



CHARON-VAX and CHARON-AXP/4100/DS/ES/GS

Networking Guide

Applies to CHARON-VAX and CHARON-AXP/4100/DS/ES/GS starting from Build 108-00 (October 2009).

© 2010 Stromasys SA

Document number: 60-16-005

Version: 013

Last update: 26 August 2010

TOC

Charon's networking guide.....	1
Document history	3
Introduction.....	3
Architecture	3
Basic architecture (physical NIC).....	4
The emulated adapter	4
The emulated port	5
The Charon's NDIS driver.....	5
The host NIC driver	5
The NIC/wire	5
Basic architecture (virtual NIC)	6
The emulated adapter	6
The emulated port	6
The Charon's NDIS driver.....	7
The Virtual NIC	7
The Virtual LAN (protocol).....	7
The host NIC driver	7
The NIC/wire	7
Pass through	8
The emulated adapter	8
The Charon's NIC pass through driver.....	8
NIC/Wire	8
Configuration.....	9
Planning you network layout.....	9
Tools you need.....	9
Driver installation	9
Basic architectures (physical/virtual NIC).....	9
Pass Through architecture.....	10
Driver configuration.....	10
Basic architecture (physical NIC)	10
Basic architecture (virtual NIC)	10
Pass through architecture	11
Post driver configuration check	11
Basic architectures (physical/virtual NIC).....	11
Pass through architecture	12
Charon Configuration file settings.....	12
Basic architectures (physical/virtual NIC).....	12
Pass through architecture	13
The Common Configuration parameters	13
Basic architectures (physical/virtual NIC).....	13
The emulated adapter	13
station_address	13
adapter_mode	13
ethernet_port_interface	13
rx_fifo_size	13
The port (ndis5_chpack_port).....	14
Interface	14
port_retry_on_tx.....	14
port_show_driver_statistics.....	14
port_enable_mac_addr_change.....	14
port_pending_rx_number.....	14

port_pending_tx_number.....	15
port_delay_tx_on_overload.....	15
suspend_msg_on_mac_change	15
Pass through architecture	15
Troubleshooting (basic architectures)	16
Some known problems/issues/suggestions	16
MAC address change	16
How to disable.....	16
Collecting Charon's networking statistics	17
The OS running under emulator's statistics	17
The NDIS5/NDIS6 driver statistics.....	17
The host's NIC statistics.....	18
Overflow	18
000000F7 - RX FIFO is full for non critical protocol. Dropping the packet for	18
0000011A - TX queue overflow for	18
Transmit rate times slower then receive	19
Packet duplication.....	19
Problem lookup flow chart.....	20
Troubleshooting (pass through)	21
Submitting BUG report	21

Document history

<i>Date</i>	<i>Who</i>	<i>Comment</i>
13/May/2009	Boris Ovsyankin	Initial revision; The document skeleton with basic principals. The document need review correction and extension.
1/Jul/2009	Boris Ovsyankin	Updated on Alpha manual update
26/Aug/2010	Gregory Reut	Formatting, Pass Through not supported for network

Introduction

The purpose of this document is to provide the user with better understanding of the CHARON network infrastructure, its configuration and troubleshooting.

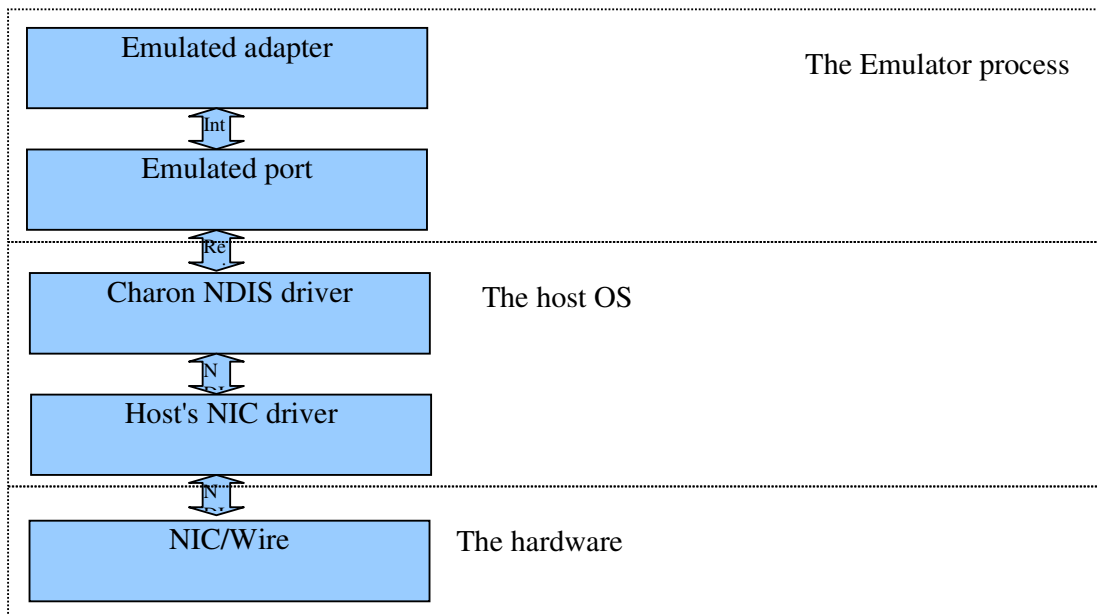
The networking is one of the mostly complicated to troubleshoot CHARON component due its nature and wide range of the possible implementation at the customer's site.

Architecture

The CHARON provides two different network architectures. These are different by the way the packets are delivered to the emulated network adapter. The mostly common and flexible architecture based on the full emulation of the network adapter within emulated environment and CHARON NDIS5/NDIS6 drivers used to interact with the host's NIC. The second network architecture is based on the host adapter specific NIC mapped to the emulated environment with CHARON provided pass through driver. This architecture is limited to the number of the known NIC's and emulated environments.

Basic architecture (physical NIC)

The following schema outlines the CHARON network architecture components involved into the packet transmit/delivery once the CHARON connected to the physical adapter.



The Emulated port, CHARON NDIS driver, Host's NIC driver and NIC/Wire might be referenced as 'virtual' wire, which characteristics depends on the available resources such as CPU, memory, etc...

This configuration implements one to one relationship between number of physical adapters and number of the emulated adapters.

The emulated adapter

Emulated adapter is the emulator's component which emulates specific NIC adapter such as DECNA, SGEC, DEMNA, DEBNA, DE500BA, etc... The adapter emulation seen to the OS running under emulator (for example OpenVMS) as physical network adapter. This connection might extend the schema above with extra box at the top as OS running under emulation NIC's driver and connection arrow with is typically implemented as adapter receive/transmit rings.

The emulated adapter connected to the emulated port with internal link. The emulated OS pushes packets to the emulated NIC, which in turn pass them to the port. At the same time, the port delivers packets to the adapter emulator on the packet receive.

On packet receive, the adapter emulation checks its internal receive FIFO and pushes the packet into the FIFO once it's not empty, or tries to deliver the packet to the emulated environment (the OS running under emulation driver's receive ring).

The adapter emulation receive FIFO was designed as small buffer which should allow to stay over short packet storms without losing the received packets.

Once the receive FIFO is become not empty, the adapter emulation activates the sequencer thread, which tries to free the FIFO by delivering packets from FIFO to the emulated environment.

The emulated port

The port component separated from the emulated adapter to provide various ways to connect the emulated adapter to the outer world while keeping adapter emulation and object design. In the basic architecture, the port is connected to the CHARON NDIS5/NDIS6 packet driver.

On packet transmit, the emulated port pushes the packet to the CHARON NDIS5/NDIS6 driver once provided by the adapter emulation. At the same time, port provides the adapter emulation with the status of the transmitted packet.

On packet receive, the port delivers the packet to the adapter emulation.

The emulated port uses internal connection to the CHARON NDIS5/NDIS5 packet driver with mapped into the kernel receive and transmit queues. The port pushes packets to the transmit queue and read from receive queue on packet availability.

The CHARON NDIS driver

The CHARON NDIS5/NDIS6 drivers implements NDIS protocol driver. This drivers sit on top of the physical or virtual NIC adapters (low binding) and act like dummy interconnect passing all packets to/from NIC driver without filtering or packet content analysis.

The host NIC driver

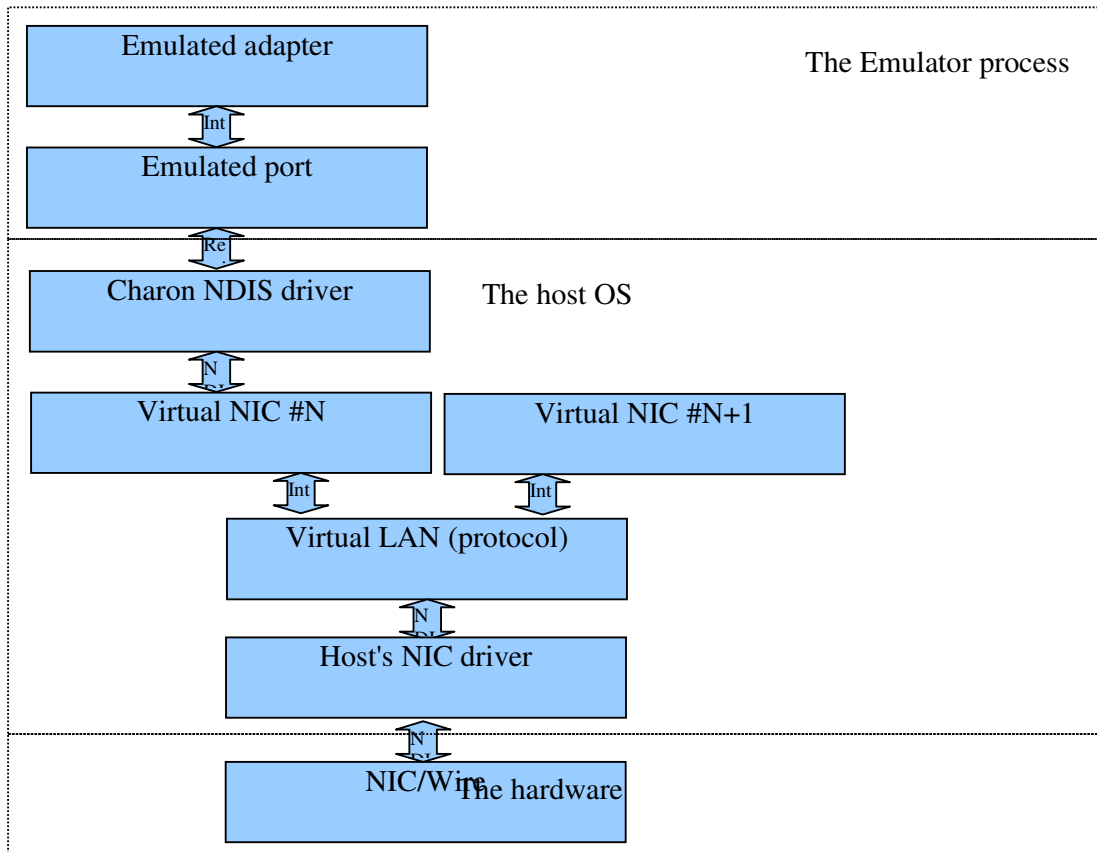
The host NIC driver is card's specific driver provided by the card vendor.

The NIC/wire

The NIC is network adapter connected to the wire.

Basic architecture (virtual NIC)

The following schema outlines the CHARON network architecture components involved into the packet transmit/delivery in case of CHARON MUX (virtual LAN) configuration.



In difference with direct connection to the physical adapter, this configuration has the Virtual LAN between the physical layer and CHARON port. The Virtual LAN might be configured using various technologies with and without connection to the physical LAN. The recommended way is to use the CHARON MUX driver, which implements on-site local virtual LAN connected to the physical LAN, which allows sharing single hardware adapter for both emulator and host operation without conflicts.

This configuration implements one-to-many relationship between number of physical adapters and number of the emulated adapters and host adapters.

Note such configuration requires more resources than configuration with physical NIC and might be less stable in production environment.

The emulated adapter

Same as above.

The emulated port

Same as above.

The CHARON NDIS driver

Same as above.

The Virtual NIC

The Virtual NIC and Virtual LAN (protocol) forms the on-site virtual LAN. This LAN might be connected to the outer world (CHARON MUX) or not (Microsoft Virtual Ethernet Adapter).

The CHARON MUX virtual network virtual NIC implement virtual miniport capable to handle the NDIS protocol and filters at the upper binding. The virtual NIC does not differ from physical one except handling of the carrier status, which is always on for more then one virtual NIC configured.

The virtual NIC is connected to the Virtual LAN (protocol) with internal binding which allow packet flow between virtual NIC as well as to/from the physical LAN.

The virtual NIC permanent MAC address is calculated from the physical adapter's permanent MAC address to be unique on both virtual and physical LAN's.

The virtual LAN can be seen as extension of the LAN with software router implemented by the CHARON MUX.

The Virtual LAN (protocol)

The Virtual LAN protocol is the part of the CHARON MUX virtual LAN which is binded to the physical host NIC. The protocol part passes packets to/from the virtual NIC's into the outer physical LAN. The protocol part performs the filtering on MAC address in order to define the packet destination. No other parts of the packet are analyzed.

Note, that Virtual LAN protocol forces the physical NIC below to go into the promiscuous mode in order to handle the traffic for all configured virtual NIC's which might not work in case of 'clever' routers on the physical LAN.

Note, that being in promiscuous mode, the physical NIC receives all packets which increase the demand to the resources thus filtering is performed in software.

The host NIC driver

Same as above.

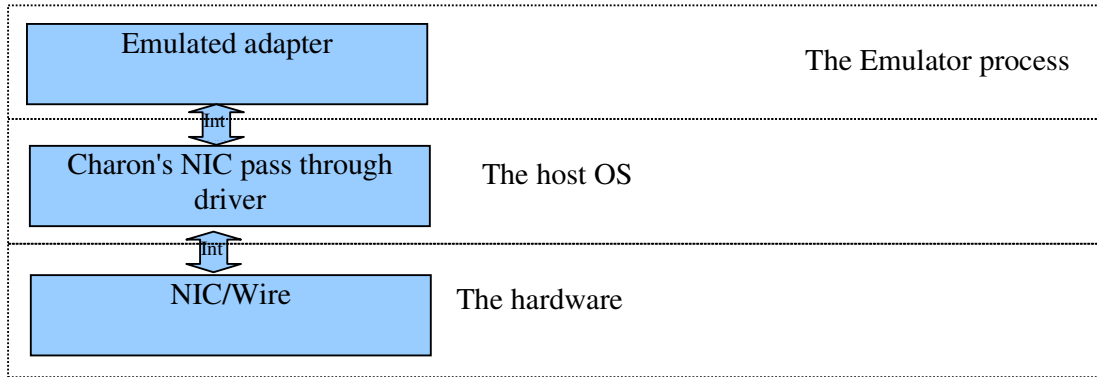
The NIC/wire

Same as above.

Pass through

Pass Through is currently implemented for FDDI DEPFA adapter only. Ethernet network adapters are not supported in Pass Through mode.

The following schema outlines the CHARON network architecture components involved into the packet transmit/delivery once the CHARON connected to the physical adapter via emulated adapter specific pass through driver.



This configuration is specific to the emulated network adapter and 'maps' physical host's NIC into the emulated environment with host's pass through driver and tiny adapter emulation.

This configuration is very limited, but at the same time provides lower possible overhead on packet flow from/to emulated environment.

The emulated adapter

The emulated adapter is tiny layer which main purpose is mapping of the requests done in the emulated environment into the requests to the pass through driver. So, it is not full emulation of the adapter's functionality as above.

The emulated adapter connected to the pass though driver with internal link and implementation specific.

The CHARON NIC pass through driver

The pass through driver 'opens' the hardware NIC to the adapter emulation. The driver acts like host's NIC driver which bypasses all Windows NDIS environment.

NIC/Wire

Same as above.

Configuration

The configuration of the CHARON network depends on the system you plan to implement and outlines step by step below.

Planning you network layout

Before you start with CHARON network configuration, you need to collect the data about your LAN, local system and plan the installation. Here you define:

- how many emulated adapter(s) you need;
- how many and which physical adapter(s) you will use for CHARON and Host;
- which network architecture you will implement;
- what you plan to run in emulated environment (DECnet, IP, Cluster, ...) over each planned emulated adapter;
- what you are running on the LAN, look onto the possible clashes with planned system;

Tools you need

To configure the CHARON network you need:

- CHARON Network Control Center;
- the driver set from the kit you are using;
- notepad;

Driver installation

Depending of the network architecture you select, you need to install the drivers provided with your kit. Please, be aware that the drivers with different major/minor versions might be not compatible, so it is recommended to re-install the drivers to prevent from the failure.

Basic architectures (physical/virtual NIC)

- Stop all emulators running on the host;
- Run Network Control Center;
- Select 'Install/Upgrade' option;
- Follow the Network Control Center wizard;
- Depending on desired network architecture, select only NDIS5/NDIS6 (physical NIC) or both NDIS5/NDIS6 and CHARON MUX (virtual NIC) drivers;
- Start installation and follow the quest;
- Re-boot after the driver installation;

Note, always use Network Control Center to install/upgrade/remove CHARON network drivers, because the tool checks for consistency, hidden obstacles and configuration.

Note, it is safe to remove all CHARON drivers first, then reboot the system, then re-install them from the kit driver set location and finally reboot one more time.

Pass Through architecture

- To be updated

Driver configuration

Now the time to configure the CHARON driver(s), or better say, its interconnect within host's network layout.

Basic architecture (physical NIC)

- Install the NDIS5/NDIS6 driver as described above;
- Stop all emulators running on the host;
- Open adapter property page (via Network Connections applet);
- Reset the 'Locally Administered Address' or 'Physical Address' (physical MAC) to 'Not present' (its permanent value);
- Repeats two last steps for all physical adapters you want to use for CHARON;
- Close ALL Windows networking configuration applets including 'Network Connections';
- Start Network Control Center;
- Select 'Configure NIC' option;
- Choose the adapter you plan to use with for CHARON;
- 'Designate adapter to CHARON';
- Repeat last two steps for all physical adapters you want to use for CHARON;

Note, being designated to CHARON, the physical adapter will loose all host's protocol bindings other then CHARON NDIS5/NDIS6 protocol. This might result in loosing host's TCP/IP connection once you select adapter used by the host.

Note, always use Network Control Center to configure CHARON drivers, because in Windows network architecture, some components (protocols, filters ...) might be hidden and not displayed in the standard configuration applets.

Note, always use Network Control Center to release the adapter to the host once you are done with its usage for CHARON (the action to release adapter is reverse to the above). Note, releasing adapter to the host will not restore its previous bindings (configuration) including TCP/IP configuration. To restore the host's adapter configuration, do that, according with desired protocol's configuration guides.

Basic architecture (virtual NIC)

- Install the NDIS5/NDIS6 driver as described above;
- Stop all emulators running on the host;
- Open adapter property page (via Network Connections applet);
- Reset the 'Locally Administered Address' or 'Physical Address' (physical MAC) to 'Not present' (its permanent value);
- Repeats two last steps for all physical adapters you want to use for CHARON;
- Close ALL Windows networking configuration applets including 'Network Connections';

- Start Network Control Center;
- Select 'Configure NIC' option;
- Choose the adapter you plan to use as outer connection for the local virtual LAN;
- Select 'Multiplex the adapter' and follow the quest;
- Select the multiplexed adapter and go next;
- Add as many Virtual adapters as you planned;
- Come back to the Network Control Center adapter configuration page;
- Select the adapter and you want to use for CHARON connection (this might be virtual or physical except multiplexed adapter)
- Designate adapter to CHARON;
- Repeat last two steps for each adapter you planned to use for CHARON;

Note, being designated to CHARON or multiplexed, the physical adapter will loose all host's protocol bindings other then CHARON NDIS5/NDIS6 or MUX. This might result in loosing host's TCP/IP connection once you select adapter used by the host.

Note, always use Network Control Center to configure CHARON drivers, because in Windows network architecture, some components (protocols, filters, ...) might be hidden and not displayed in the standard configuration applets.

Note, always use Network Control Center to release the adapter to the host once you are done with its usage for CHARON (the action to release adapter is reverse to the above). Note, releasing adapter to the host will not restore its previous bindings (configuration) including TCP/IP configuration. To restore the host's adapter configuration, do that, according with desired protocol's configuration guides.

Pass through architecture

To be updated

Post driver configuration check

Perform this step to ensure, that the CHARON drivers are installed, configured and functioning properly. These steps repeat the driver installation/configuration troubleshooting;

Basic architectures (physical/virtual NIC)

- Perform the driver installation and configuration as described above;
- Stop all emulators running on the host;
- Start Network Control Center;
- Select 'Troubleshoot NIC designated to CHARON';
- Select the NIC configured for CHARON above;
- Make sure the cable is plugged in (if NIC is virtual, multiplexed adapter's cable);
- Click next;
- You will see the NIC bindings – only NDIS5/NDIS6 protocol should be selected.
- Click next;

- You are the page of the DECnet tests; This test page is designed to check that MAC address can be changed to one you planned for that adapter (no duplication), packet flow is normal and there is no MAC address duplication on your LAN; Fill the DECnet address and click next OR skip the test if you do not plan to run DECnet over that link;
- Click next and wait for the test completion;
- You are at the page of the TCP/IP tests; This test page is designed to check that packet flow is normal and there are no IP or MAC address duplication on your LAN. Fill the IP address and netmask OR specify the LAN address instead of node address to run partial tests (such as MAC address duplication). Note, you might skip the tests;
- Click next and wait for the test completion (might take significant time);
- You are at the CHARON configuration file suggestion page; Here, you might select one of the emulated adapters to obtain the suggested configuration file settings for that adapter;
- Click finish;
- Repeat above steps for all adapters you configured for CHARON use;

Note, if you are capable to go through the troubleshoot without problems for all adapters you planned for CHARON use, you might take next step and use them in the configuration file. Otherwise, please read troubleshooting section for deeper problem analysis.

Pass through architecture

Not implemented yet for network adapters

CHARON Configuration file settings

In order to connect the adapter emulation with outer world you need to define this connection in the emulator's configuration file. The way you do that, depends on the architecture you selected and configured in previous steps as well as on emulated adapter type.

Basic architectures (physical/virtual NIC)

You might just copy to the configuration file the lines suggested by the Network Control Center at the driver's configuration check step or obtain them as described at the driver's configuration check step by skipping all tests as follows.

- Perform the driver installation and configuration as described above;
- Stop all emulators running on the host;
- Start Network Control Center;
- Select 'Troubleshoot NIC designated to CHARON';
- Select the NIC configured for CHARON above;
- Click next;
- Click next;
- Select skip the test and click next;
- Select skip the test and click next;
- Select the emulated adapter and copy paste suggested options;

Note, if the emulated adapter you want to configure does not listed, please check the configuration

guide for its configuration. At the same time, you can use the port's instantiation and configuration as is.

Pass through architecture

To be updated.

The Common Configuration parameters

This section describes common configuration parameters, its meaning and purpose (use cases). For specific adapter configuration please read the CHARON configuration guide.

Basic architectures (physical/virtual NIC)

In basic architecture, you need to configure the emulated adapter and port. The configuration parameters available mentioned below. For details, please also read the CHARON configuration guide.

The emulated adapter

station_address

Default – host's physical/virtual adapter permanent MAC address.

The `station_address` defines the permanent MAC address of the emulated NIC adapter and by default initialized with the host's physical/virtual permanent address.

You might need to set this parameter if: (1) your software is licensed with some MAC address value; (2) you are running system booted remotely by the boot server; (3) you want to disable MAC address change.

Please note, that typically emulator will try change MAC address on the host's physical/virtual to the emulated adapter's permanent address at the startup and shutdown, which might introduce extra MAC address changes once the `station_address` is miss used.

adapter_mode

Default – auto.

Some adapter emulation implements the `adapter_mode`, which defines the adapter's speed and duplex settings. Being set to auto, the mode is read from the underlying host's physical/virtual NIC adapter.

Selects one of the supported by emulated adapter (for example: DE500BA) communication modes to be reported to software running under emulator. The accepted values are as follows:

- “Auto”
- “10BaseT-HD”
- “10BaseT-FD”
- “100BaseT-HD”
- “100BaseT-FD”

ethernet_port_interface

Default – blank (means disconnected cable).

The port's instance name must be set to properly configured port to connect the adapter emulation to the outer world.

rx_fifo_size

Default – depends on port, with $\text{ndis5_chpack_port} = ((\text{rx_ring_length} * 3 * 1518) / 1024)$ Kb.

The adapter emulation implements rx_fifo to work around the short term conditions, when the port reads the packets from the port receive ring faster then OS running under emulation is able to handle the emulated adapter receive ring (short packet storm).

Please note, that increasing of this FIFO typically does not make sense, because for short storms the size is enough, while for permanent high load in any case will be overflowed with '000000F7 - RX FIFO is full for non critical protocol. Dropping the packet for ...' message in the CHARON log file.

Please note, that the receive ring treats some protocols as critical (MOP, Cluster, ...) and some as not critical (IP, DECnet, ...) and performs the filtering and drops the non critical packets first. This is done to stay in cluster even loosing the packets.

Increasing the size of the FIFO, and even high load which result in placing the packets into the FIFO, increases the packet round trip, which might become significant. There is no ways to solve that, except decreasing the physical speed of the link (making less packets delivered to emulator) or increasing the host system performance (CPU and memory resources) to let the emulator handle the incoming traffic faster.

The port (ndis5_chpack_port)

Interface

Must be set to the emulated NIC instance name or “(disabled)” to emulate black hole. The black hole, is emulates the condition when the cable is connected, all transmitted packets are dropped and there is no received packets.

port_retry_on_tx

Default 3, the value in range [3, 100]. The number of times, the NDIS5/NDIS6 driver will try to re-transmit failed packet.

Please note, that this option not need to be modified from its default value, because typically, together with software re-transmit the NIC does hardware re-transmit and thus, even single failure might mean serious networking problems (see troubleshooting NDIS5/NDIS6 driver's counters).

port_show_driver_statistics

Not implemented.

port_enable_mac_addr_change

Default – true.

By setting to false, this option disable MAC address change. This might result in condition, when MAC address under emulation is different then on host's virtual/physical NIC and this connection does not receive the direct traffic (only broadcast and enabled multicast packets).

This option was developed to allow work around in conditions where the MAC address change is impossible or not desired.

See more about MAC address change in troubleshooting.

port_pending_rx_number

Default – 64, range [1, 4096] entries (1518 + low ring overhead bytes).

This parameter controls the receive ring size mapped by the port to the kernel's NDIS5/NDIS6 driver. Please note, that increasing number of entries does not required in most cases.

The condition when you need to increase this value, is when troubleshooting of the running connection (see driver counters) indicates dropped by the driver receive packets. This is almost impossible condition, because receiver reads the packets faster then they are delivered to this ring.

port_pending_tx_number

Default – 64, range [1, 4096] entries (1518 + low ring overhead bytes).

This parameter controls the transmit ring size mapped by the port to the kernel's NDIS5/NDIS6 driver. Please note, that increasing number of entries does not required in most cases.

The condition when you need to increase this value, is when you see in CHARON log file the message '0000011A - TX queue overflow for ...'. This is almost impossible condition, because typically driver transmit packets faster the adapter emulation read them from the emulated environment.

port_delay_tx_on_overload

Default – true.

This parameter controls whether port will implement increasing delay for lookup onto the host's emulated environment transmit ring. This delay is initiated once the port's TX ring coming to overflow (red zone condition – 5/6 of the transmit ring is full, delay 2 ms per packet; yellow zone – 2/3 of the transmit ring full, delay 1ms per packet; otherwise delay 0).

You might want to set this parameter to false, if you see transmit rate times slower then receive. By setting this parameter to false, you might hit CHARON log file the message '0000011A - TX queue overflow for ...' which in turn drops the packet with collision status. Finally, you might try to increase the port_pending_tx_number to fix dropped TX packets.

suspend_msg_on_mac_change

Not implemented.

Pass through architecture

To be updated.

Troubleshooting (basic architectures)

This section describes how to troubleshoot the CHARON network. In order to do so, you need CHARON Network Control Center.

Some known problems/issues/suggestions

This is a list of the typical known problems and configuration suggestions.

MAC address change

CHARON networking tries to run in the normal physical/virtual NIC if possible, because normal adapter mode performs the packet filtering in hardware which unloads the host from doing that. But, in emulation world, the MAC address might need to be changed due to DECnet usage or some other software running under emulator which required change to the NIC filter (MAC address is part of the NIC's packet filter).

Unfortunately, in Windows the MAC address change is over-complicated procedure which requires the stopping the NIC driver (and all binded to it protocol stack), changing the host NIC's driver parameters and starting the NIC driver (and all binded to it protocol stack). This requires the administrative privileges and might not go smooth or even go (some NIC drivers implemented in a way which makes the automated MAC address change impossible).

The modern server multi-port NICs requires lots to time to restart, which by itself generates the problems because the traffic is disconnected from the wire during the MAC address change. For example, the DECnet algorithm of the MAC address duplication might not work because of that.

So, that MAC address change is heavy and risky procedure, even if work in most cases.

How to disable

You might want to disable MAC address change because:

- your adapter does not implement that;
- you experience problems with MAC address change;
- your environment is stable and you don't want to have extra point for potential failure;

Please note, that the dynamic MAC address change is required in very rare configurations, where you need to run with one MAC address and then switch to another. For example, you had registered the satellite at boot server with address A and then running satellite need to switch to address B (for example due to DECnet MAC address change). But even in this example, you might register your satellite with address B, define this address as emulated adapter's permanent and specify same address B for your host NIC as physical address (via your NIC property applet).

Note, that here you need extra care and return the host's NIC setting back once do not use the adapter for CHARON anymore to prevent from MAC address duplication on the LAN.

So, the steps to disable mac address change are follows:

- Make sure, you do not have desired MAC address on the LAN; For DECnet MAC address, use Network Control Center's troubleshoot option;
- Define the emulated adapter's station_address with desired MAC address (AA-BB-CC...);
- Set the port's parameter port_enable_mac_addr_change to false;
- Run your host's NIC property applet (via Windows 'Network Connections');

- Go to Configure -> Advanced;
- Change the 'Locally Configured Address' or 'Physical Address' to the desired MAC address (AABBCC...);
- Save new options and exit;

Collecting CHARON networking statistics

This section describes how you can collect the statistics about CHARON networking as well as which statistics are required for the problem resolution and what they mean.

The OS running under emulator's statistics

The OS running under emulation provides the statistics about:

1. The adapter status and error count;
Fatal error count and/or disabled status might indicate adapter initialization problem;
2. The receives/transmits counts;
Indicates the packet flow between the port and adapter emulation;
3. The collision counts;
Indicate the port's TX ring overflow;
4. The MAC address and running protocols;
Indicates the success or failure to change the MAC address;

For OpenVMS all these counters can be read with “analyze/system; >show lan/full”.

The NDIS5/NDIS6 driver statistics

The Network Control Center allows to monitor the NDIS5/NDIS6 driver statistics during emulator run. You also might store this statistics in file, which being captured during problem reproduction might give a clue about the problem source.

1. Run the emulator with reproducer;
2. Run the Network Control Center;
3. Select 'Monitor CHARON NIC activity';
4. Click next;
5. Select the NIC adapter to monitor;
6. Click next;

Note, enable storing real-time statistics during problem reproduction.

The NDIS5/NDIS6 provides statistics in:

1. Errors (receive, transmit, resource, ...);
Indicates problems with installation or luck in resources. For example, having non zero dropped rx frames, you need increase port_pending_rx_number;
2. Receive/Transmit counters;
Indicates the receive/transmit traffic over NDIS5/NDIS6 driver;
3. Mac address, mode,...;
Indicates the active MAC address and driver mode;

The host's NIC statistics

1. Receive/Transmit counters;
Indicates the receive/transmit traffic over NIC;
2. line speed/duplex;
Indicates the link's speed and duplex settings;

Overflow

000000F7 - RX FIFO is full for non critical protocol. Dropping the packet for

...

Imagine a bath. The bath is RX FIFO, the pipe A is packets coming from the wire and the pipe B are packets processed by the OS under emulation.

It is quite obvious, that with constant pipe A and pipe B traffic, where $A > B$, the question of the overflow is only question of the time. With larger bath – that will take longer before overflow happened. But, if high A traffic or low B traffic (actually difference $A-B$ where $A > B$) are temporary (have short duration), then bath will work and receive FIFO will survive overflow. At the same time, if $A > B$ is permanent condition, the RX FIFO will be overflowed in any case.

Please note, that even RX FIFO (bath) is measured in Kb, the speeds in application to the emulator, is measured in packets per second, because the emulator should process frames (Ethernet = [60, 1518] bytes) with almost the same speed.

The lesson from this school exercise is that to prevent overflow, you need to create condition where $A \leq B$. This means one or more from:

- decrease amount of packets coming through host's physical/virtual NIC (user application);
- decrease amount of packets coming through host's physical/virtual NIC by decreasing the link speed/duplex settings;
- add the host's CPU/memory resources to make emulator faster;
- once user application uses large packet sizes (more than Ethernet frame size), think of FDDI where possible, which will decrease amount of packets coming through host's physical/virtual NIC up to 6 times;

0000011A - TX queue overflow for ...

The bath is port's TX ring, the pipe A is OS under emulation pushing the packets into the emulated adapter's transmit ring and the pipe B is NDIS5/NDIS6 driver reading the packets from the port's TX ring.

Here the situation is a bit different than above, just because the condition when emulator is capable to generate the packet load faster the driver is able to pass it to the NIC is unusual. This means, that the throughput (in packets per second) of the 'virtual wire' is worth than emulator performance. To solve that problem, as above, you need to guarantee $A \leq B$:

- Make sure, that host has enough resources to perform I/O (at least 1 host CPU designated for I/O with the hyper threading – off);
- Try to increase NIC's speed/duplex settings (if possible);
- Try to increase the port's parameter `port_pending_tx_number` from its default value; This might fix the problem once the overflow coming from the short storms (The sources of the short storms are follows: (1) the delay in activation NDIS5/NDIS6 driver's thread

responsible for reading packets from port's TX ring and passing them to the NIC; (2) the long packet round trip from NDIS5/NDIS6 driver to NIC and back);

- Try to set `port_delay_tx_on_overload` to true; This will slowdown transmitter (pipe A speed) once the port's TX ring approaching yellow/red zone; Please note, that this slowdown might introduce another problem, where the transmit throughout is times lower the receive, see below);

Transmit rate significantly slower then receive

Note, transmit rate might be couple of times slower in normal conditions.

This problem does not seen as overflow messages in the CHARON log file, but also might come from the overflow.

By default, the `port_delay_tx_on_overload` set to true, which will suspend the transmitter on approaching yellow/red zones on port's TX ring. At the same time, just setting the `port_delay_tx_on_overload` to false, might result in '0000011A - TX queue overflow for ...'.

The proposed road map is follows:

1. set port's `port_delay_tx_on_overload` to false;
2. run your reproducer;
3. if you have '0000011A - TX queue overflow for ...' continue, otherwise jump to #5;
4. read the '0000011A - TX queue overflow for ...' and evaluate its suggestions, by re-starting from #2; See note below;
5. If the problem is not solved, solved (transmit speed is acceptable) or the problem is not solved, the problem does not depend on CHARON networking. In such case, the application and LAN should be analyzed with care;

Note, increasing the `port_pending_tx_number` to values above 256/512 could be dangerous, because outlines permanent nature of the overflow (for example: running on single CPU host or 10Mbps half duplex) and thus requires restoring the `port_delay_tx_on_overload` to true.

Packet duplication

Running ping you got packet duplication. This typically means, that you have one or more node(s) with the same TCP/IP or MAC address on the LAN. This also might happened, if you have single node with multiple interfaces connected to the same LAN, when these interfaces are configured with the same MAC (same DECnet address) or TCP/IP address.

The proposed road map is follows:

1. Run the reproducer and make sure that the problem is reproduced;
2. Stop all emulators on the host;
3. Run Network Control Center;
4. Select 'Troubleshoot NIC designated to CHARON';
5. Select the NIC configured for CHARON above;
6. Make sure the cable is plugged in (if NIC is virtual, multiplexed adapter's cable);
7. Click next;
8. You will see the NIC's bindings – only NDIS5/NDIS6 protocol should be selected;
9. If the bindings are not correct, fix them, exit the tool and re-start from #1;

10. Click next;
11. You are the page of the DECnet tests; This test page is designed to check that MAC address can be changed to one you planned for that adapter (no duplication), packet flow is normal and there is no MAC address duplication on your LAN; Fill the DECnet address and click next OR skip the test if you do not plan to run DECnet over that link;
12. Click next and wait for the test completion;
13. You are at the page of the TCP/IP tests; This test page is designed to check that packet flow is normal and there are no IP or MAC address duplication on your LAN. Fill the IP address and netmask OR specify the LAN address instead of node address to run partial tests (such as MAC address duplication). Note, you might skip the tests;
14. Click next and wait for the test completion (might take significant time);
15. You are at the CHARON configuration file suggestion page; Here, you might select one of the emulated adapters to obtain the suggested configuration file settings for that adapter;
16. Click finish;
17. You had IP/MAC address duplication messages, then fix them in emulated environment and re-start from #1, otherwise continue;
18. The node you just pinged might also have MAC/IP address duplication; So, evaluate your LAN to remove this problem, including single node with multiple interfaces into the LAN;
19. Check that your emulator does not sit with two interfaces, with same MAC/IP addresses (DECnet/IP enabled on both interfaces) into the LAN;

Problem lookup flow chart

This section is an attempt to create more or less complete set of the instructions on what to do step by step if you have problem with CHARON networking.

1. If emulator is failed to start due to configuration problems, read the configuration manual and the 'Configuration' section of this manual;
2. Passing #1 means, that you are able to start emulator without configuration problems;
3. Do the steps described in 'Post driver configuration check' section above; Do that, even if you done it at the configuration stage;
4. If at previous step, you did not find the errors for all adapters involved in your configuration, continue, otherwise fix them and re-run reproducer;
5. Read the 'Some known problems/issues/suggestions' once your problem is listed there;
6. If at previous step, you did not find the errors for all adapters involved in your configuration, continue, otherwise fix them and re-run reproducer;
7. Collect the running instance statistics as described in 'Collecting CHARON networking statistics'; Make sure, you done that during problem reproduction;
8. Analyze the statistics; it shows operation on all modules of the CHARON network architecture. Check, that link parameters are correct, you have packet flow, the MAC address is correct/changed properly, you run all protocols properly configured in your emulated environment, you do not have fatal errors, etc...
9. If at previous step, you did not found/fix the errors, submit the bug report as described in 'Submitting BUG report' section.

Troubleshooting (pass through)

To be updated.

Submitting BUG report

Please, do not hurry to submit the BUG report. Please, evaluate the instruction listed in the 'Troubleshooting' section first. This might solve your problem and, if not, collect necessary information required for BUG evaluation.

If this guide does not help, and you evaluated all suggested options, please provide the following information:

1. Your configuration file;
2. Your log file;
3. The statistics as described in 'Collecting CHARON networking statistics' section;
4. Description of your LAN topology;
5. Description of your problem with instructions on how to reproduce;