed during the test period, but this should not affect the test results.

## 2.2 Test data sets

All tests were run using navigation data as the test data set. The navigation system transmits two message types: A short attitude data message each 0.01 second (100 Hz) and a longer navigation data message each 0.1 second (10 Hz). The attitude data message is the most time critical. The protocol uses one start bit, one stop bit and one parity bit, so a byte of data is transferred using 11 bits.

| | Message length | Transfer length | Bit rate | Nominal transfer time |
|---|---|---|---|---|
| 10 Hz | 75 bytes | 825 bits | 19200 b/s | 42.97 milliseconds |
| 100 Hz | 37 bytes | 407 bits | 115200 b/s | 3.53 milliseconds |

*Table 2.1 Navigation System data protocol*

The tests were done using dummy messages containing an accurate timestamp and otherwise filled with one byte characters. True navigation data contains a checksum byte; this was not implemented for the test data.

## 2.3 Code

The code used for the tests implemented a transmitter and a receiver in one program, with the transmitter running as a separate thread. The code was written in C++, and used the WIN32 API to access the serial ports. The original version of the code, version 1, read an entire message (75 or 35 bytes) in a single call to the serial port. Where no code version information is given in the test results, this code was used. This code failed when the jitter got high, and the serial port call started to return less than a full message due to time-outs in the driver. These time-outs can be accessed through the API, but are usually not part of a serial port configuration routine. A new version of the code, version 2, was implemented that read byte by byte from the port. This version had no problems with jitter, and the extra delay caused by extra calls to the serial port was insignificant. The test result sections state when this version was used.

As the normal system time functions have low resolution (typically a couple of milliseconds), a high resolution clock based on the processor clock ticks was used for the time stamps. Messages were time stamped immediately before transmitting, and the clock was read again immediately after receiving a full message.

## 2.4 Tests

The standard tests were run for 10 minutes each. A number of technologies were tested. The support for the different technologies may vary between operating systems.

### 2.4.1 Serial line

Data was transmitted between COM ports on the PC using a real RS-232 serial line. This test serves as the baseline for the other tests.

### 2.4.2 RS-232 to USB converter

This is a popular technology, where one or more RS-232 serial ports are connected to an USB plug. Different directions were tested.

### 2.4.3 Serial line to ethernet converters

The converters make it possible to mix real serial lines and ethernet connectivity, for example using a Local Area Network (LAN) to transmit the message over long distances. The tests used the DeviceMasters and NPorts described in section 5. Several different tests were run using different mixes of ethernet and serial lines and different loads on the converters and the ethernet connection.

## 2.5 Presentation of results

The results for each test are presented as average latency and jitter. Jitter will be described by three parameters: Standard deviation, 99th percentile (i.e. 1 % of the data has a longer transmission time than this), and maximum delay. All times will be given as milliseconds (ms), unless otherwise noted.

# 3 RS-232 serial line test

In this test data was transmitted from one COM port to another on the test PC using an RS-232 line. Separate tests were run of 10 Hz and 100 Hz messages. The test program was the only program running during the tests. Both versions of the code were tested. The results give the baseline for the latency.

## 3.1 Code version 1

This test used the original code version, where the entire message is read in one call to the COM interface. The results are as follows:

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 45.17 ms | 2.20 ms | 0.03 ms | 45.18 ms | 45.76 ms |
| 100 Hz | 3.95 ms | 0.42 ms | 0.01 ms | 3.96 ms | 4.46 ms |

*Table 3.1 Results for the physical serial line test, code version 1*

## 3.2 Code version 2

The results for version 2 of the code are given below. This version of the code reads one byte at a time.

| Message type | Average latency | Average - nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 45.18 ms | 2.21 ms | 0.03 ms | 45.19 ms | 45.72 ms |
| 100 Hz | 3.97 ms | 0.44 ms | 0.01 ms | 3.98 ms | 4.49 ms |

*Table 3.2 Results for the physical serial line test, code version 2*

The difference between version 1 and version 2 is insignificant.

# 4 USB to Serial converter test

USB to serial converters are handy devices for temporary installations like increasing the number of serial ports on a portable computer. This makes them useful for logging data for test and evaluation purposes. As such data often needs to be time synchronized the latency of these converters is of interest.

## 4.1 RS-232 to USB converter

The converter was a Prolific PL2303X using USB 2.0. The results given below shows that at least this product has a performance close to a real COM port, and can be used for time critical data.

### 4.1.1 USB – RS-232

In this test, the message goes out on the USB connector, then back on an RS-232 line.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 45.68 ms | 2.71 ms | 0.03 ms | 45.70 ms | 46.11 ms |
| 100 Hz | 4.07 ms | 0.54 ms | 0.02 ms | 4.16 ms | 4.50 ms |

*Table 4.1 Results for USB – RS-232*

The average is a bit worse than for a real RS-232, line but not significantly so. The jitter is the same as in the RS-232 tests.

### 4.1.2 RS-232 – USB

This is the reverse of path above, and the most relevant message path.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 44.36 ms | 1.39 ms | 0.30 ms | 44.86 ms | 45.12 ms |
| 100 Hz | 5.06 ms | 1.53 ms | 0.28 ms | 5.54 ms | 5.93 ms |

*Table 4.2 Results for RS-232 – USB*

For some reason, average latency goes down for the 10 Hz samples and up for the 100 Hz samples. Jitter increases for both message types. The jitter is not large enough to make time synchronization impossible.

# 5 Serial line to Ethernet converter tests

The following sections give the results for the tests using the DeviceMasters (DM) and NPorts. Each product is presented in a separate section, with each test in a separate subsection. The heading of these subsections describe the message path, so that for instance "LAN – DM1 –RS-232" says that messages travel from the test PC to the DM1 on the LAN, and then on an RS-232 line back to the PC. The PC is always the start and end point, and is not given in the heading. In addition to the message path, the load on the converter and/or the LAN is given. Low converter load is when only the COM port(s) used in the test are active, while under high converter load other COM ports on the converter are sending and receiving 100 Hz samples to and from another computer, to see how the unit handles simultaneous high load on many serial ports. The tests with high LAN load are made to see if that has any significant effect on latency. The LAN load is given as megabit/s. The LAN traffic that generates the load is between two other computers, so that it only creates a load on the LAN and not on the test PC or the converters.

The same tests have been performed on both products as far as that as been possible given the differences in functionality.

## 5.1 DeviceMaster serial hub, 16 ports

The tests used two identical Comtrol DeviceMaster Serial Hubs with 16 serial ports (DM for short). They were connected to the same Local Area Network (LAN) as the test PC with 100 Mb/s connections. Data can be sent to and from the DMs by serial line or by the LAN. The DMs were configured to poll with 1 ms intervals; the default is 10 ms. The DMs will be called DM1 and DM2 respectively. The XP DeviceMaster driver was version 1.9.6.0, and the NS-link version 3.06. The devices are configured through an administration program installed on a PC and a browser interface on the devices. Further information can be found in [1] and at www.comtrol.com.

The tests with high DM load kept 11 other COM ports busy receiving and/or sending 100 Hz data from/to another computer.

### 5.1.1 LAN – DM1 – RS-232, low DM load

The messages are sent out on the LAN to the DM and back to the PC on an RS-232 line.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 47.38 ms | 4.41 ms | 0.51 ms | 48.19 ms | 48.55 ms |
| 100 Hz | 5.44 ms | 1.91 ms | 0.16 ms | 5.82 ms | 7.07 ms |

*Table 5.1 Results for LAN – DM1 – RS-232, low DM load*

The results are worse than the baseline tests, but not dramatically so.

### 5.1.2 LAN – DM1 – RS-232, high DM load

The messages are sent out on the LAN to the DM and back to the PC on an RS-232 line. Eleven other COM ports on the DM are kept busy.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 48.44 ms | 5.48 ms | 0.93 ms | 50.77 ms | 53.27 ms |
| 100 Hz | 6.40 ms | 2.87 ms | 0.79 ms | 8.46 ms | 12.01 ms |

*Table 5.2 Results for LAN – DM1 – RS-232, high DM load*

Both average and jitter increases compared to the low DM load test, but not dramatically.

### 5.1.3 LAN – DM1 – RS-232, ping flood 480 Mb/s

The messages are sent out on the LAN to the DM and back to the PC on an RS-232 line. A ping flood between two other computers generates 480 Mb/s load on the LAN.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 47.38 ms | 4.41 ms | 0.51 ms | 48.20 ms | 48.59 ms |
| 100 Hz | 6.42 ms | 2.89 ms | 0.16 ms | 6.80 ms | 7.90 ms |

*Table 5.3 Results for LAN – DM1 – RS-232, ping flood 480 Mb/s*

The results show an increase in average latency when compared to the first test, but no increase in jitter. The results for the 99$^{th}$ percentile are lower for this test than for the high DM load test.

### 5.1.4 RS-232 – DM1 – LAN, low DM load

The messages are sent out on an RS-232 line to the DM and back to the PC on the LAN (the reverse of the previous tests).

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 44.89 ms | 1.92 ms | 0.74 ms | 46.87 ms | 47.37 ms |
| 100 Hz | 5.58 ms | 2.05 ms | 0.65 ms | 6.84 ms | 8.79 ms |

*Table 5.4 Results for RS-232 – DM1 – LAN, low DM load*

The results show slightly higher latency and jitter than the first test with the opposite message path.

### 5.1.5 RS-232 – DM1 – LAN, high DM load

The messages are sent out on an RS-232 line to the DM and back to the PC on the LAN. Eleven other COM ports on the DM are kept busy.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 46.50 ms | 3.53 ms | 1.39 ms | 49.80 ms | 51.75 ms |
| 100 Hz | 7.08 ms | 3.55 ms | 1.29 ms | 9.76 ms | 11.75 ms |

*Table 5.5 Results for RS-232 – DM1 – LAN, high DM load*

The high DM load increases both the latency and the jitter.

### 5.1.6 LAN – DM1 – RS-232 – DM1 – LAN, low DM load

In this test data was transmitted to the DM via the LAN, and then transmitted on a RS-232 line between two ports on the DM, and then back to the test PC on the LAN.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 47.32 ms | 4.35 ms | 1.09 ms | 48.83 ms | 49.37 ms |
| 100 Hz | 8.42 ms | 4.90 ms | 0.91 ms | 9.76 ms | 11.72 ms |

*Table 5.6 Results for LAN – DM1 – RS-232 – DM1 – LAN, low DM load*

Again, latency and jitter increases. The 100 Hz message shows a larger increase in average latency than the 10 Hz message.

### 5.1.7 LAN – DM1 – RS-232 – DM1 – LAN, high DM load

This test has the same message path as the previous test; in addition 11 other ports on the DM are kept busy.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 50.05 ms | 7.08 ms | 1.71 ms | 54.68 ms | 56.64 ms |
| 100 Hz | 9.85 ms | 6.32 ms | 1.47 ms | 12.69 ms | 18.55 ms |

*Table 5.7 Results for LAN – DM1 – RS-232 – DM1 – LAN, high DM load*

The results show an increase in both average latency and jitter when compared to the low load test. The $99^{th}$ percentile of the 100 Hz data latency is above 12 ms.

### 5.1.8 Test 8: RS-232-DM1-LAN-DM2-RS-232 (Serial Tunneling), low DM load

In serial tunneling mode, DM1 receives data on a RS-232 line, transmits these to DM2 on the LAN, DM2 then outputs them on a RS-232 line. As this means that the message is retransmitted, an increase of the latency is to be expected. How large the increase is depends on what criteria the DM uses to start transmission on the output COM port on DM2

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 94.10 ms | 51.13 ms | 0.40 ms | 95.12 ms | 95.93 ms |
| 100 Hz | 12.50 ms | 8.97 ms | 0.37 ms | 13.37 ms | 14.94 ms |

*Table 5.8 Results for Serial Tunneling*

The increase in latency is large, and seems to indicate that DM2 does not start transmitting until it has received an entire message.

### 5.1.9 Test 10: RS-232-DM1-LAN-DM2-RS-232, high DM load

This is the same message path as the previous test, but with a high DM load. The test had to be run with version 2 of the code due to time-out problems.

| Message type | Average latency | Average - nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 95.17 ms | 52.20 ms | 4.57 ms | 117.45 ms | 151.38 ms |
| 100 Hz | 14.82 ms | 11.28 ms | 5.50 ms | 42.55 ms | 70.14 ms |

*Table 5.9 Results for Serial Tunneling with high DM load*

Both the average latency and the jitter increases. The increase in jitter is dramatic and indicates that the unit may not be suited for this kind of operation. However, 90% of the 100 Hz data has latency less than 17 ms.

### 5.1.10 Test 9: RS-232-DM1-LAN-DM2-RS-232, ping flood 480 Mb/s

This is the same message path as the previous test, but with a high LAN load.

| Message type | Average latency | Average - nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 93.33 ms | 50.36 ms | 0.36 ms | 94.99 ms | 94.17 ms |
| 100 Hz | 12.50 ms | 8.96 ms | 0.38 ms | 13.39 ms | 14.66 ms |

*Table 5.10 Results for Serial Tunneling with high LAN load*

The result is the same as the result for low LAN load.

### 5.1.11 Summary of DeviceMaster results.

The DeviceMaster shows only a small sensitivity to the LAN load used in these tests. It shows a high sensitivity to the COM port load. The default poll interval of 10 ms must be changed to 1 ms for time critical data.

## 5.2 Moxa NPort 5410

This is a small RS-232 to Ethernet converter with four serial ports. It is recommended, and used as a standard, by FLO/IKT. Its functionality is similar to the DM, but with broadcast (UDP) functionality added. It seems to have a minimum delay of 10 ms from the last byte is received on a serial port till the data is transmitted on the LAN, this is known as a "Force transmit" timeout. This delay can not be set lower. The devices are configured through an administration program installed on a PC and a browser interface on the devices. The NPort Administration Suite version was 1.5, the firmware version 2.3. Further information can be found in [2] and at www.moxa.com In the high NPort load tests all the COM ports are active. All tests on this device were run with version 2 of the test code.

The tests revealed a peculiar effect when the NPort used DHCP to get its IP-address. This effect is discussed in section 5.2.15. The effect has been removed from all other results where it occurred.

### 5.2.1 LAN – NPort – RS-232, low NPort load

The messages are sent out on the LAN to the NPort and back to the PC on an RS-232 line.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 46.83 ms | 3.87 ms | 0.20 ms | 47.67 ms | 48.00 ms |
| 100 Hz | 5.29 ms | 1.76 ms | 0.25 ms | 6.37 ms | 6.69 ms |

*Table 5.11 Results for LAN – NPort – RS-232, low NPort load*

All parameters show an increase when compared to a real RS-232 line.

### 5.2.2 LAN – NPort – RS-232, high NPort load

The message path is the same as above, but all COM ports are kept busy.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 47.74 ms | 4.77 ms | 0.85 ms | 50.20 ms | 50.92 ms |
| 100 Hz | 6.10 ms | 2.57 ms | 0.90 ms | 8.66 ms | 10.55 ms |

*Table 5.12 Results for LAN – NPort – RS-232, high NPort load*

Both latency and jitter is higher than when the load is low.

### 5.2.3 LAN – NPort – RS-232, ping flood 670 Mb/s

The message path is the same as above, while a ping flood generates 670 Mb/s traffic on the LAN. The LAN load is higher than for the DM tests because a different server was used to generate the flood.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 46.85 ms | 3.88 ms | 0.21 ms | 47.71 ms | 48.10 ms |
| 100 Hz | 5.27 ms | 1.74 ms | 0.20 ms | 6.11 ms | 6.71 ms |

*Table 5.13 Results for LAN – NPort – RS-232, ping flood 670 Mb/s*

The results are about the same as for the first, unloaded, test.

### 5.2.4 RS-232 – NPort – LAN, low NPort load

The message path is the reverse of the previous tests.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 51.97 ms | 9.00 ms | 3.11 ms | 57.69 ms | 70.37 ms |
| 100 Hz | 15.20 ms | 11.67 ms | 0.59 ms | 15.76 ms | 16.27 ms |

*Table 5.14 Results for RS-232 – NPort – LAN, low NPort load*

The maximum latency value for 10 Hz is a wild point connected to the last sample, the next highest value is 57.69 ms. There is a dramatic increase in both latency and jitter, especially for the 100 Hz message. This may be an effect of the ”Force transmit” limit of 10 ms. The histogram shown in Figure 5.1 indicates that most messages have 10 ms added.

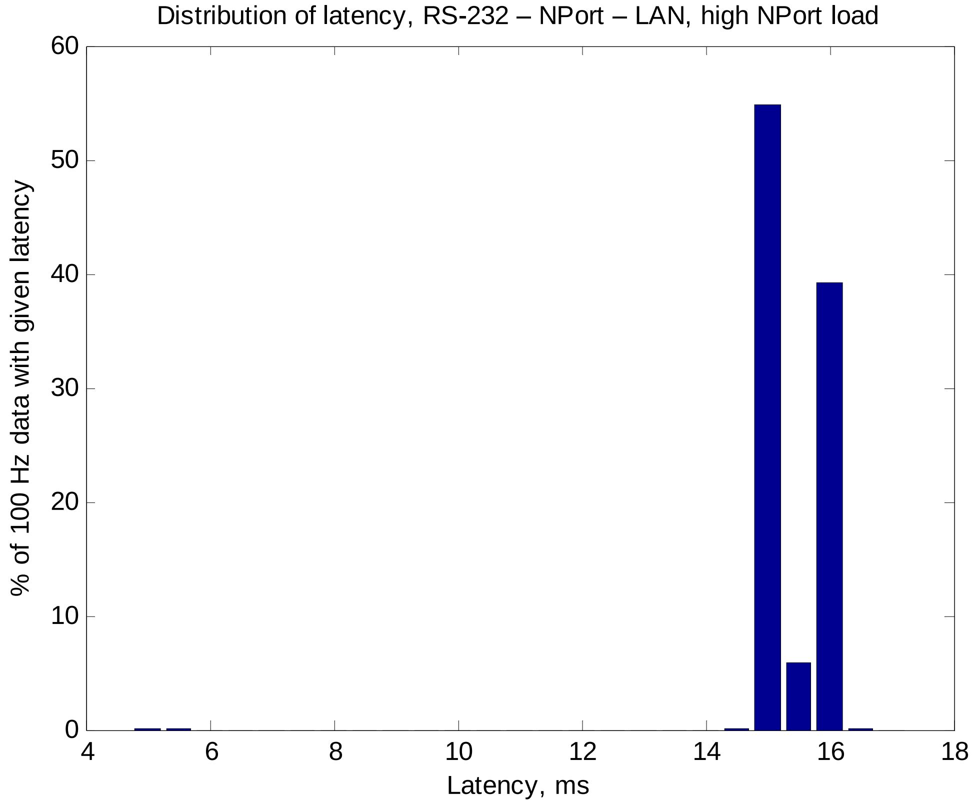


*Figure 5.1 Distribution of latency in the RS-232 – NPort – LAN, low NPort load test*

### 5.2.5 RS-232 – NPort – LAN, high NPort load

This is the same message path as the previous test, but with all COM ports kept busy.

| Message type | Average latency | Average – nominal | Std. Deviation | $99^{th}$ percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 53.23 ms | 10.26 ms | 3.25 ms | 59.65 ms | 66.46 ms |
| 100 Hz | 14.06 ms | 10.53 ms | 3.40 ms | 16.73 ms | 17.23 ms |

*Table 5.15 Results for RS-232 – NPort – LAN, high NPort load*

The lower average for the 100 Hz samples in this test compared to the previous is caused by more samples with approximately six ms delay. The distribution is shown in Figure 5.2.

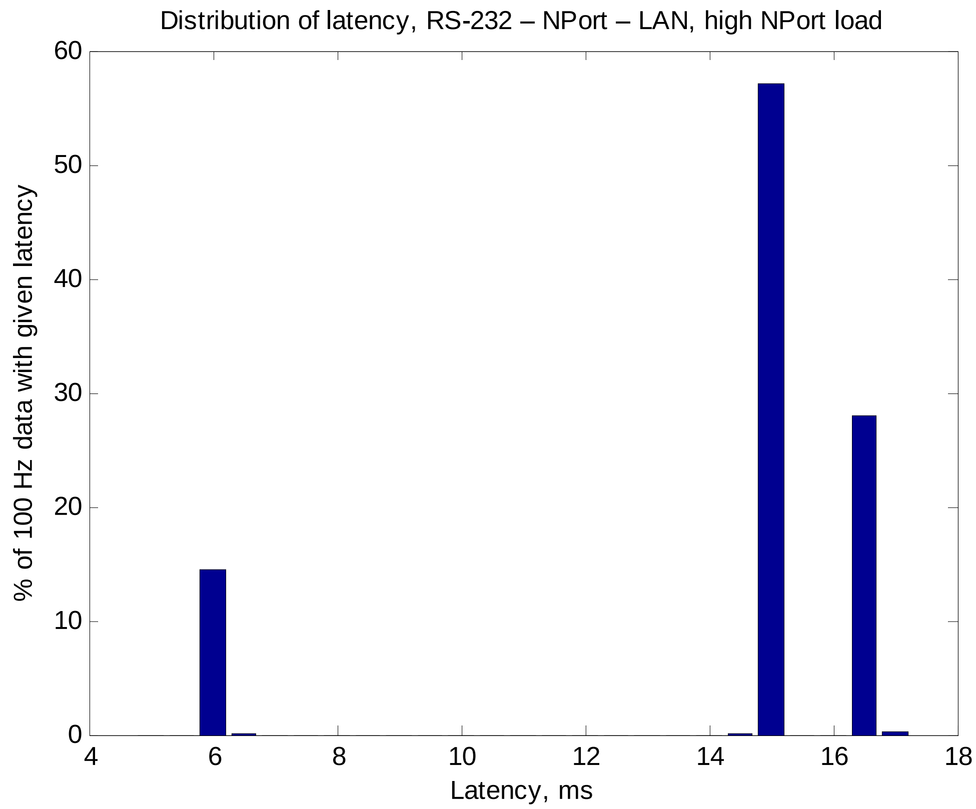


*Figure 5.2 Distribution of latency in the RS-232 – NPort – LAN, high NPort load test*

#### 5.2.6 RS-232 – NPort – LAN, ping flood 670 Mb/s

This is the same message path as the previous test, but with a ping flood running at the same time.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 51.98 ms | 9.02 ms | 3.12 ms | 57.69 ms | 68.89 ms |
| 100 Hz | 15.16 ms | 11.63 ms | 0.65 ms | 15.76 ms | 16.24 ms |

*Table 5.16 Results for RS-232 – NPort – LAN, 670 Mb/s*

This matches results without LAN load.

#### 5.2.7 LAN – NPort – RS-232 – NPort –LAN, low NPort load

In this test, two ports on the NPort were connected with an RS-232 line.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 54.07 ms | 11.10 ms | 3.15 ms | 59.64 ms | 60.62 ms |
| 100 Hz | 12.37 ms | 8.84 ms | 3.17 ms | 18.68 ms | 19.69 ms |

*Table 5.17 Results for LAN – NPort – RS-232 – NPort –LAN, low NPort load*

The jitter is high, especially for the 100 Hz messages.

### 5.2.8 LAN – NPort – RS-232 – NPort –LAN, high NPort load

The message path is the same as in the previous test, and the two other COM ports on the NPort are kept busy.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 55.00 ms | 12.03 ms | 3.24 ms | 61.60 ms | 65.48 ms |
| 100 Hz | 13.46 ms | 9.93 ms | 3.29 ms | 19.65 ms | 22.59 ms |

*Table 5.18 Results for LAN – NPort – RS-232 – NPort –LAN, high NPort load*

The results are a bit worse than for the test with low NPort load, but not dramatically so.

### 5.2.9 LAN – NPort – RS-232 – NPort –LAN, ping flood 670 Mb/s

The message path is the same as in the previous test, with a ping flood added.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 54.05 ms | 11.09 ms | 3.15 ms | 59.54 ms | 72.33 ms |
| 100 Hz | 12.25 ms | 8.71 ms | 3.15 ms | 17.71 ms | 19.67 ms |

*Table 5.19 Results for LAN – NPort – RS-232 – NPort –LAN, 670 Mb/s*

The results are the same as for the low NPort load test.

### 5.2.10 RS-232 – NPort1 – NPort2 – RS-232 (Pair Connection Mode)

This is equivalent to the Serial Tunneling Mode of the DeviceMasters. The messages go on an RS-232 line to NPort1, on the LAN from NPort1 to NPort 2, and back to the test PC on an RS-232 line.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 80.66 ms | 37.69 ms | 0.56 ms | 82.20 ms | 82.92 ms |
| 100 Hz | 13.17 ms | 9.64 ms | 0.43 ms | 14.60 ms | 15.22 ms |

*Table 5.20 Results for Pair Connection Mode*

There is a marked increase in average latency, but not so much in jitter. The 10 Hz message's increase is less that the nominal transfer time, which might indicate that data is output on NPort2's serial port before the entire message has been received on NPort1.

### 5.2.11 RS-232 – NPort1 – NPort2 – RS-232 (Pair Connection Mode), High NPort load

The message path is the same as above; in addition all ports on one of the NPorts were kept busy.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 82.06 ms | 39.09 ms | 1.22 ms | 85.29 ms | 86.51 ms |
| 100 Hz | 14.67 ms | 11.14 ms | 1.23 ms | 17.92 ms | 20.97 ms |

*Table 5.21 Results for Pair Connection Mode with high NPort load*

As has been shown before, a high NPort load leads to an increase in both latency and jitter.

#### 5.2.12 RS-232 – NPort1 – NPort2 – RS-232 (Pair Connection Mode), ping flood 670 Mb/s

The message path is the same as above; in addition a ping flood was run between two other computers.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 80.60 ms | 37.63 ms | 0.49 ms | 82.13 ms | 82.30 ms |
| 100 Hz | 13.12 ms | 9.58 ms | 0.36 ms | 13.89 ms | 15.25 ms |

*Table 5.22 Results for Pair Connection Mode with high LAN load*

There is no significant change from the low NPort load case.

#### 5.2.13 RS-232 – NPort1 – NPort2 – RS-232 (UDP Mode), one port to one port

The UDP mode is a one way, one-to-many, broadcast mode, where one NPort sends the data out on the LAN without getting any kind of acknowledgment from the receivers. The message path is the same as in Pair Connection Mode, but the NPorts are configured differently.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 78.78 ms | 35.81 ms | 0.43 ms | 80.02 ms | 80.25 ms |
| 100 Hz | 11.49 ms | 7.95 ms | 0.47 ms | 13.04 ms | 13.47 ms |

*Table 5.23 Results for UDP Mode, one port to one port*

As can be seen, this mode is a bit faster than the pair connection mode.

#### 5.2.14 RS-232 – NPort1 – NPort2 – RS-232 (UDP Mode), one port to four ports

This is a test of the one-to-many functionality. In this configuration, data comes in one COM port on NPort1, is transmitted to NPort2 on the LAN, and output on all its COM ports.

| Message type | Average latency | Average – nominal | Std. Deviation | 99th percentile | Maximum latency |
|---|---|---|---|---|---|
| 10 Hz | 84.31 ms | 41.34 ms | 0.32 ms | 85.23 ms | 85.84 ms |
| 100 Hz | 16.82 ms | 13.28 ms | 0.42 ms | 18.19 ms | 18.66 ms |

*Table 5.24 Results for UDP Mode, one port to one port*

There is a marked increase in latency when compared to the one-to-one test, and the latency is also larger than for the pair connection mode. There is no significant difference if a ping storm is run at the same time.

#### 5.2.15 DHCP effect

One of the ordinary 10 minutes NPort tests showed a maximum latency of more than 0.5 seconds for the 10 Hz samples. Close inspection of the data revealed that this was not a wild point, see Figure 5.3. The impression one gets from a visual inspection is that the NPort has stopped transmitting for 0.5 seconds, buffered the incoming messages in this period, and then transmitted them as fast as it was able.

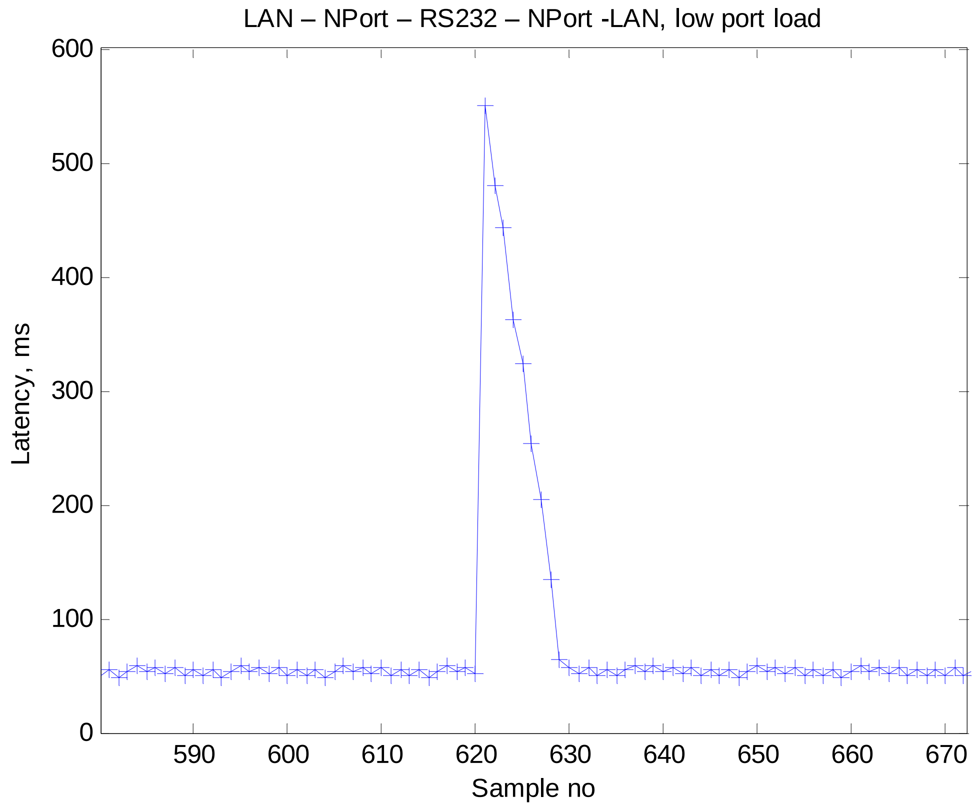


*Figure 5.3  Maximum latency of 0.5 seconds*

This effect was investigated further by running long time tests to see how often it occurred. Figure 5.4 shows that it occurs every two hours, and other tests showed that it occurred for all message paths using the NPorts.

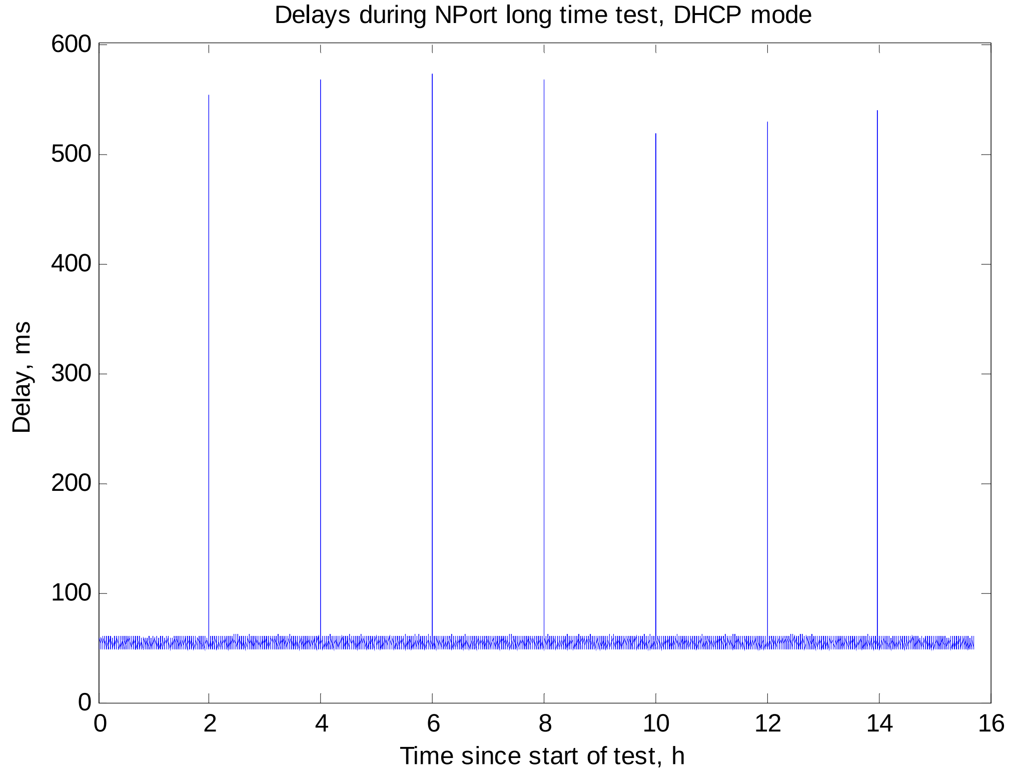


*Figure 5.4 Latency during an overnight test*

Further investigation showed that it occurs when the NPort uses DHCP mode to get an IP-address, but not when it uses a static IP-address. The NPort user guide states that the NPort contacts the DHCP server at fixed intervals to confirm its IP-address in DHCP mode, and that is probably what happens here. The user guide does not state how often this is done, so the explanation can not be absolutely confirmed.

The DeviceMaster has not been subjected to a similar long-time test.

### 5.2.16 Summary of NPort results

The NPort is insensitive to the LAN load used in these tests, but it is sensitive to the COM port load. DHCP should not be used if the NPort is used for time critical data. When data comes in on a serial line there is often an extra 10 ms delay, probably caused by the ”Force transmit” limit of 10 ms.

# 6 Conclusion

The RS-232 to USB converter shows little increase in latency and jitter when compared to a real serial line, and can safely be used for time critical data.

Both the DeviceMasters and the NPorts are largely unaffected by the LAN loads used in these tests. This might change if the LAN load get close to the maximum capacity of the LAN, but that would probably cause problems for other users of the LAN as well.

Both products are sensitive to the COM port load. A high amount of traffic on many COM ports will increase both latency and jitter, sometimes dramatically.

The NPorts are sensitive to the network configuration method selected. Static IP-addresses should be used if possible.

The sensitivity to COM load and configuration leads to the following recommendation: If serial to ethernet converters are to be used to transmit time critical data, tests should be performed on the products selected for installation using representative loads on the COM-ports and the LAN, and using the same configuration as they will have in the finished installation.

## References


[1] "DeviceMaster Serial Hub User Guide," Comtrol Corporation, 2004.

[2] "NPort 5400 Series User Manual," Moxa Technologies Co. Ltd., 2005.


Acronyms and abbrevations

| | |
|---|---|
| COM | Serial port interface |
| LAN | Local Area Network |
| Gb | Gigabit |
| Mb | Megabit |
| b | Bit |
| s | Second |
| API | Application Programming Interface |
| WIN32 | 32 bit Windows API |
| Hz | Hertz |
| DM | DeviceMaster |
| DHCP | Dynamic Host Configuration Protocol |
| UDP | User Datagram Protocol |
| IP | Internet Protocol |
| FLO/IKT | Forsvarets Logistikkorganisasjon/Informasjons- og kommunikasjonsteknologi (Defence Logistsics Organization/Information and Communication Technology) |