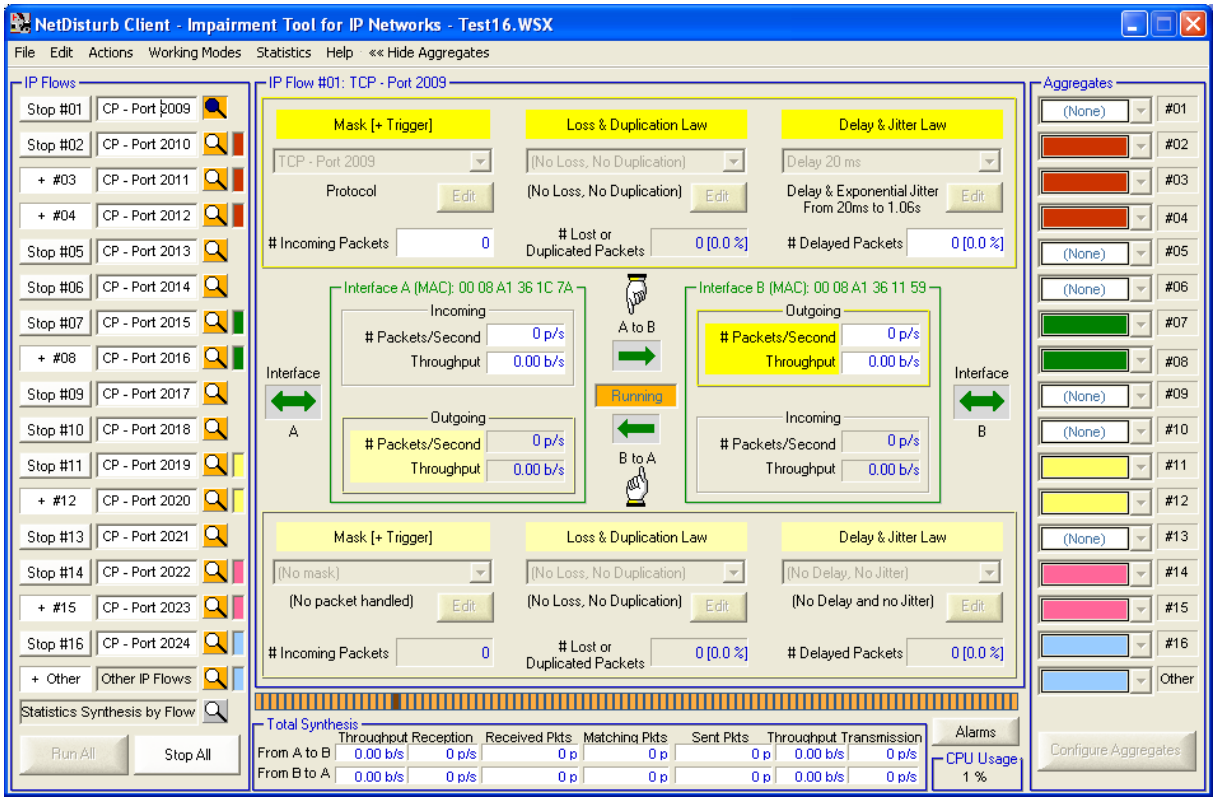
NetDisturb

Version 4.4

Impairment Emulator Software for IP Networks (IPv4 & IPv6)



User Guide

ZTI / 1 boulevard d'Armor / BP 20254 / 22302 Lannion Cedex / France
Phone: +33 2 96 48 43 43 / Fax: +33 2 96 48 14 85
Email: contact@zti-telecom.com / Web: www.zti-telecom.com

The content of this User Guide is provided for informational use only. It is subject to change without notice, and must not be used as a commitment by ZTI.

ZTI could not be liable for any direct or indirect damages caused by the software or User's guide imperfection.

By any chances, if mistakes have slipped in this guide, do not hesitate to contact our client support and make remarks.

Except when allowed by license agreement between ZTI and User, no part of this guide or the software may be reproduced, transmitted in any form or by any means.

To contact us:

ZTI
1 boulevard d'Armor
BP 20254
22302 Lannion Cedex
France

Phone: +33 2 96 48 43 43
Fax: +33 2 96 48 14 85
Web: http://www.zti-telecom.com or http://www.zti.fr
Email: contact@zti-telecom.com (marketing & sales)
support@zti-telecom.com (technical support)

Software License Agreement

This is an agreement between you (legal entity or physical person) and ZTI.

- **COPYRIGHT**

The enclosed Software and documentation (here after called the Products) remains the property of ZTI.
French copyright laws and international treaties protect this product.
ZTI grants you the right to use the products according to the following:

- **USE OF THE SOFTWARE**

You may:

- Install the software on the hard disk of your system in accordance with the software protection described in the next paragraph.
- Make 1 backup copy of the software, provided that this copy is not used or installed on any computer.
- Use the Products correctly.

In accordance with copyright and patent laws, the Licensee undertakes:

- To use the Products only for its own use
- Not to modify the Products
- Not to make illegal copy of the Products
- Not to give, rent, sublicense or sale the Products
- To protect and respect ZTI and its Products reputation.

- **SOFTWARE PROTECTION**

NetDisturb is licensed on a per workstation basis. You will need to purchase a separate license for each machine that you install it on. Each licensed copy of the software installed on a workstation has a unique Site Code, which requires the corresponding unique Site Key to be entered before the tool is being operational.

- **LIMITED WARRANTY**

The Software is supplied without any express or implied warranty regarding the performance or results obtained by the use of the Products.
ZTI warrants that the software media (i.e. CD-ROM) will be free of material defects for a ninety (90) days period following your purchase. The limited warranty applies to the media and not to the information contained on it. If the media does not comply with this limited warranty, the only remedy is the replacement of the media software
In no event, ZTI will be liable for any kind of direct or indirect damages caused by the Products.

- **JURISDICTION**

French laws will govern this agreement.
The court of GUINGUAMP-France shall finally settle all disputes arising out of or in connection with this Agreement.

For further information, please contact: ZTI customer support department.

ZTI
1 boulevard d'Armor
BP 20254
22302 Lannion Cedex
France

Phone: +33 2 96 48 43 43
Fax: +33 2 96 48 14 85
Email: support@zti-telecom.com or support@zti.fr
Web: http://www.zti-telecom.com or http://www.zti.fr

Table of contents

PART 0 Preface

0.1 Organization of this manual

This user guide is aimed at helping you to discover and use **NetDisturb**. This manual is organized as follows:

- **Part 1: Product Overview**

Briefly describes the key features of the **NetDisturb** software.

- **Part 2: What's new in NetDisturb version 4.4**

This part is a general overview of new features and main corrections provided with **NetDisturb** version 4.4 and important information to upgrade from previous versions.

- **Part 3: Install NetDisturb**

Product requirements and how to install the software downloaded from the Internet or from the CD-ROM.

- **Part 4: Software License Configuration**

Describes how to configure the license and how to proceed for the license transfer

- **Part 5: Uninstall NetDisturb**

Describes how to uninstall the software.

- **Part 6: Run NetDisturb**

Describes how to run the **NetDisturb** Server and **NetDisturb** Client.

- **Part 7: Using the NetDisturb Client**

Describes how to use the **NetDisturb** Client.

- **Part 8: Using the NetDisturb Server**

Describes how to use the **NetDisturb** Server.

- **Part 9: Annexes**

Describes additional information about the mathematical laws used by **NetDisturb**, the default context value and the parameters saved in the Registry database.

0.2 Minimum System Requirements

NetDisturb requires the following minimum system requirements to operate properly:

- Platform: PC running Windows NT4 (SP6 recommended), 2000 or XP
- Pentium processor with 256 Kb memory at least
- Two identical Ethernet NICs: Ethernet, Fast Ethernet, or Gigabit Ethernet network interface card.

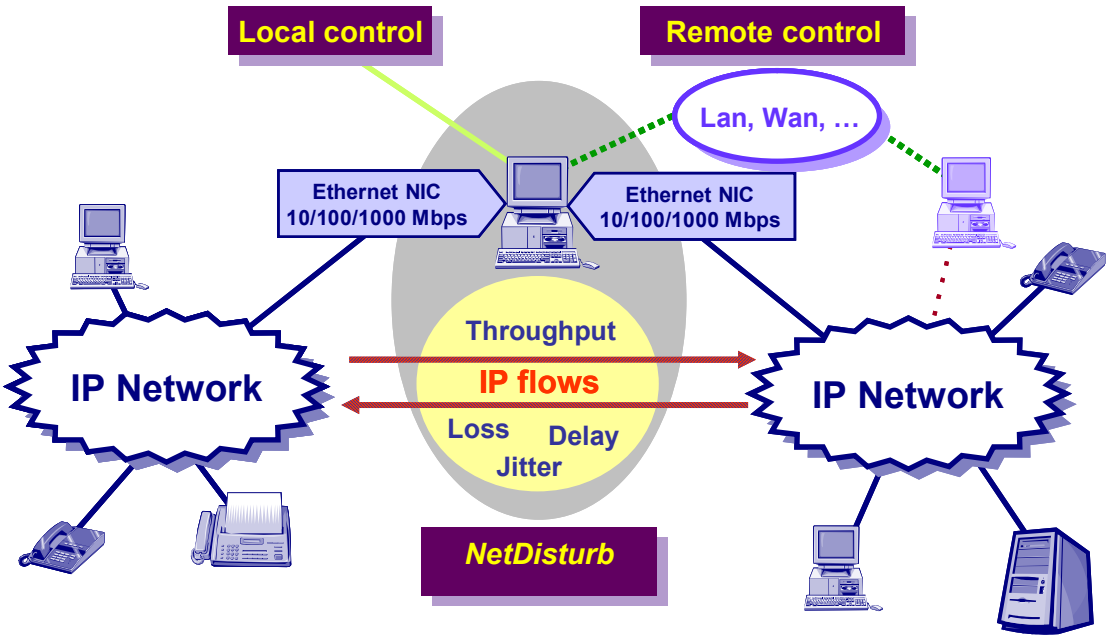
- 1024 x 768 display and DPI setting = Normal size (96 DPI)
- 15 MB free hard disk space

Note: PC multiprocessors and hyper-threading are also supported.

PART 1 NetDisturb Overview

NetDisturb is an IP network emulator software that can generate impairments (latency, delay, jitter, bandwidth limitation, lost and duplicate packets) over IP networks (IPv4 and IPv6). **NetDisturb** allows the user to disturb flows on an IP network and so to study the behavior of applications, devices or services in a disturbed network environment.

NetDisturb is inserted between two Ethernet segments (on the same IP network or two different IP networks) and operates bi-directional packet transfer on Ethernet, Fast Ethernet and Gigabit network interface cards.

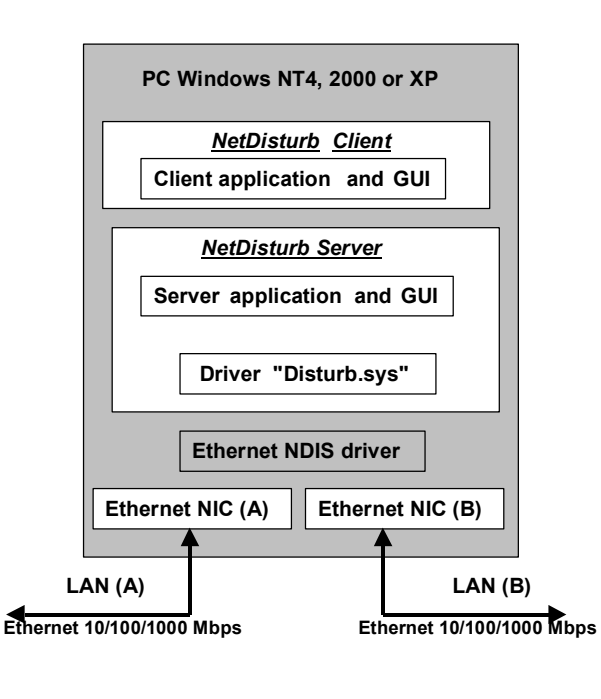


Product Requirements

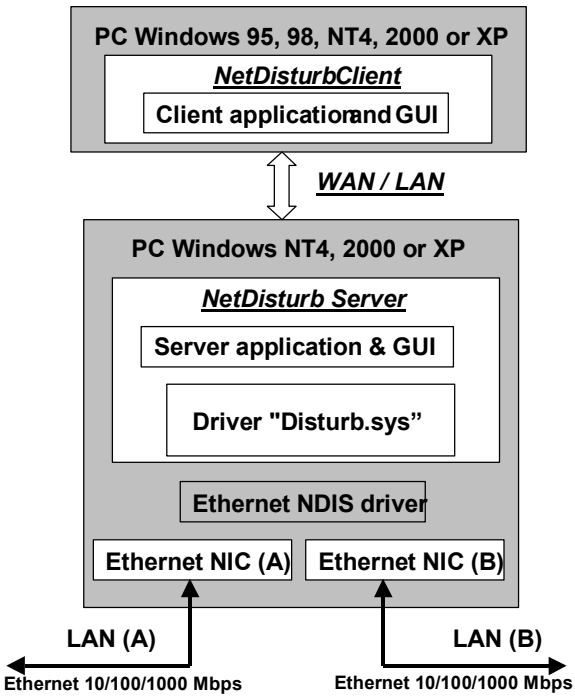
* Platform: pentium PC running Windows NT4 (SP6), 2000 or XP with Microsoft TCP/IP installed and at least 256 MB Ram.
* Hyper-threading and PC multiprocessors are also supported.
* Two Identical Network Interfaces Cards (NIC): Ethernet, Fast Ethernet, or Gigabit Ethernet network interface card.
* 1024 x 768 display and DPI setting = Normal size (96 DPI).

Configurations

Based on a Client-Server architecture, the **NetDisturb** software is made of two parts: Server and Client. The Server handles the impairment characteristics and the Client manages the Server using a simple graphical interface.
This allows two configurations where the Server and the Client parts may be installed on the same PC host (local control), or the Server part is located on one PC and the Client part is located on a second PC (remote control). In this second configuration, the Client dialogs with the Server by using a Wan (for example: PSTN or ISDN) or a LAN link.
Both configurations require two identical Ethernet Cards for the Server.



Client and Server on the same PC



Client and Server on 2 PCs

The "Disturb.sys" driver is located in the kernel of the operating system and is installed above the NIC drivers. This driver is used by **NetDisturb** to handle the exchanges with the NICs.

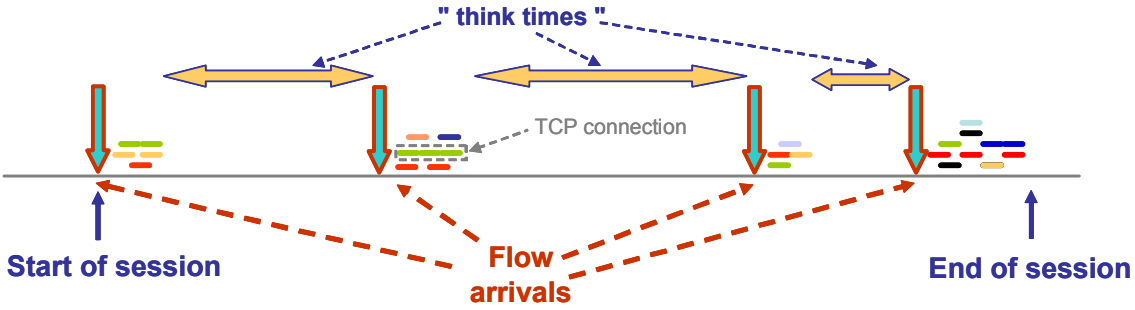
Products features

What are the major features of **NetDisturb** V4.4?

Key features

- Client-Server Architecture
- Impairments: Latency, Loss, Duplication, bandwidth limitation, Delay and Jitter (mathematical laws and user-defined files)
- 16 configurable IP flows per direction with optional trigger condition
- Aggregates of IP flows can be defined (set of IP flows sharing the same Delay & Jitter Law)
- Unidirectional or bidirectional packet impairments
- Connections per IP flow: impairments are applied to the IP flow or to each connection of the IP flow
- Ethernet / Internet modes (desequencing of the packets)
- Easy to use and intuitive Graphical User Interface
- Statistics display and export detailed statistics in a file

NetDisturb is based on the notion of IP flows.
A flow is a set of packets with a set of common packet properties, and can be unidirectional or bidirectional.
Flows are part of sessions (successions of flows and "think times") related to some homogeneous user activity (e-commerce, mail, MP3 file, web, etc.).



An IP flow is described by using a n-tuple.
In the typical case, the following 5-tuple is used: IP addresses, protocol and port numbers.
An IP flow is composed of connections (such as TCP connections to make FTP transfer by example).

To define the n-tuple for an IP flow, **NetDisturb** uses the notion of mask.

A mask is the combination of the following optional parameters:

IP version (IPv4, IPv6 or IPv4 & IPv6)

Ethernet header
- MAC destination address
- MAC source address

List of VLAN-ID (Ethernet frames 802.1Q)

IP Header

- Destination IP address
- Source IP address
- Protocol (ICMP, TCP, UDP, SIP, …)
- Differentiated services (TOS)

List of Ports (for TCP or UDP packets)

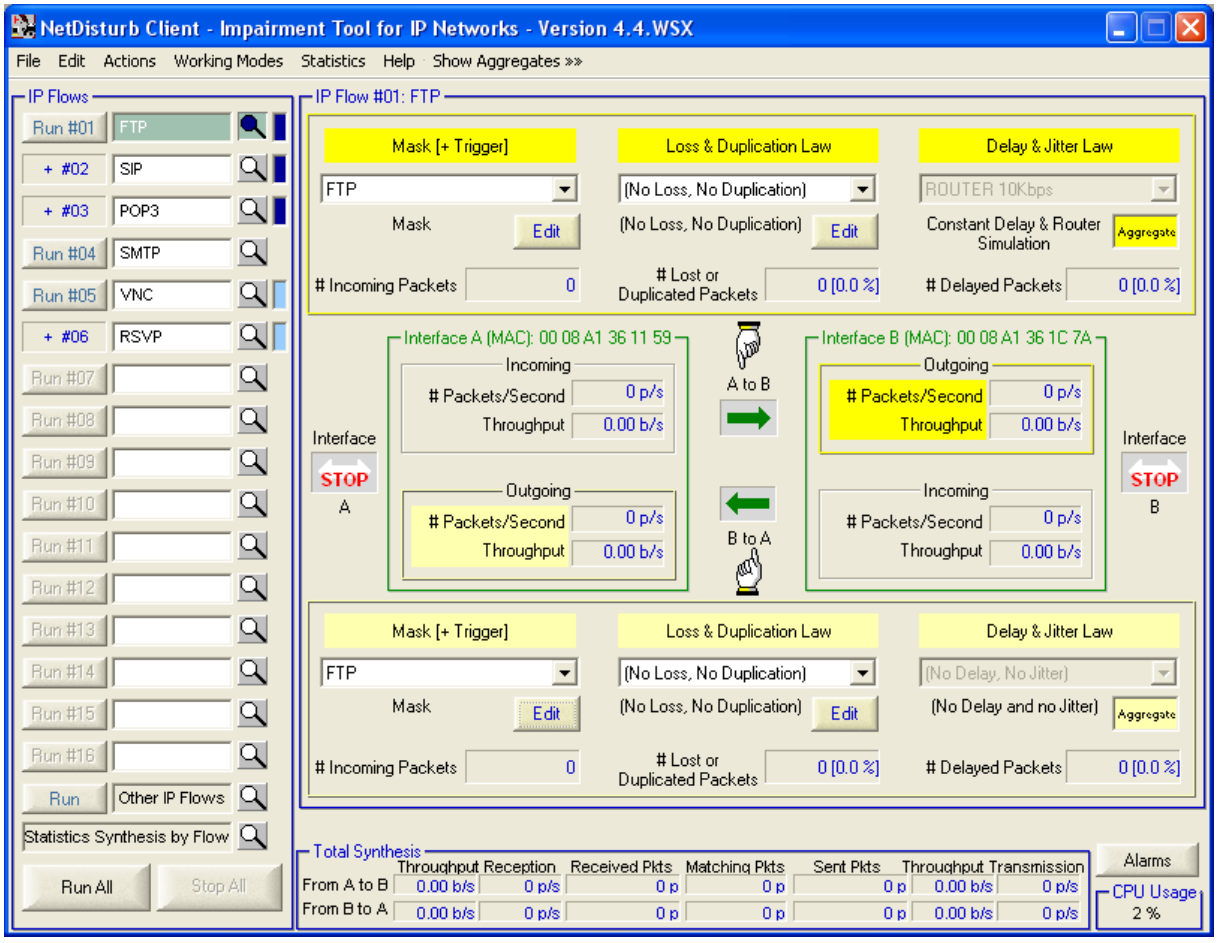
- Destination port list
- Source port list

Note: a trigger can be associated optionally with the mask.

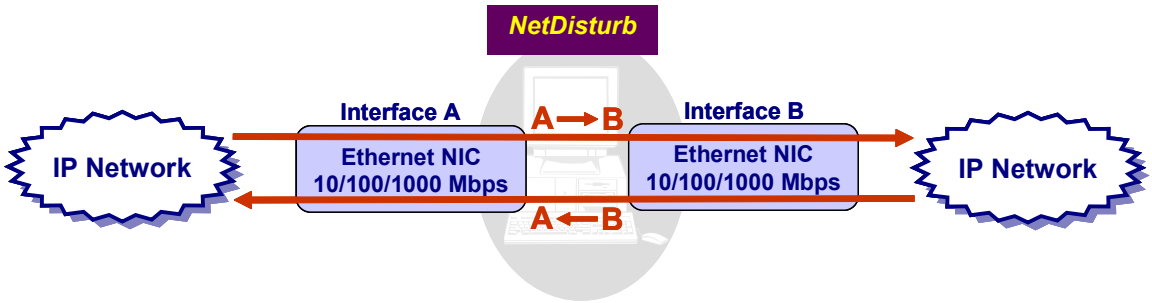
With **NetDisturb** you can define up to 16 masks, i.e. 16 IP flows. An additional item named "Other IP Flows" is in charge to handle all IP flows that have not been user defined. For this item no mask can be defined, but impairments can be applied.

NetDisturb manages up to 10 000 connections – all flows included.

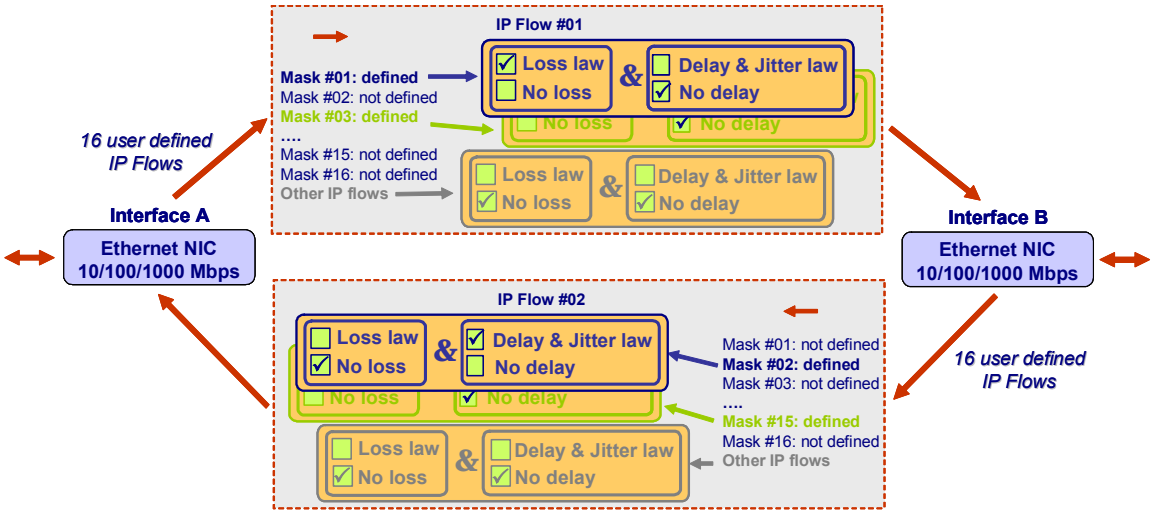
The client window below illustrates the management of IP flows by **NetDisturb**.



The graphical user interface represents the NIC cards as "Interface A" and "Interface B" as illustrated below.



For each direction A → B or B → A, 16 flows can be defined by the user. And for each IP flow, loss & duplication and / or delay laws can be applied as shown in the figure below.



In the above example, **NetDisturb** has been configured with the following parameters:

Direction A → B

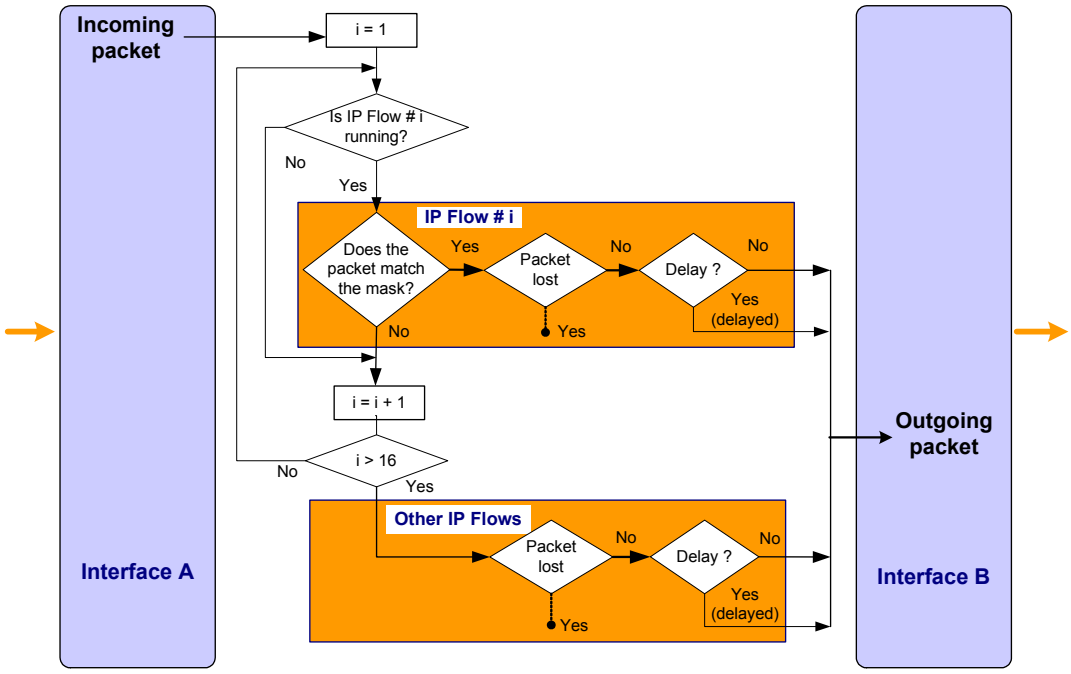
- the Mask #01 defines the "IP Flow #01", and a loss law is applied to the packets of this flow,
- the Mask #03 defines the "IP Flow #03", and a loss law is applied to the packets of this flow,
- As no loss and delay law is applied to the 'Other IP flows', all packets that don't match the masks #01 and #03 are relayed directly from the interface A to B.

Direction B → A

- the Mask #02 defines the "IP Flow #02", and a delay law is applied to the packets of this flow,
- the Mask #15 defines the "IP Flow #15", and a loss law is applied to the packets of this flow,
- As no loss and delay law is applied to the 'Other IP flows', all packets that don't match the masks #02 and #15 are relayed directly from the interface B to A.

How does it work?

We illustrate how **NetDisturb** handles incoming packets with the following scheme from the A interface to the B interface.



Depending of the active user-defined IP flows, **NetDisturb** identifies if the incoming packet belongs to an IP flow before applying loss or delay treatments.
If this packet matches with the mask of an IP Flow (IP Flow #i for example), then **NetDisturb** identifies if this packet must be lost/duplicated and/or delayed.
If this packet does not match any mask (a mask defines an IP flow), then **NetDisturb** applies the treatments for the 'Other IP Flows' and identifies if this packet must be lost/duplicated and/or delayed.
For each packet received on an interface, **NetDisturb** analyzes in order the masks from 1 to 16 before considering this packet to belong to the "Other IP Flows".

So **NetDisturb** can apply impairments on the IP flows defined by the user either unidirectional (A → B or B → A) or bidirectional (the same impairments apply for the two directions: A → B and B → A).

Introduction of a Trigger for the Mask

NetDisturb allows optionally using a trigger to condition the impairment with an event.
The Trigger is an intermediate step after the frame has been classified into an IP Flow and before the frame is impaired.

The Trigger includes various parameters:

- The activation condition based on the Ethernet frame content.
- The delay before applying the impairments
- The impairment duration (0 = no limit).
- The number of cycles for the trigger (0=unlimited) if the impairment duration is not null.

Thus two main categories of triggers are defined:
- the Trigger without limitation of duration to apply the Impairments
- the Trigger with a limitation of duration to apply the Impairments (and optionally a loop counter)

As soon as the activation condition is performed, the impairment on the IP flow can be immediate or delayed with a duration expressed in milliseconds (delay of impairment).
If the impairment is immediate, the frame that has triggered can be included or not (if the delay before impairment is null).
The impairment can be time limited according to a duration expressed in milliseconds.

When **NetDisturb** is running an IP flow with a defined trigger, four states are possible:

⇨ Waiting for the Trigger: the impairments do not apply. This state is the initial state of the Trigger.

⇨ The Trigger was found: the impairments still do not apply because a delay is defined before the impairments. This state changes to the next state when the activation condition is reached.

⇨ The Trigger is active: the impairments are applied.

⇨ The Trigger is finished: the impairments do not apply any more. This is the final state of the Trigger.

Note: a Trigger can remain active permanently if no duration limit was defined.

Packet impairments

Pre-defined loss and duplication laws:

- Loss: Constant law
  Parameter: number of packets
- Loss: Uniform law: $f(x) = dx/(\beta - \alpha)$
  Parameters: alpha, beta, threshold
- Loss: Burst Uniform law: $f(x) = dx/(\beta - \alpha)$
  Parameters: α, β, threshold(n), threshold(n + x), depth
- Loss: User-defined File
  Parameters: file name, threshold
- Loss: Percentage
  Parameter: percentage
- Loss: 1/N (1 packet is lost every N received packets)
  Parameter: range(N)
- Duplication: Percentage (send n times the received packet)
  Parameters: percentage, Min ≤ n ≤ Max
- Duplication: 1/M (duplicate 1 packet n times every M received packets). Parameters: range(M), Min ≤ n ≤ Max
- Duplication: Uniform $f(x) = dx/(\beta - \alpha)$
  Parameters: alpha, beta, threshold
- Loss (1/N) then Duplication (1/M): the loss law (1/N) is used first before the duplication law (1/M)

Pre-defined Delay & Jitter laws:

- Constant Delay & No Jitter
  Parameter = constant delay
- Constant Delay & Exponential Jitter law: $f(x) = \lambda e^{-\lambda x} dx$
  Parameters: constant delay, λ
- Constant Delay & Uniform Jitter law: $f(x) = dx/(\beta - \alpha)$
  Parameters: constant delay, alpha, beta
- Constant Delay & User File with Jitter values
  Parameters: constant delay, user file
- User File with Delay & Jitter values
  Parameter: user file
- Router Simulation & Constant Delay
  Parameters: IP throughput, max memory, constant delay

- Router Simulation & User File with Delay and Jitter values
  Parameters: IP throughput, max memory, user file

- Constant Delay & User File with Throughput and Duration values
  Parameters: constant delay, user file

Working modes

NetDisturb offers two working modes by applying impairments:
- Enable/Disable desequencing of the packets in a flow,
- Impairment laws apply to the IP flow or to each TCP/UDP connection of the IP flow.

These modes are used together.

For example, **NetDisturb** is set with the following modes:
- Enable desequencing of the packets in a flow
- Impairment laws apply to the IP flow

to simulate the Internet network with disturbed flows.

Another example is to use the following modes:
- Disable desequencing of the packets in a IP flow
- Impairment laws apply to each TCP/UDP connection of the IP flow

to disturb VoIP communications in the same way on an Ethernet network.

Enable/Disable Desequencing Packets
Impairment may introduce changes in the packet sequence – for example by introducing different delays for the packets of a flow.
One of the Ethernet characteristics is to keep packets received in order. Internet hasn't got this constraint regarding the packet order: some packets can use one route while others use another one, with the consequence the receiver may get packets unordered.
NetDisturb can simulate the Internet network (enable desequencing packets) or can react as Ethernet does (disable desequencing packets).

Impairment laws apply to the IP flow or to each TCP/UDP connection of the IP flow

NetDisturb can analyze IP packets to dispatch them into the TCP or UDP connection they belong to. This mode makes possible to apply the same impairment values to each packet of each connection.

For instance if the impairment has been defined with a loss law: lose the third packet for 10 packets received.

- *Impairment laws apply to the IP flow*

When this option is selected, every received packet matching the mask for this flow is considered to belong to the same flow. Processing is carried out in "continue". With the previous example of loss law (lose the 3rd packet on 10 received), **NetDisturb** will lose the 3rd packet for ten received packets whatever the TCP/UDP connection belongs to.

- *Impairment laws apply to each TCP/UDP connection of the IP flow*

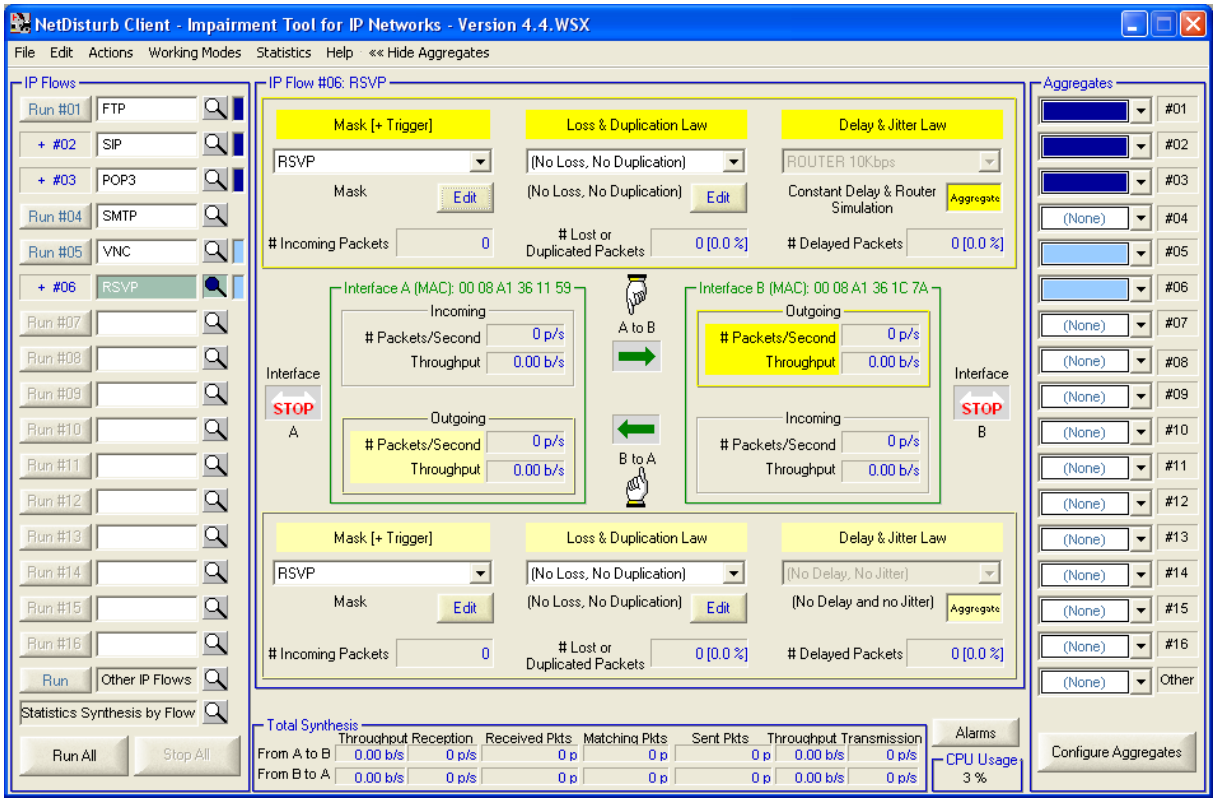
When this option is selected, **NetDisturb** analyses each received packet in order to associate this packet to a TCP or UDP connection already existing by using these parameters: protocol, IP addresses and port numbers. If the connection doesn't exist, a new one is created.
With the previous example of loss law (lose the 3rd packet on 10 received), **NetDisturb** will lose the 3rd packet for ten received packets of each TCP or UDP connection.

Up to 10000 connections can be handled simultaneously by **NetDisturb**.

IP Flows and Aggregates

Up to 8 aggregates of IP flows can be defined. An aggregate is a consecutive set of IP flows sharing the same Delay & Jitter Laws. All IP flows of an aggregate share only one aggregate's Delay & Jitter law (with one law per direction).
The IP flow order in the aggregate defines the priority of packets to delay. While the top IP flow packets get the highest priority, the other IP flow packets are queuing until there are no higher priority packets.
In the example illustrated below, two aggregates have been defined:

- the dark blue colored aggregate collects three IP flows (#01, #02 et #03)
- and the light blue aggregate collects the IP flows #04 et #05.



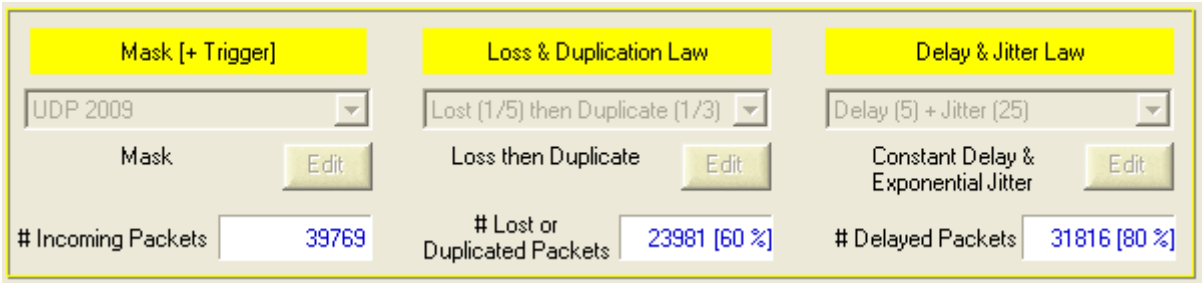
Statistics & Alarms

Different statistics are calculated and displayed by **NetDisturb**:
- for each IP Flow (and for both directions)
- Statistics synthesis by Flow
- Total synthesis & Alarms

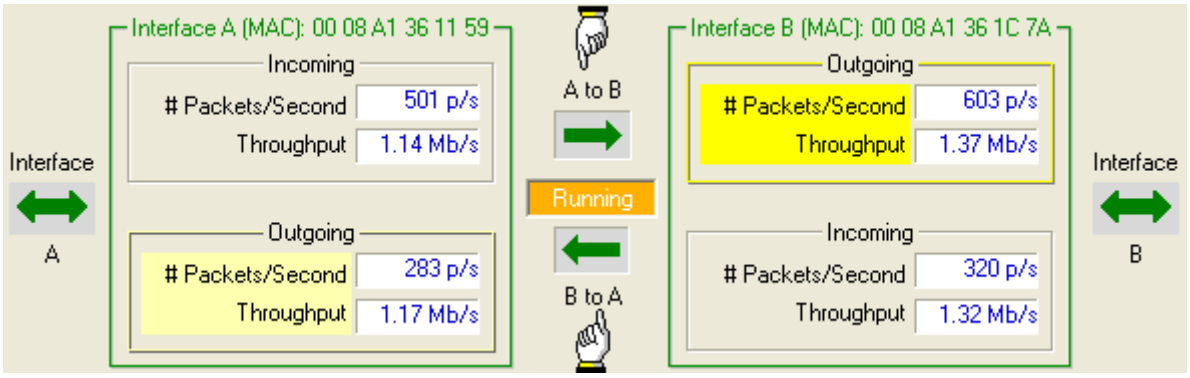
These statistics can be saved in a file for a later use.

Statistics for each IP Flow

For each direction (A → B or B → A) **NetDisturb** displays:
- The number of packets matching the mask
- The number and the percentage of lost or duplicated packets
- The number and the percentage of delayed packets



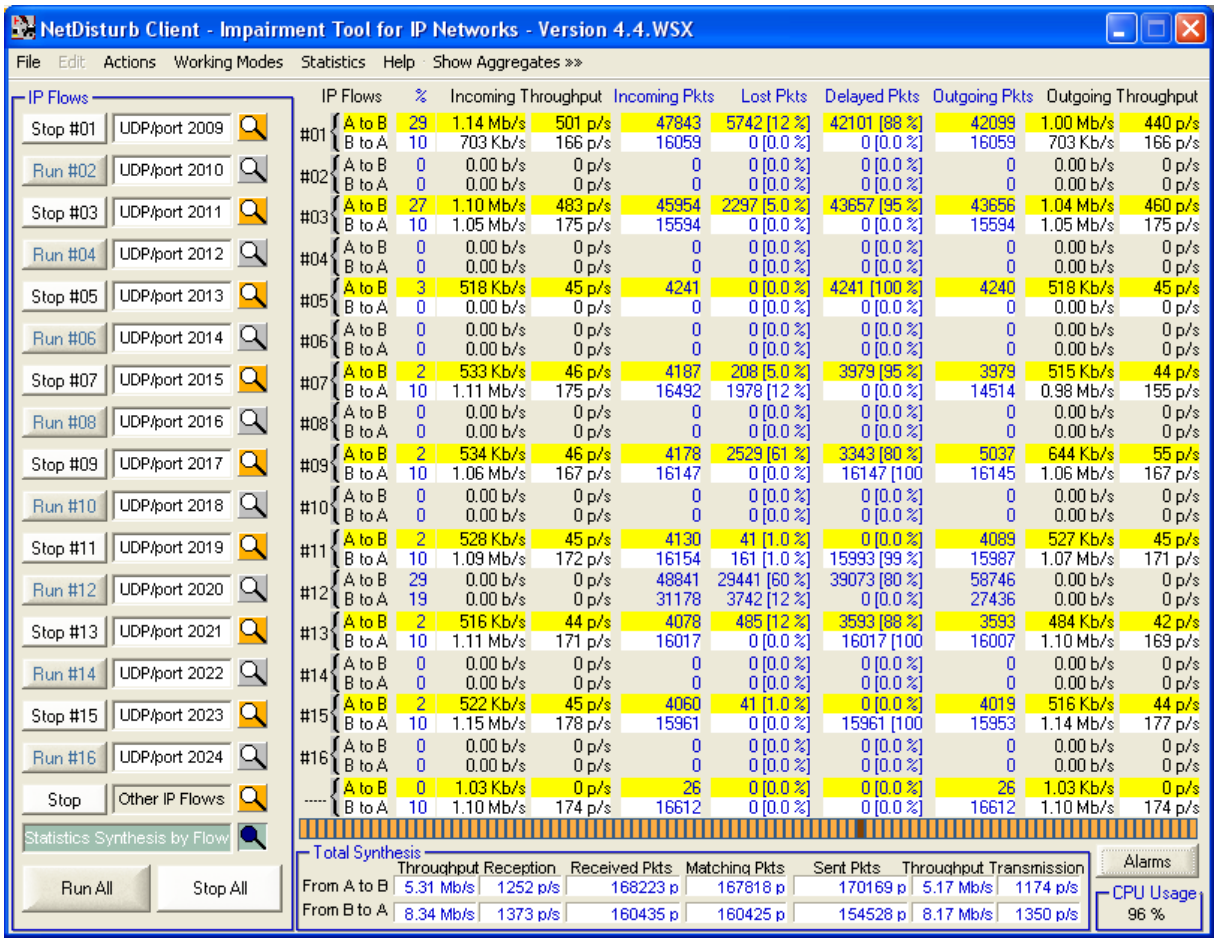
and a complete view of traffic statistics (number of packets and throughput) over the A and B interfaces as shown below:



Statistics Synthesis by Flow

The synthesis for all IP Flows displays for each flow and for each direction:

- The incoming throughput and number of received packets per second
- The number of packets matching the mask
- The number of lost packets
- The number of delayed packets
- The outgoing throughput and the number of sent packets per second



Statistics Synthesis by Flow - example

Total synthesis

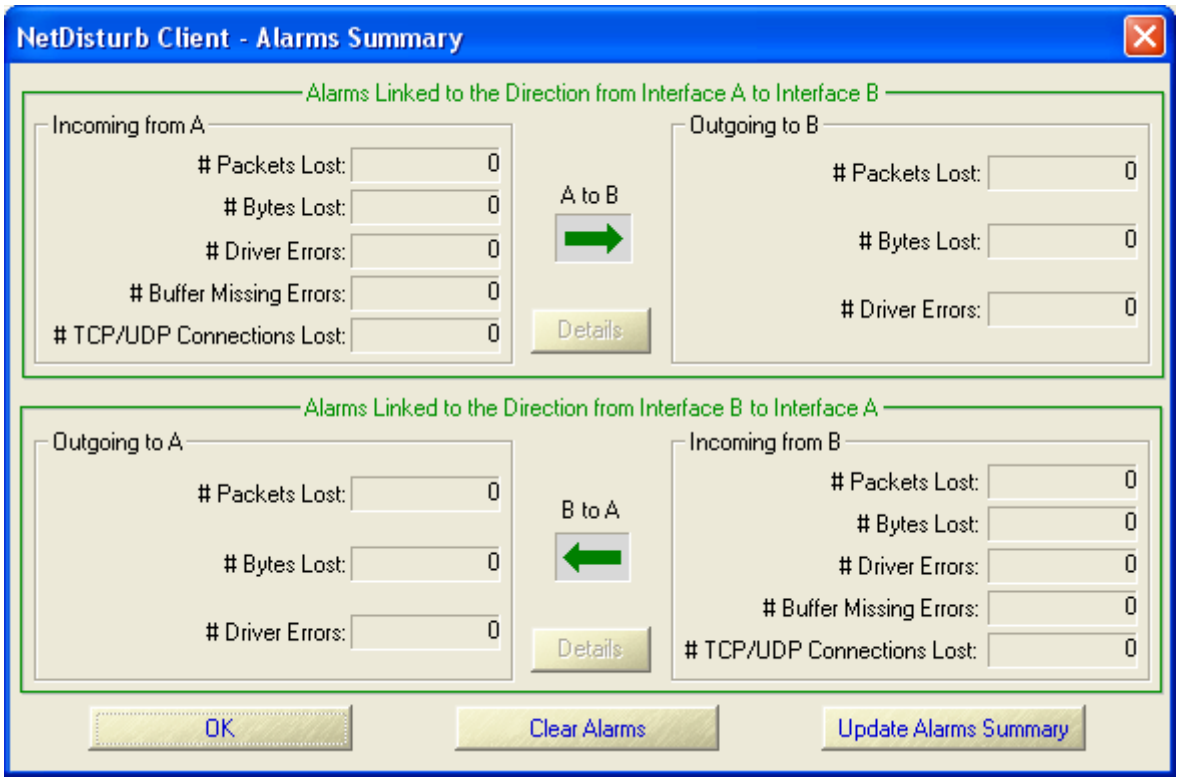
At the bottom of the Client window, the total synthesis displays the following parameters for both directions (A → B or B → A):

- Throughput and number of packets per second received
- Number of packets received
- Number of matching packets
- Number of packets sent
- Throughput and number of packets per second transmitted

Alarms

The alarms encountered by the **NetDisturb** driver can be displayed by the user and are classified per direction for both interfaces:

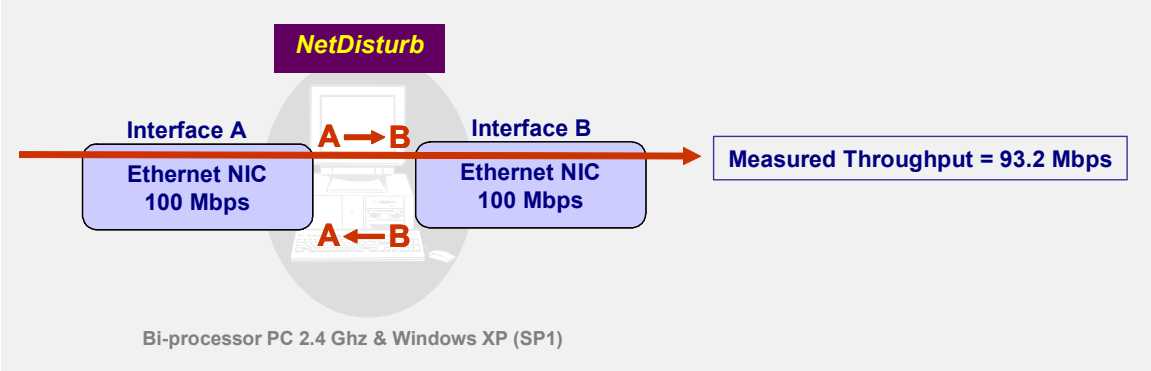
Incoming direction	*Outgoing direction*
• Number of lost packets • Number of lost bytes • Number of errors returned by the Driver at the Interface • Number of missing buffers to keep packets • Number of ignored flows (when the multi-flows option is active).	• Number of lost packets • Number of lost bytes • Number of errors returned by the Driver at the interface



Performances

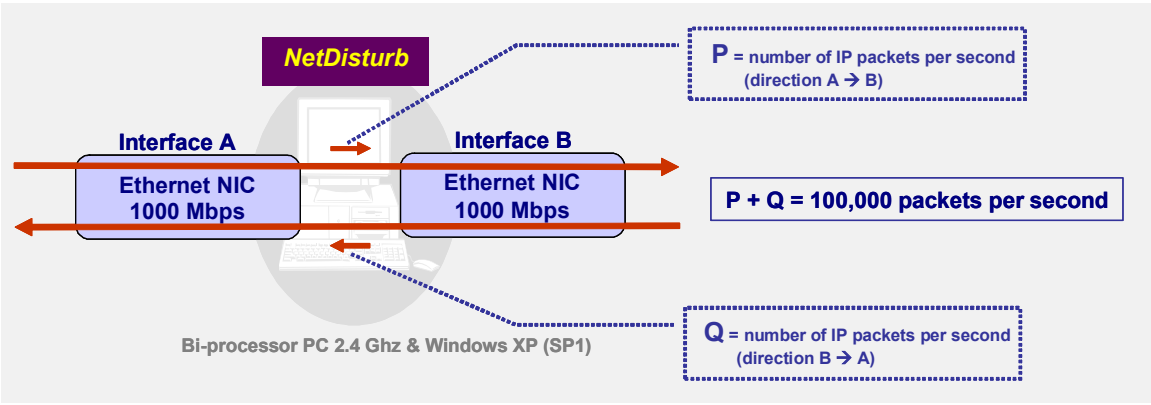
To illustrate the key performances of **NetDisturb**, 2 examples are presented hereafter (by using a bi-processor PC 2.4 Ghz with windows XP SP1).

Example 1: use of 2 Fast Ethernet NICs



NetDisturb is configured with 16 IP flows (no loss and no delay for each flow). With Fast Ethernet NICs, the throughput measured is 93.2 Mbps in one direction.

Example 2: use of 2 Gigabit Ethernet NICs



By using 2 Gigabit NICs, **NetDisturb** can handle up to 100,000 packets per second with 16 IP flows defined (for both directions).

These two examples show some performances of **NetDisturb**. This will avoid heavy investments in expensive hardware solutions.

Applications

- *Performance & Acceptance Tests:* Qualify and evaluate the behavior of IP equipments (phone, fax, gateway, etc.) and applications (audio and video streaming, etc.) on IP networks.
- *Configuration and control of IP Equipments for product verification and test:* Define different QoS levels in an Intranet or Internet environment to configure terminals, gateways and routers.
- *Test Laboratories:* **NetDisturb** provides repeatable QoS on different flows using configuration mode and values (loss, duplicate, delay) defined by the user, and so re-create real world problems in the lab.
- *Applications test:* **NetDisturb** allows testing applications such as Voice over IP, streaming audio and video, and other distributed applications.
- *Emulation of symmetric or asymmetric network conditions (Lan, Man, Wan):* latency, jitter, packet loss, bandwidth limitations, etc. to test IP applications (VoIP, streaming audio & video, etc.), services and products sensitive to various real conditions.

*Some publications mentioning the use of **NetDisturb***

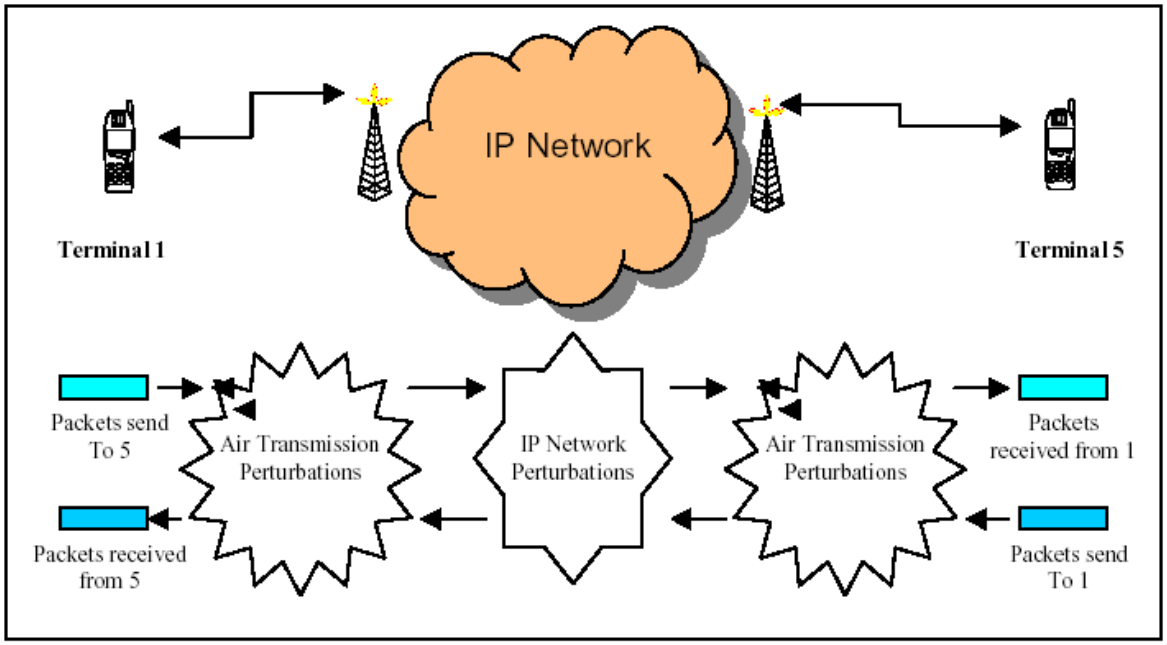
- The Communications and Information network Association of Japan (CIAJ) which represents manufacturers supplying network devices and terminals has published a report on 2002: Report on speech quality investigation of VoIP Terminals (gateways and IP phones):

http://www.ciaj.or.jp/tusin/pressrelease/voip_1e.html
"We adopted **NetDisturb**, … as a network simulator because of its ease of installation and operation in Windows".

- 3GPP Technical Specification Group Services and System Aspects TSG-S4

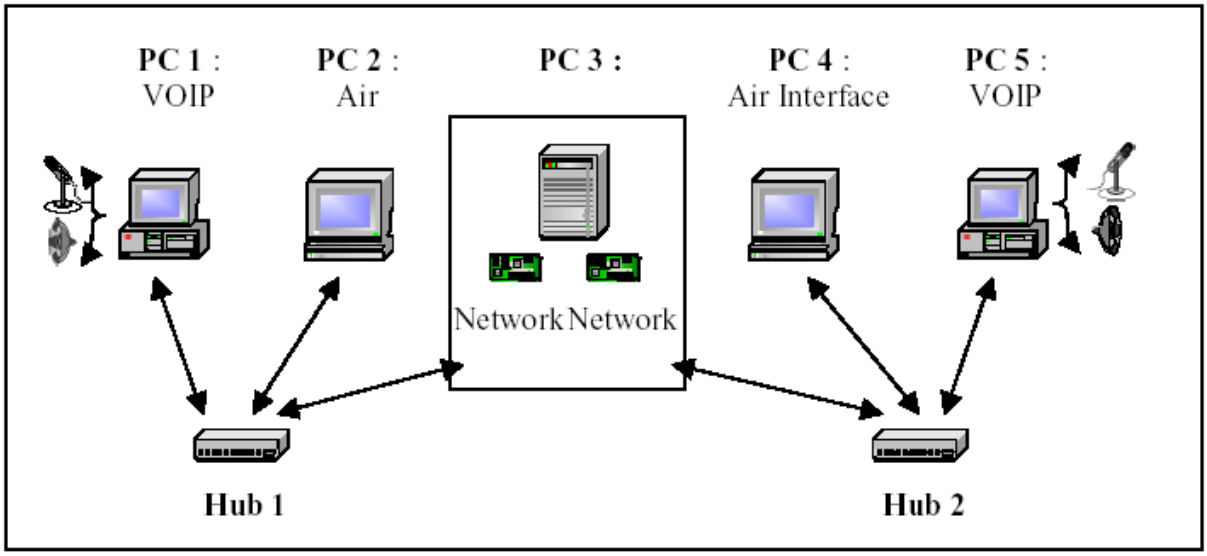
- Test Plan for the Adaptative Multi-Rate Wide-Band (AMR-WB) and Narrow-Band (AMR-NB) in packet switched networks.
- Test Plan for 3G packet switched conversation tests (comparison of quality offered by different speech coders over packet switched networks)
  NetDisturb is used as the simulated network.

The following illustrations describe the system that is simulated for these tests.



Packet switch audio communication simulator

This is simulated by using 5 PCs as shown below, with PC# 3 using the **NetDisturb** software as network simulator.



Simulation platform

Customer references

Present on the market since 1998, **NetDisturb** is used in more than 40 countries.
See some worldwide references of satisfied customers:
Alcatel, ANZ Bank, AT&T, Bell Canada, Cisco, Commtech Wireless, Department of Defense, Equant, France Telecom, Gensight, Global Crossing, Iwatsu, Juniper, Motorola, Nortel Networks, NEC, NTT, Panasonic, Philips, PIKA Technologies, Polycom, Psytechnics, Raytheon, Schlumberger, Scopus, Tekelec, TF1, Toshiba, UTStarcom, WL Gore, Xerox, etc. as well as many universities and telecom institutes.

Conditions of use

NetDisturb is licensed on a per workstation basis. You will need to purchase a separate license for each machine that you install it on.
Each licensed copy of the software installed on a workstation has a unique Site Code that requires the corresponding unique Site Key to be entered before being operational.

Delivery

Includes CD with documentation, printed installation guide, technical support and software maintenance (including major and minor software upgrades) for a period of twelve months from the date of purchase.

To download an evaluation of **NetDisturb** please visit us at:
http://www.zti-telecom.com/pages/main-ip.htm

PART 2 What's new in NetDisturb version 4.4

This part is a general overview of new features and improvements provided with **NetDisturb** version 4.4 and important information to upgrade from previous versions.

More details regarding features and improvements included in the different versions of **NetDisturb** can be found in the version.txt file located in the installation directory (default settings: C:\Program Files\NetDisturb).

2.1 New features in NetDisturb v4.4 (from NetDisturb 4.2)

- IPv6 Support
- Additional Delay & Jitter law to enable the Throughput variation over the time
- New Aggregate feature to associate multiple IP Flows to share on Delay & Jitter law.
- Additional Frame Ethernet mask features to define a mask:
  - The 'Equal' or 'Different' operator is a new characteristic of each parameter of the Mask to extend the filter definition,
  - List of protocols.
  - List of Differentiated Services available with both IPV4 and IPv6.
  - Filter IPv4, IPv6 or both IP versions.
- Start and Duration of Impairments based on the new Trigger definition. The Trigger is an additional parameter to the mask.
- New User-define file format helping to add comments in the file.

*The contexts created with version 4.2 and version 4.3RC3 are reused automatically. When saved, they get the new **NetDisturb** v4.4 file format.*

2.2 Upgrading from versions 4.2 and 4.3

You shouldn't uninstall the previous version of **NetDisturb** to keep your license scheme.

By upgrading from a version 4.2 or 4.3, you keep your existing User-defined file and the new contexts you created.

NetDisturb version 4.4 requires Acrobat Reader. Please see paragraph 2.4 for more details.

2.3 Upgrading from versions 4.1 and older

You don't need to uninstall the previous version of **NetDisturb** to keep your license scheme. However, this license will not enable you to use **NetDisturb version 4.4**, because the license date of version 4.1 and previous is too old.

- For any question or further information regarding the license upgrade, please contact ZTI :
  Email: contact@zti-telecom.com or contact@zti.fr
  Phone: +33 2 96 48 43 43
  Fax : +33 2 96 48 14 85

2.4 Acrobat Reader version compatibility

To access the **NetDisturb's** help, Acrobat Reader is required. **NetDisturb** supports Acrobat Reader version 4.01 to 7, that have been tested successfully.

If your Acrobat Reader's version is too old, you can use the Acrobat Reader's version from the **NetDisturb's** CR-ROM or download it straight from the Adobe web site: www.adobe.com.

PART 3 Install NetDisturb

NetDisturb is supported on the following platforms: Windows XP Home or Professional, Windows 2000 Professional or Server, Windows NT 4 Service Pack 6 Workstation or Server.
The minimum screen resolution is 1024 x 768 and the DPI setting should be "Normal size (96 DPI)".

If you have the **NetDisturb** CD-ROM version, please refer directly to paragraph 3.2.

**** To run NetDisturb your computer's screen resolution must be at least 1024 X 768 and the DPI setting should be set up with the "Normal size (96 DPI)" value.***
**** To install NetDisturb for Windows NT4, 2000 and XP, you must log on with your administrators rights.***

The default settings of **NetDisturb** come with a 15-day limited license. When it reaches the deadline, **NetDisturb** stops running. Go to PART 4 for more information about the license program.

3.1 How to install the software downloaded from the Internet

The installation procedure is a standard installation program.

*Please note that the **NetDisturb** installation procedure will be different in the last part, depending on the target Operating System: Windows NT4 or Windows 2000 and Windows XP.*

- Before to proceed with the **NetDisturb** Setup, please be sure your system does meet the following minimum requirements:
  - ⇨ OS supported: Windows NT4 (SP6 at least), Windows 2000 or XP.
  - ⇨ Minimum screen resolution: 1024 x 768
  - ⇨ Your PC needs at least 2 NIC already installed, configured and fully operational.

- If you have downloaded the software from our website, you have downloaded the file **NetDisturb**.zip containing the Setup_NetDisturb.exe file and the related documentation.
  You must first unzip this file in a temporary directory.

  NetDisturb is composed of two parts: **NetDisturb** Client and **NetDisturb** Server. **This setup will install both the Client and Server parts on the same system.**

- Then run "Setup_**NetDisturb**.exe" from this temporary directory to launch the setup procedure and follow instructions on the screen.

The default settings install **NetDisturb** in the following directory:
C:\Program Files**NetDisturb** with the following subdirectories:

C:\Program Files**NetDisturb**
C:\Program Files\ **NetDisturb** \Client
C:\Program Files\ **NetDisturb** \Driver
C:\Program Files\ **NetDisturb** \Server
C:\Program Files\ **NetDisturb** \Server\Script

At the end of setup, a manual driver installation is required if Windows NT4 is used. In that case, please refer to paragraph 3.1.2.

Otherwise, the only necessary operation is to uncheck protocols from NICs used with **NetDisturb**.

3.1.1 NetDisturb Driver Installation for Windows 2000 or XP

The **NetDisturb** driver sets in the kernel of Windows 2000 or XP. The **NetDisturb** driver is installed on the top of the driver for each Network Interface Card (NIC) installed in your PC. For Windows 2000 or XP, the **NetDisturb** driver is considered as a protocol. The **NetDisturb** driver handles the exchanges between two NICs.

The setup procedure realizes the installation of the **NetDisturb** driver transparently. The **NetDisturb** driver is mapped on top of each Ethernet or wireless NIC if the driver of the NIC is NDIS compatible.

The **NetDisturb** driver linked to the selected NICs remains available transparently: it doesn't appear in the protocol list.

There is still an important manual operation you have to do before using NetDisturb:

1. In order to avoid unexpected traffic generated by the protocol stack (TCP/IP, Client or Microsoft Networks, etc.) on the NICs that **NetDisturb** will use, you have first to unselect all protocols.
2. To unselect protocols from a NIC used by **NetDisturb**, use the "Control Panel/Network and Dial-up Connections" or the "Control Panel/Network Connections" program and uncheck all protocols.

3.1.2 NetDisturb Driver Installation for Windows NT4

At the end of the setup, after the files have been copied to the system:

- A text file is automatically opened to explain the next step: installation of the **NetDisturb** driver on the system
- The control panel is automatically opened in order to proceed with the driver installation.

The **NetDisturb** driver sets in the kernel of Windows NT. This driver must be installed over the driver of network cards. For Windows NT, it is considered as a protocol. The **NetDisturb** driver goal is to handle exchanges between two networks interface cards (NIC).

The **NetDisturb** driver is a protocol named 'Disturbing Ethernet Driver over NDIS'.
The **NetDisturb** driver installation is carried out as any usual network driver installation. It must be installed after the network cards drivers. Different protocols can be bound to your NIC. The **NetDisturb** Client and **NetDisturb** Server need a TCP protocol stack for data exchange. So before installation, check that your Windows NT4 computer gets 2 NIC installed and a TCP stack installed.

To install the **NetDisturb** driver, you have to use the Windows control panel and select the following items:

1. Choose "Network" icon,
2. Choose "Protocols" tab,
3. Click on "Add"
4. Choose "Have Disk..."
5. Type the folder where following files are located: OEMSETUP.INF and DISTURB.SYS (by default: C:\Program Files**NetDisturb**\Driver) and press **'OK'**
6. Then select the "Disturbing Ethernet Driver over NDIS " and click on "OK"
7. The item 'Disturbing Ethernet Driver over NDIS' appears in the protocol list of 'Protocols' tab
8. Disable the other protocols bound to the network adapters as follows:
   * Choose "**Bindings**" tab,
   * In the list box select "**show bindings for all adapters**",
   * Disable all protocols bound to each Network adapter used by **NetDisturb** except: "**Disturbing Ethernet Driver over NDIS**" (you need to do it for the 2 used network adapters)
   * Disable "**Disturbing Ethernet Driver over NDIS**" for all other adapters

9. As a TCP stack is required for the **NetDisturb** Client and Server exchanges, you must bind the TCP protocol to another one adapter - for example to a modem or another adapter.

This is an example to add a TCP/IP stack onto a modem, which is not necessarily physically connected to your PC. The procedure is as follows:

a) Add a modem via Control Panel / Modem, select "**add**", then "**don't detect my modem, I will select it from a list**" and click on "**Next**"
b) Select any standard modem from the Standard Modem Types manufacturer list (for example Standard 14400 bps modem). Click on "**Next**"
c) Select a port and click on "**Next** "
d) Windows NT should present you a dialog box indicating the end of modem installation

When you have finished using the control panel, press **Close** to save all changes.

Your system is now configured and **you need to reboot your PC** to take changes into account.

3.1.3 Start Menu Shortcuts Created

Start > All Programs > **NetDisturb**
- ⇨ **1) NetDisturb Server**
- ⇨ **2) NetDisturb Client**
- ⇨ **License help**
- ⇨ **Uninstall NetDisturb**
- ⇨ **User Guide**

When launching a **NetDisturb** trial version for the first time, a message is displayed showing the remaining days of use (for example, 15 days left out of 15 in the following example):

To enter your unlimited license,
please refer to PART 4 (Software License Configuration)

3.2 How to install the software from the CD-ROM

The installation procedure is a standard installation program. On the CD-ROM, you will find the "Setup_NetDisturb.exe" file.

This setup will install the NetDisturb Client and the NetDisturb Server on the same machine.

Run this setup and follow the instructions as described in the previous paragraph.

On the CD-ROM, a second setup allows installing the **NetDisturb** Client on a machine. This is useful if you want to install the **NetDisturb** Server and the **NetDisturb** Client on two different machines.

To install the **NetDisturb** Client on a machine (Windows 95, 98, NT4, 2000 or XP), run "Setup_NetDisturbClient.exe" and follow the setup instructions to proceed with the installation.

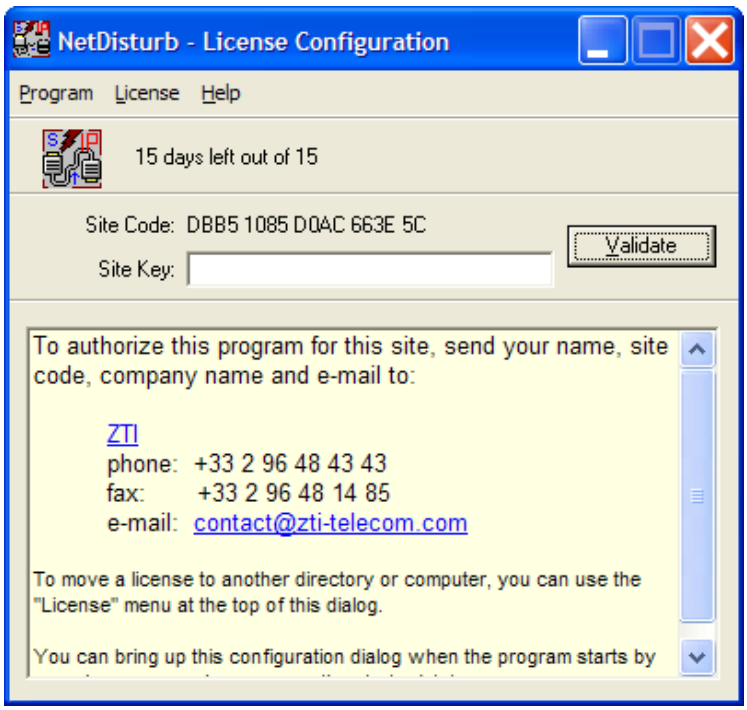
PART 4 Software License Configuration

*Note: This software is licensed on a per workstation basis. This means that you will need to get a separate license for each machine that you'll install it on. Each licensed copy of the software installed on a system has a unique **Site Code** which requires a corresponding unique **Site Key** to be entered before the tool is operational (except for a trial version: a duration of 15 days is automatically enabled at the first installation of the software. If you try to install the software again, the license program will disable the trial period).*

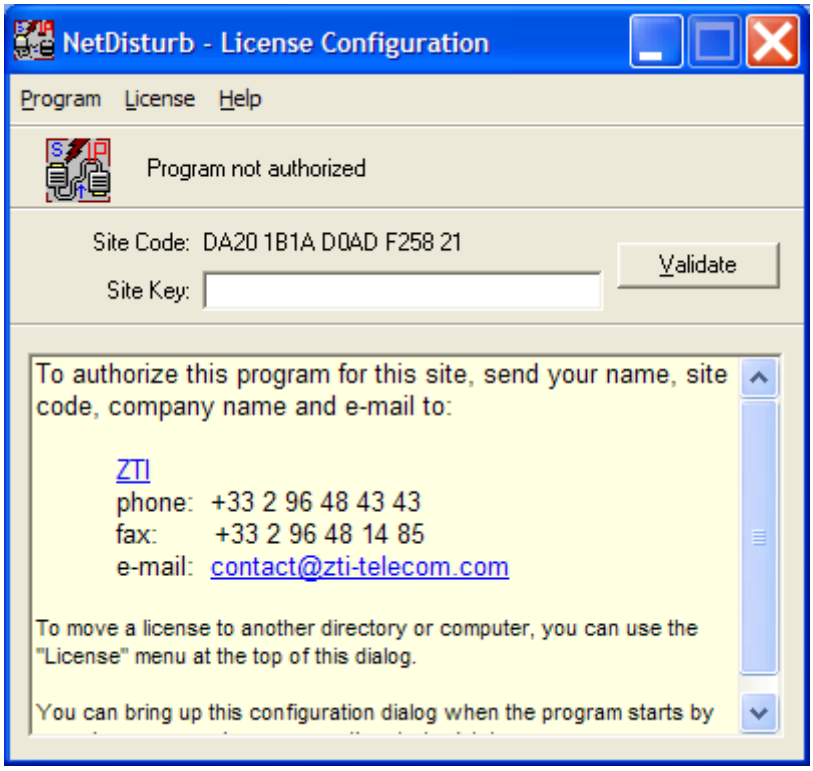
4.1 How to configure a license

If you wish to configure your license before the trial period ends, press **Enter** just after launching the **NetDisturb** Server when the following message is displayed:

You will then see the following license configuration window:



*At the end of the trial period when you launch **NetDisturb** Server, the same license configuration window appears, but saying "Program not authorized" instead of showing the remaining days of use.*

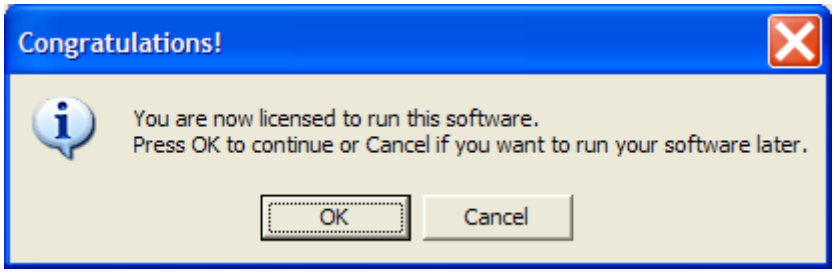


To get the **Site Key** and obtain an unlimited version, please send an email to contact@zti-telecom.com or contact@zti.fr with the following information:

a. The **Site Code** (you can copy and paste the Site Code displayed in the license window)
b. The name of the software: **NetDisturb**
c. The OS used
d. Your company's name
e. Your name and phone number
f. The purchase order's number and date of purchase

We will then email you the **Site Key**. You can now close the license's window.

After you have received the email with the **Site Key**, open the license configuration window again by pressing the Enter key as explained before.
Copy the Site Key in and then click "Validate".
After validation of the Site Key, you will get the following message:



- ⇨ Important: one **Site Code** is associated with one **Site Key**, and only one. A **Site Code** is unique for each PC installed. For security reasons, as soon as you validate a **Site Key** (trial or unlimited), the license program generates a new **Site Code** automatically.
- ⇨ For any question or further information, please contact our technical support:
  Email: support@zti-telecom.com or support@zti.fr
  Phone: +33 2 96 48 43 43
  Fax : +33 2 96 48 14 85

*When you launch **NetDisturb** Server with an unlimited license, you will see the following window:*

4.2 License Transfers

> ***A license transfer is not a duplication of any type.***
> ***Please contact ZTI or your authorized distributor for site license information and for several licenses purchase.***

Licenses can be transferred using one of the following methods:

- ⇨ **Direct transfer:** move the license to another directory on the same PC or between two PCs of a same network.

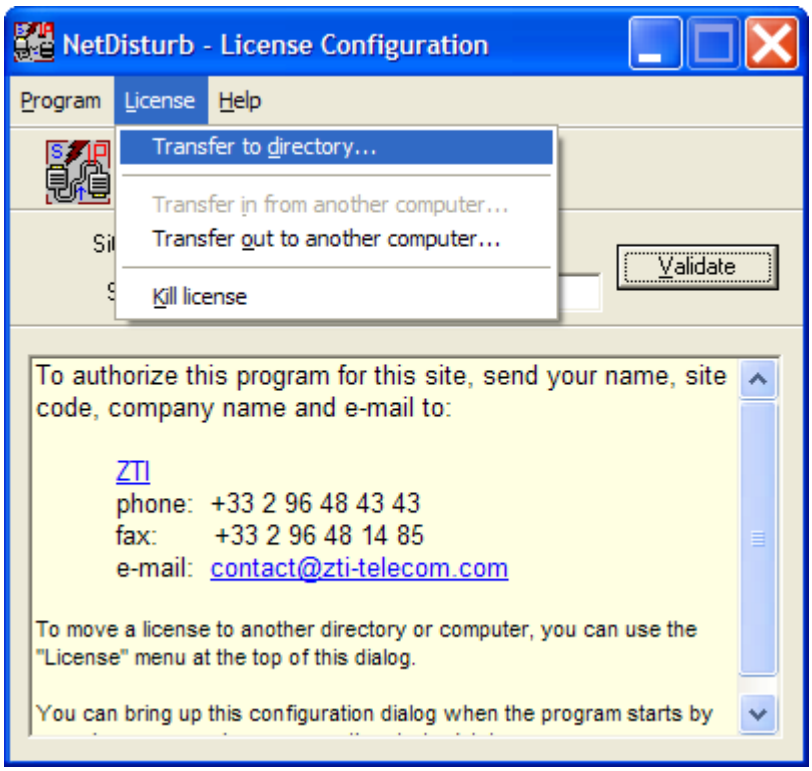
- ⇨ **Transfer by media:** move the license from a source PC to a target PC by using a floppy disk or USB key.

4.2.1 Direct Transfer: move the license from one local directory to another

This transfer mechanism must be used to move a license in two cases:
- From a source to a target directory of the same PC
- From a source to a target directory of networked PCs

- First, copy the program (copy the **NetDisturb**'s folder) to the target directory.
  For example from "C:\Program Files\ NetDisturb" to "C:\Temp\NetDisturb"

- Then run the program from its original directory (*from "C:\Program Files\NetDisturb"*). When the license configuration window appears, press **Enter** and select "License > Transfer to directory …" in the license menu as shown below:



- Provide the path name of the target program (*for example C:\Program Files\NetDisturb\NetDisturbServer.exe*).
  The license is now transferred to the new directory.

4.2.2 Transfer by Media (floppy disk or USB key) from a source PC to a target PC

A floppy disk or USB key is needed for this kind of transfer.

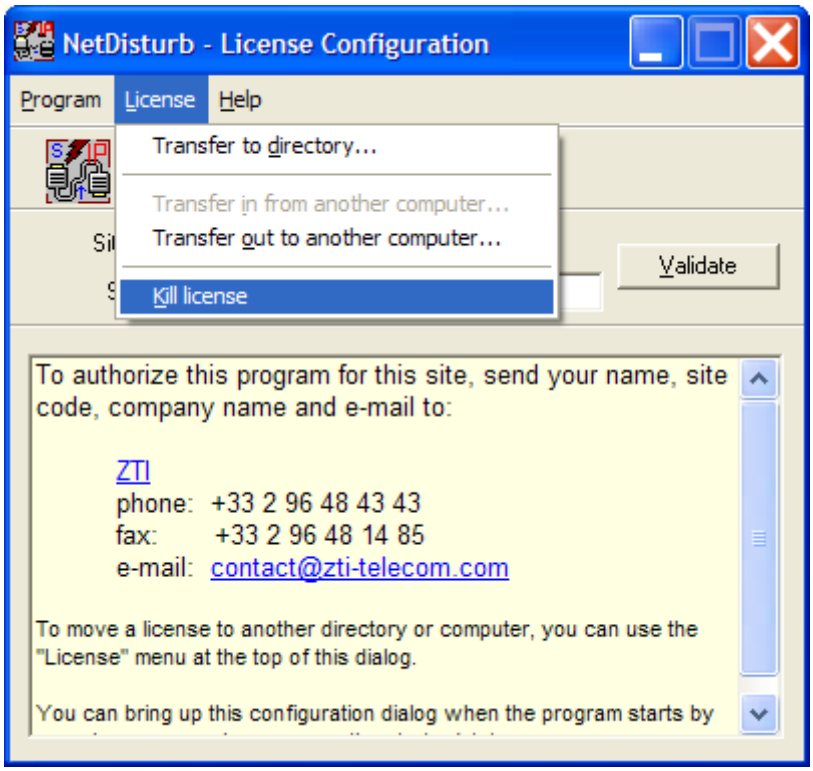
To transfer the license from the source PC (PC #1) to the target PC (PC #2), proceed as described in the following order:

1) First install the program on the target PC (PC #2).

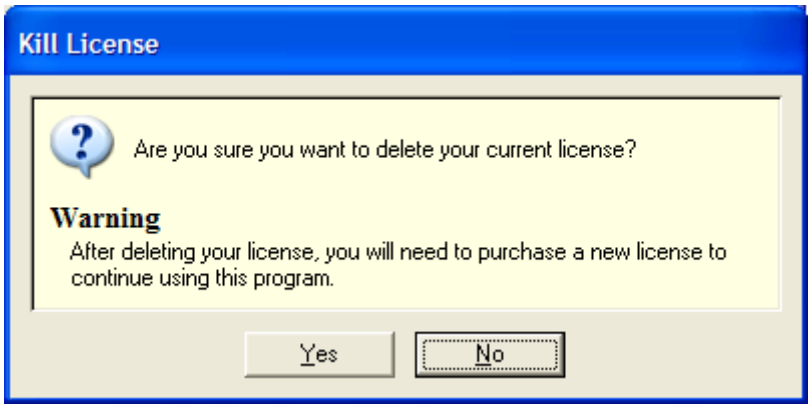
2) Run the software on PC # 2 and delete the trial license in order to get an unauthorized license on this PC.
If the "Transfer in from another computer …" item of the license menu is disabled, you must kill the license.

How to kill a license?

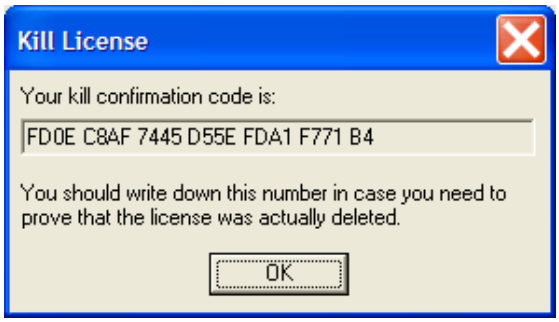
When the license configuration window appears, press **Enter** and select “License > Kill license” in the license menu.



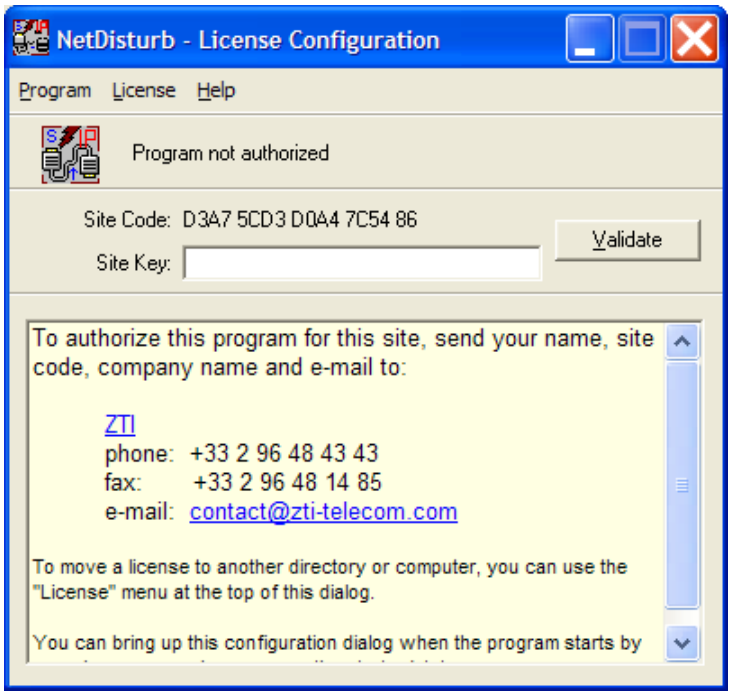
A message box will appear:



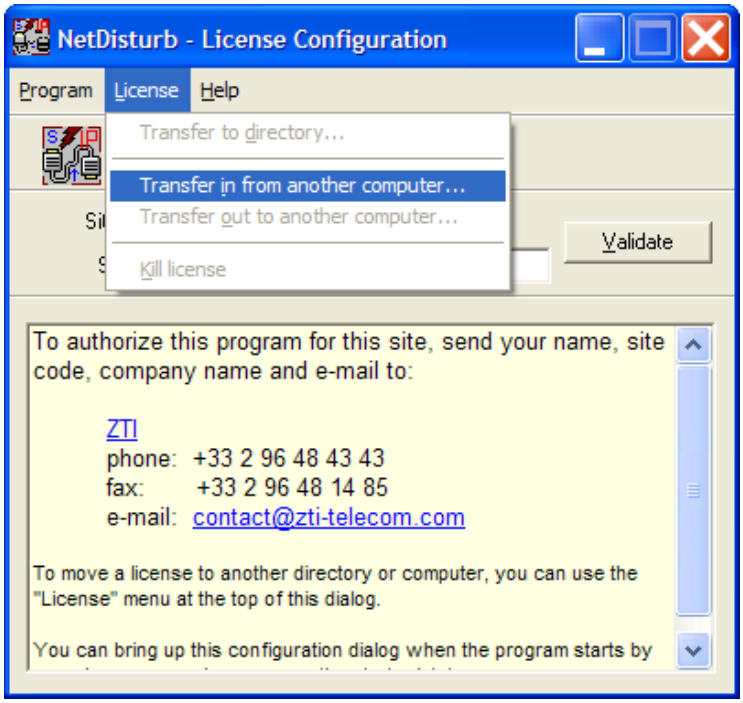
Press 'Yes' to kill the license and a confirmation code is displayed:



Click 'OK' and the license window displays now "Program not authorized":

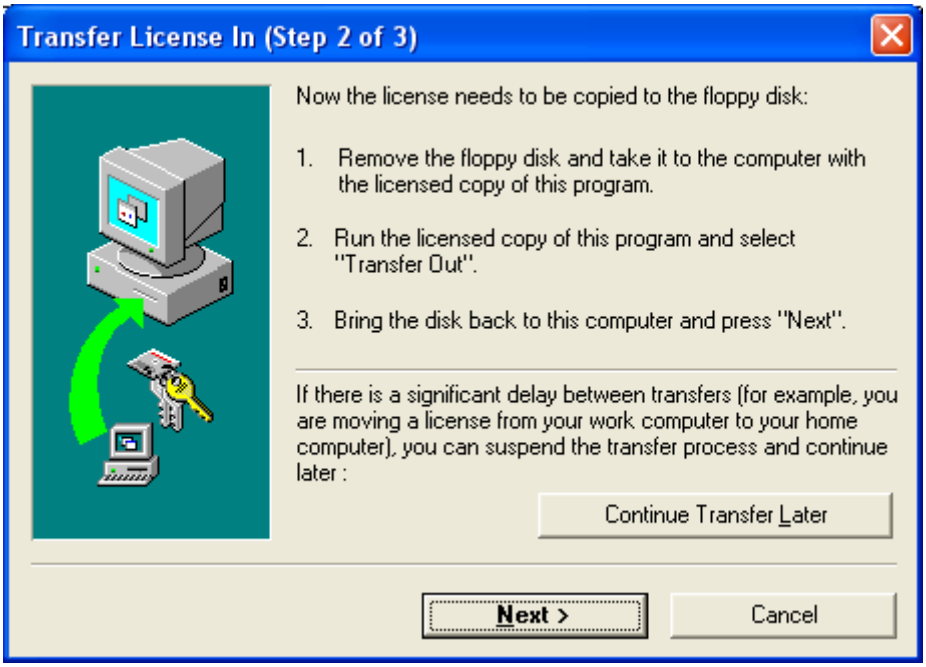


3) Select “License > Transfer in from another computer …” from in the license menu:

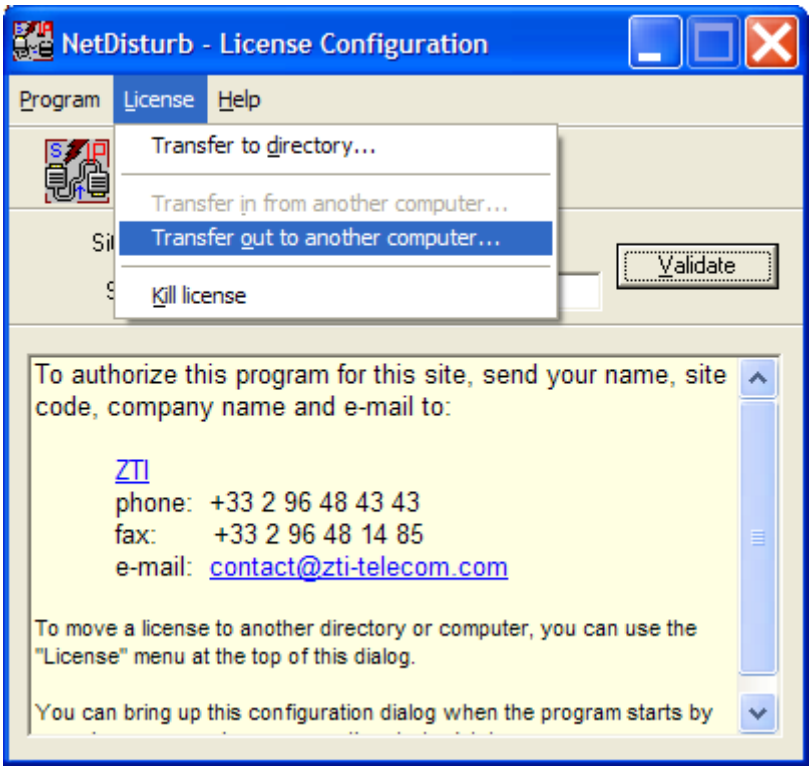


The "Transfer License In (Step 1 of 3)" window is displayed:

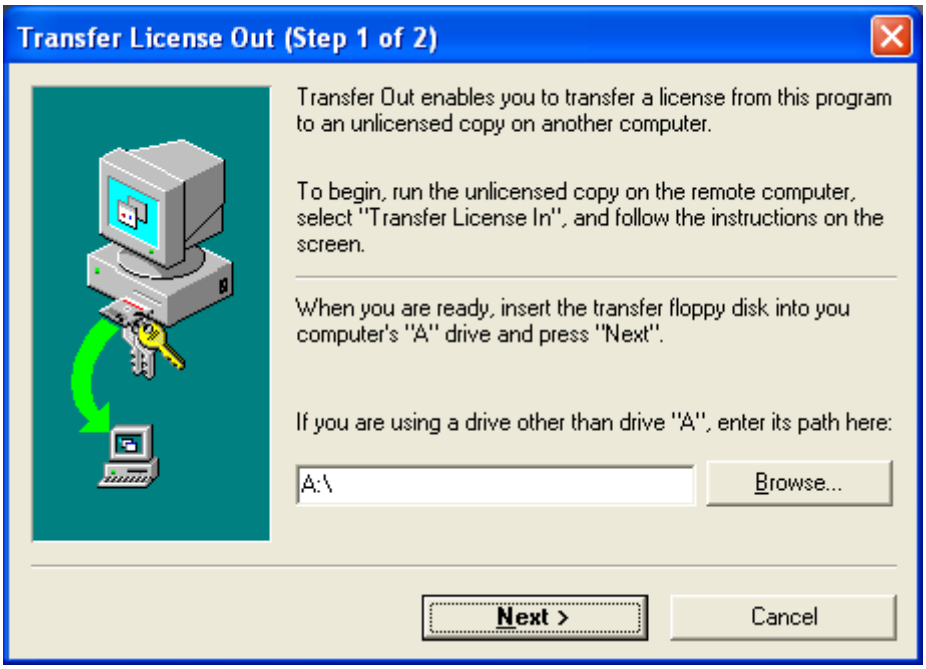
4) Insert a floppy disk or use a USB key as requested in step 1 of 3 and specify the path.
Then press “Next >”: the "Transfer License In (Step 2 of 3)" window is displayed:



5) Go to the source PC (PC #1) and insert the media (floppy disk or USB key). Then start the program on PC #1. When the license configuration window appears, press **Enter** and select “License > Transfer out to another computer …” as shown below:

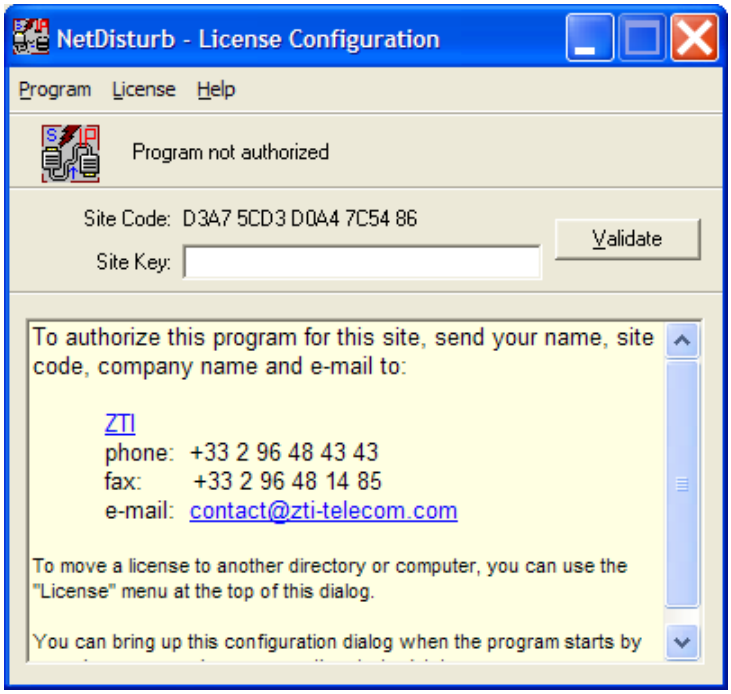


The following window is displayed:



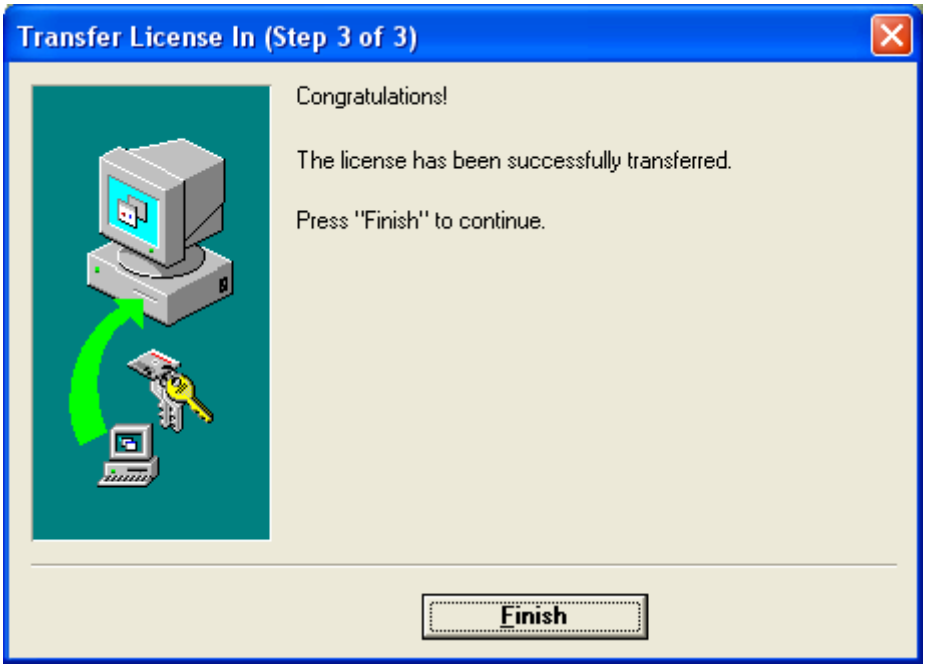
Input the media path (floppy disk or USB key) and then press "Next >".

When the license is put on the media, you get the “Program not authorized” message:



You can check that the license is not available anymore on the source PC since the LanTraffic V2 software license is on a workstation basis.
Contact us to get information on site license (contact@zti.fr or contact@zti-telecom.com).

6) Remove the media from PC #1 and return to PC #2.
Click the 'Next' button on the step 2 of 3 of the “Transfer license in” window (on PC #2) to complete the transfer.
The unlimited license key is now transferred from the source PC to the target PC, and you get the following message:

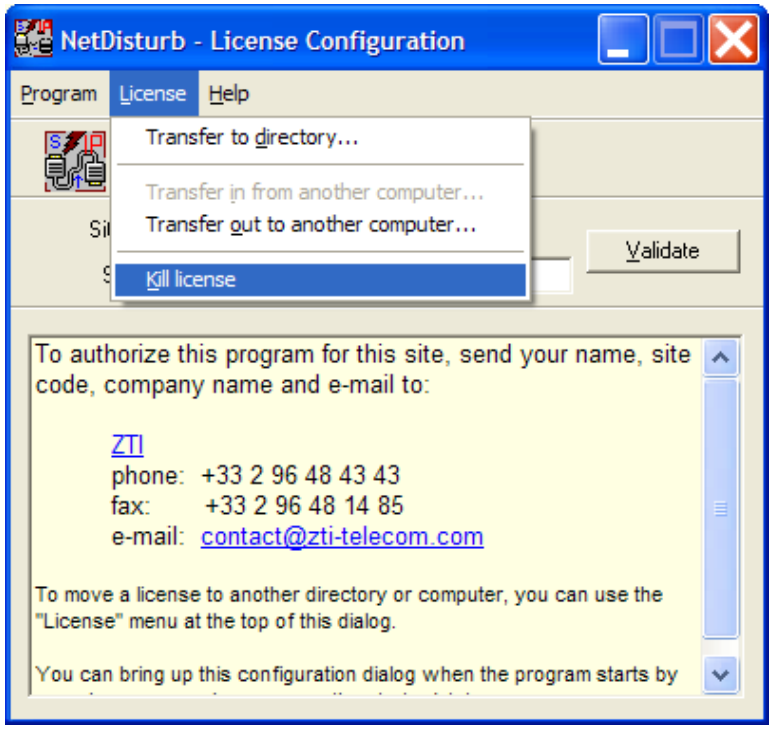


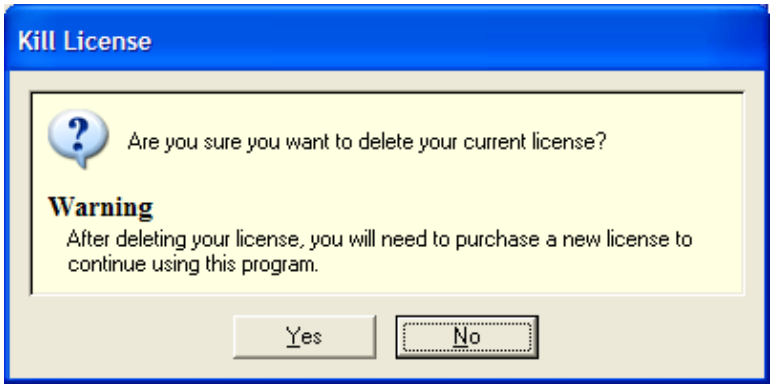
Click Finish to continue.

4.3 How to kill a license

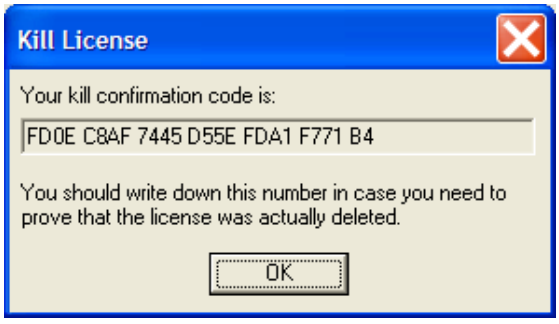
If you would like to transfer an unlimited license key onto a PC where a trial period is still active, you should first delete the active trial period. If you don't delete the active trial period, you will not be able to transfer an unlimited license.
To delete the trial license, you should proceed as follows:

- From the license configuration window, select "License > Kill License" in the license menu as shown below:





- Press 'Yes' and your license is now deleted. Please write down the kill confirmation code. This code may be requested by ZTI.



PART 5 Uninstall NetDisturb

The uninstall procedure is a standard uninstall program.

To uninstall **NetDisturb** select “Uninstall NetDisturb” in the “Start > Programs > NetDisturb” menu.

How to uninstall the **NetDisturb** driver:

⇨ **Windows 2000 or XP**

All software components installed by the installation procedure are removed during the uninstall procedure including the **NetDisturb** driver.

⇨ **Windows NT4**

In Windows NT4, the Disturb driver is not removed by the un-installation procedure because it has not been added by the installation procedure.

At the end of the uninstall Procedure, a text file is automatically opened in order to explain how to uninstall the **NetDisturb** driver when you are using Windows NT4.

By using the Control Panel, run "Network" and then choose the "Protocols" tab.
Select the driver "Disturbing Ethernet Driver over NDIS" and click on "Remove".

Then, you must restart your PC.

PART 6 Run NetDisturb

As **NetDisturb** is composed of 2 parts (**NetDisturb** Server and **NetDisturb** Client), you need to run these two programs with the following order:

1. **NetDisturb Server**
2. **NetDisturb Client**

6.1 First Run

1) Run first the NetDisturb Server

Use the Windows start menu:

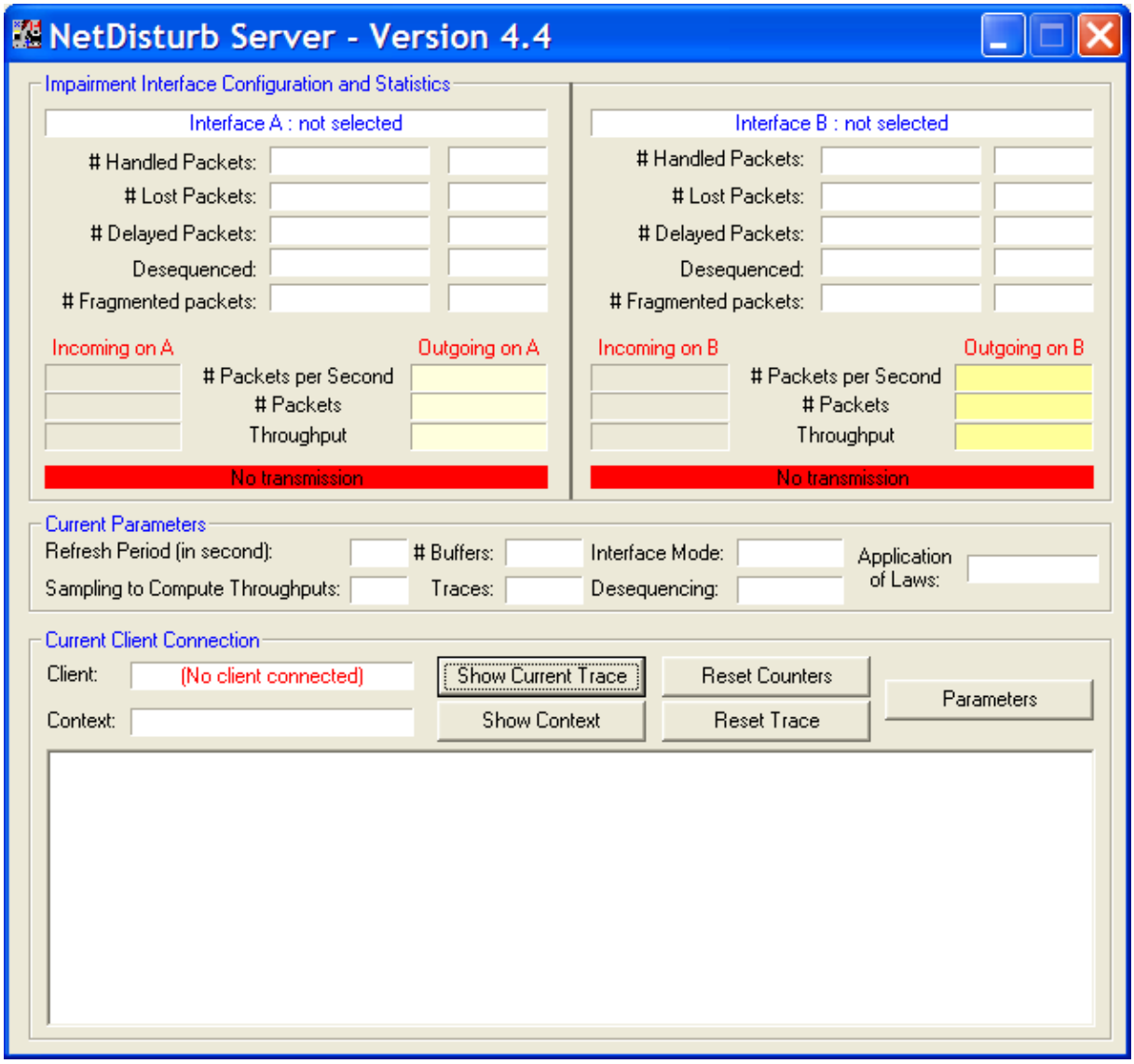
Start ▶ Programs ▶ **NetDisturb** ▶ 1) **NetDisturb** Server

After a few seconds and depending of your license,
you will get one of the following license windows:

Limited license	***Unlimited license***
	

Press **Enter** **only if you need to configure your license**,
If you don't, allow a few seconds for the main window of **NetDisturb** to open.

When you run the **NetDisturb** Server for the first time, the default window is displayed:



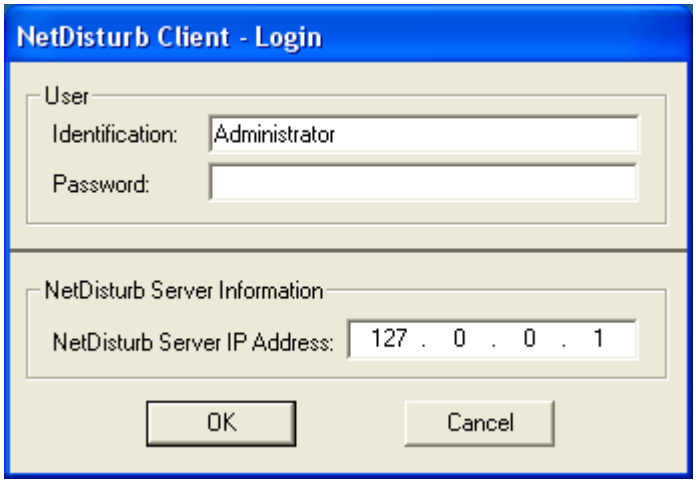
No board (or NIC) has been selected: the **NetDisturb** Client must be used to select the network interfaces (or NICs).

2) Then run the NetDisturb Client

Use the Windows start menu:

Start ▶ Programs ▶ **NetDisturb** ▶ 2) **NetDisturb** Client

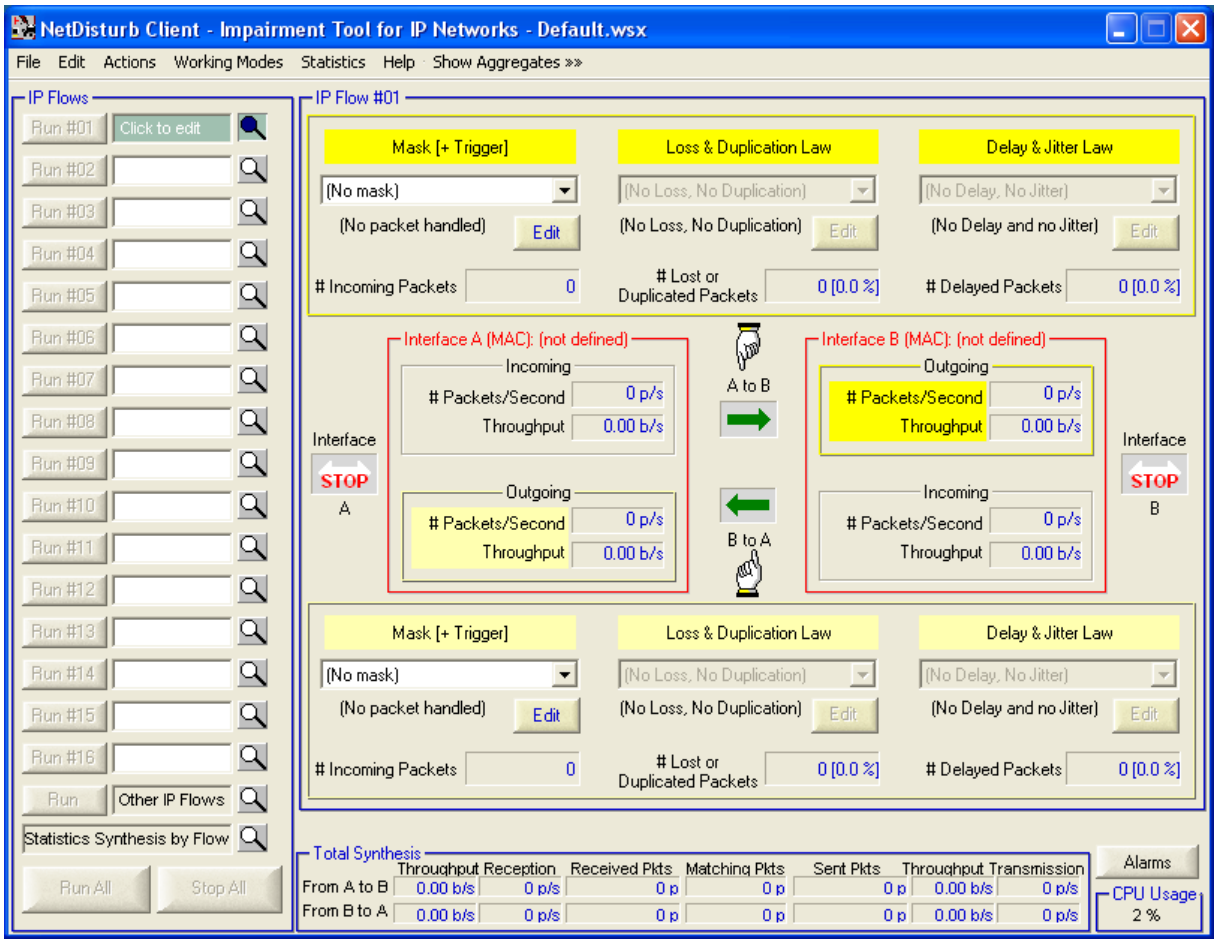
And the **NetDisturb** Client will ask you then to input parameters:



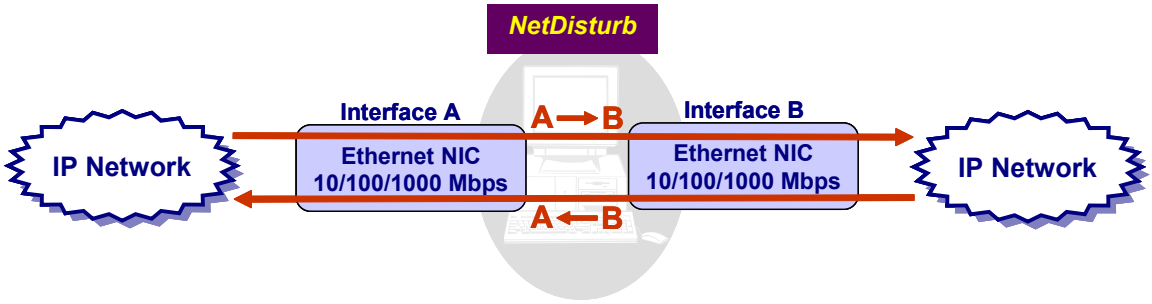
- **User Identification** = Administrator
- **User Password** = (no password needed)
- **NetDisturb Server IP address** = 127.0.0.1

  (127.0.0.1 = default local IP address if the **NetDisturb** Server and the **NetDisturb** Client are installed on the same machine).

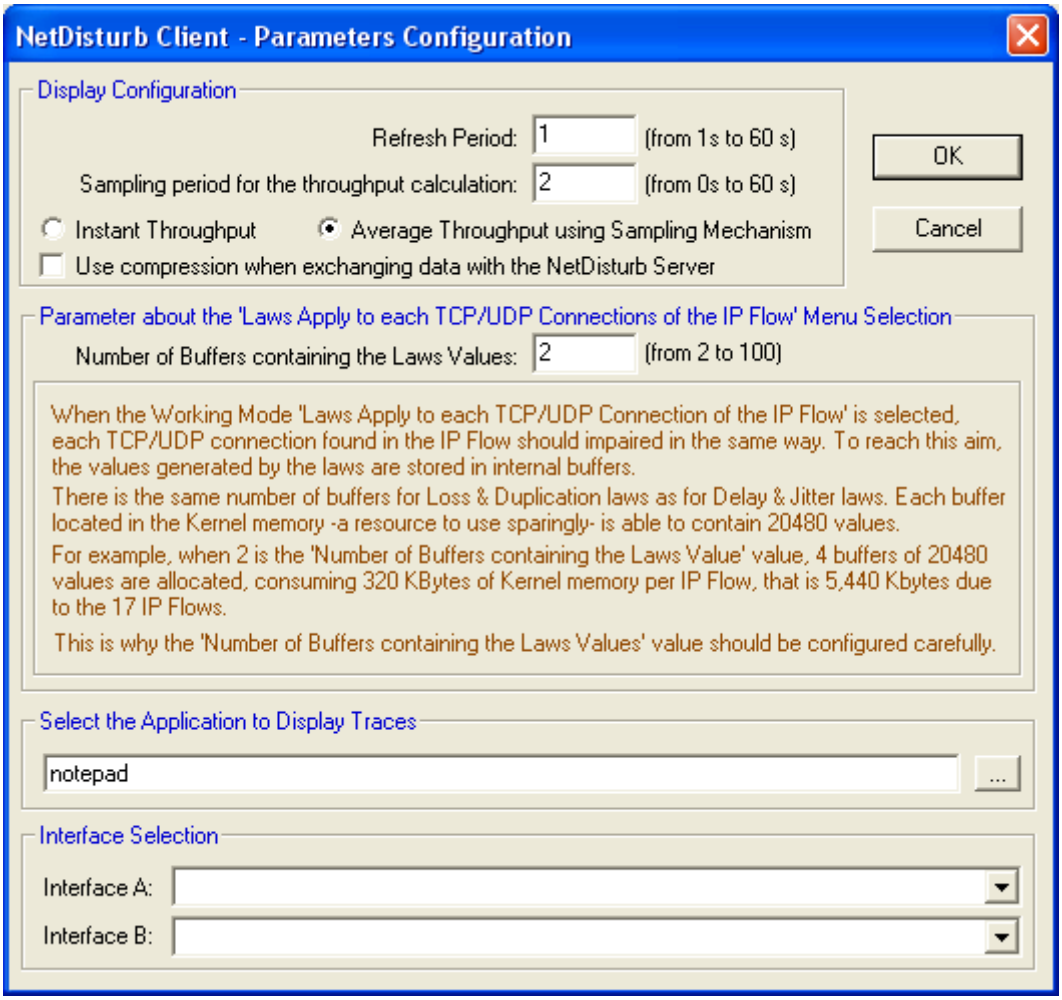
Click “OK” and the **NetDisturb** Client main window will appear:



You need then to select the NICs (interface A and interface B) that the **NetDisturb** Server is going to use.



Select “Configuration” in the Actions menu. The Parameters configuration window is displayed:

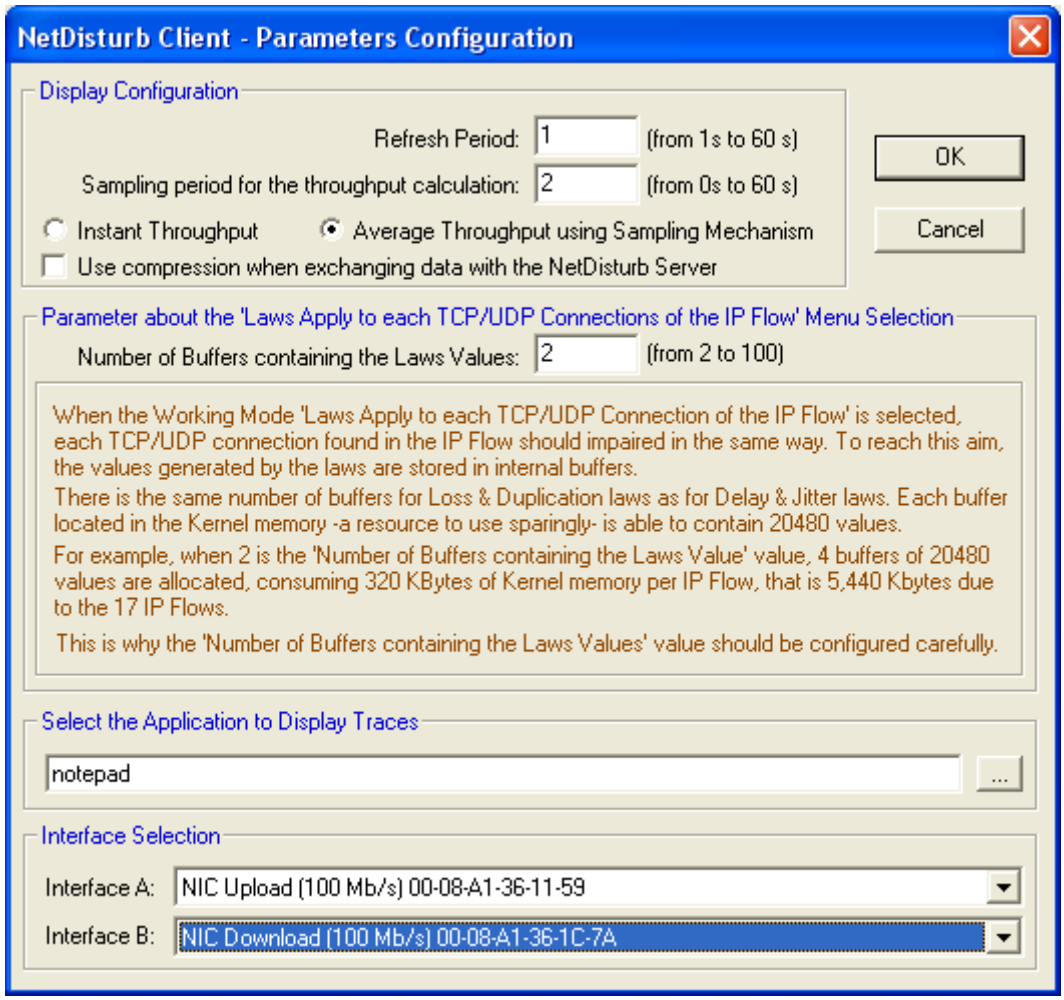


At the bottom of this window in the "Interface Selection" part, select one NIC for Interface A and another NIC for Interface B, and then validate with “OK”.

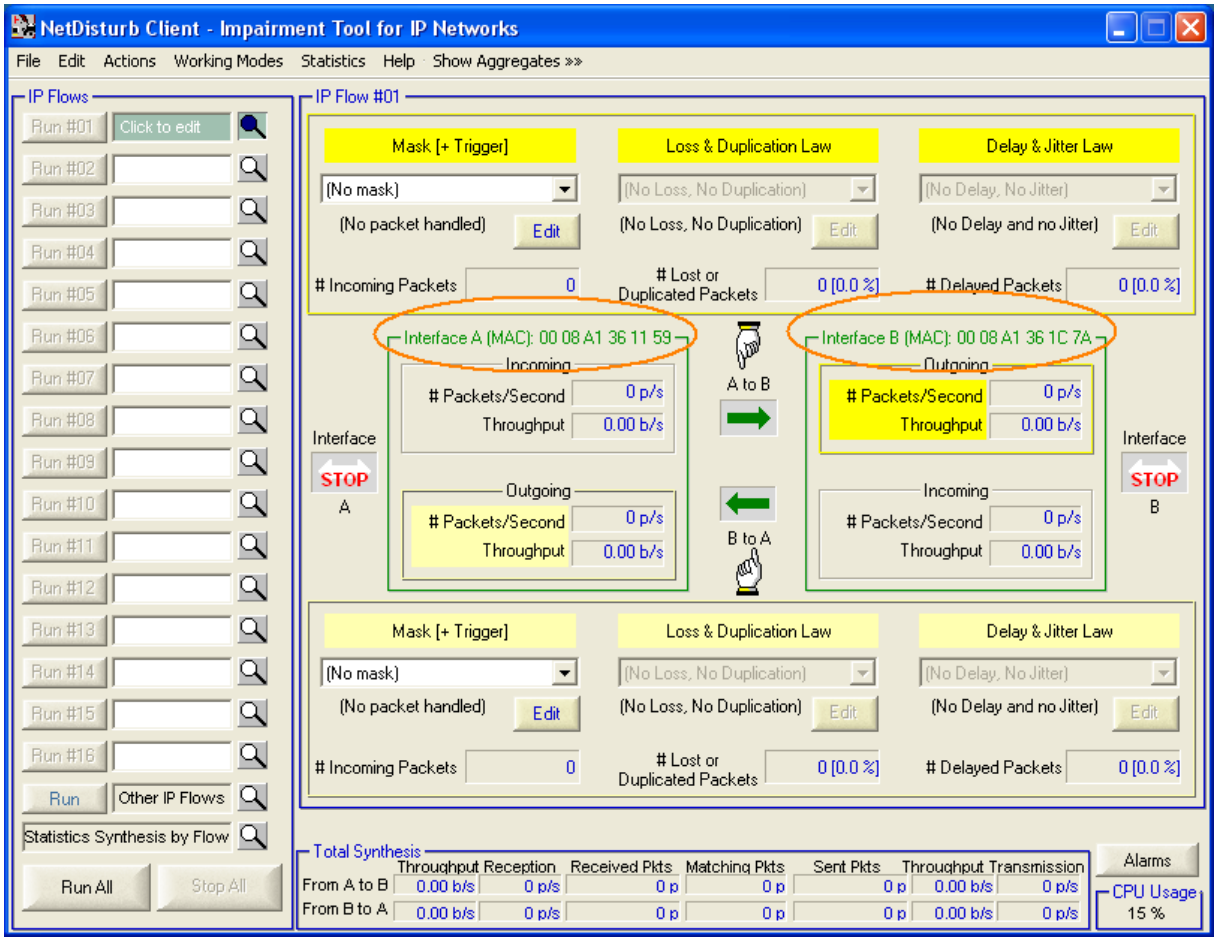
Note: you must see in the combo-box (Interface A or Interface B) all NICs available and operational. If you don’t see any NICs, please do the following steps:

- Verify that your NICs are installed and operational.
- Enable the needed NICs.
- Stop the **NetDisturb** Client.
- Stop the **NetDisturb** Server.
- Reboot your system if necessary.
- Start the **NetDisturb** Server.
- Start the **NetDisturb** Client.

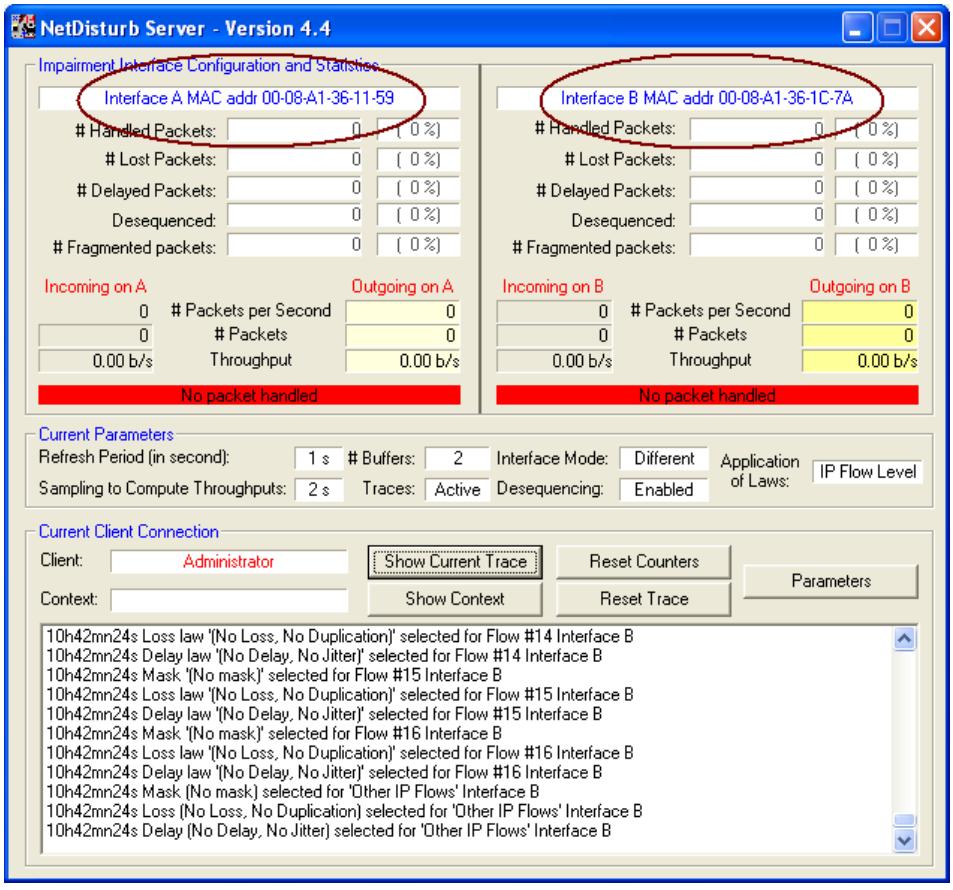
Then you should see your installed NICs in the Interface A and B combo-boxes (see the example below):



As soon as the configuration is done, the **NetDisturb** Server recognizes “Interface A” and “Interface B”.
The MAC Address of the selected interfaces is displayed in the **NetDisturb** Client and **NetDisturb** Server windows:



*Graphical user interface for the **NetDisturb** Client with two Ethernet NICs configured*



*Graphical user interface for the **NetDisturb** Server with two Ethernet NICs configured*

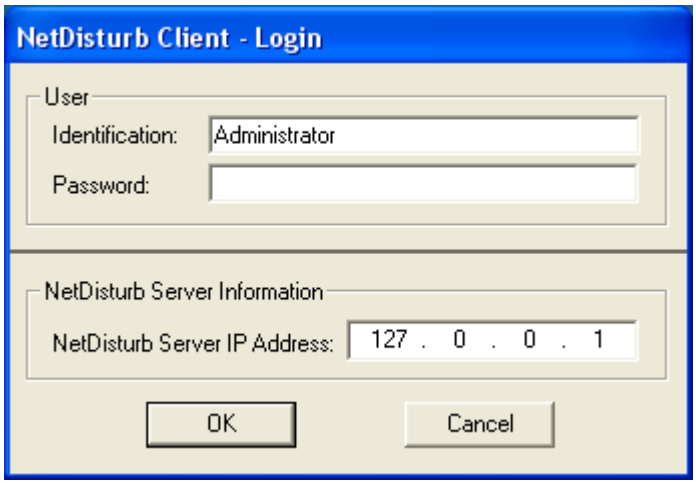
6.2 Detailed Description of the Server and Client Startup

6.2.1 The NetDisturb Server Startup Modes

Offered functionality level depends on the availability or not of the **NetDisturb** driver. If the **NetDisturb** driver is lacking, a message warns the user. In this case it is possible to continue, however only some functions will not be available - this is called the “restricted mode”.

6.2.2 The NetDisturb Client Startup Options

When starting the **NetDisturb Client**, the User identification and Server parameters window is displayed.



This parameter window is composed of two sections:

- **User section**

This section allows the user identification. The identification could be either any user name (User mode) or the ‘Administrator’ (Administrator mode).
A password is required only with the Administrator mode.

⇨ ADMINISTRATOR mode

To be connected as Administrator, **NetDisturb** Client must provide the corresponding password. With this mode the **NetDisturb** Client functionalities are fully available: **NetDisturb** Client can modify laws, stop or activate the relaying process and change the context.

⇨ USER mode

To be connected as User, **NetDisturb** Client provides a different identification than Administrator and a password is not necessary. **NetDisturb** Client functionality is reduced to the use of contexts located on the PC Server.
For this mode, the masks and laws can’t be defined.

- **Server Information section**

The **NetDisturb** Client needs the following information in order to connect with the **NetDisturb** Server:

1. The path to the remote **NetDisturb** Server folder
   This path is composed of two parts:
   - The drive, the virtual drive or the name of the **NetDisturb** Server machine.
   - The directory location of the **NetDisturb** Server where the script subdirectory (containing **NetDisturb**.tst) can be found.

2. The **NetDisturb** Server IP address

In case of connection failure (if one of the parameters is invalid), an error window pops up to specify the connection error. Then the identification window is displayed again.

Error window:

Check:

1) The IP address is correct.
2) The RAS link is correctly established and data are exchanged between client and server.
3) The **NetDisturb** Server is running.
4) Another User is not already connected to the **NetDisturb** Server or the **NetDisturb** Client is already running on the Server machine.

6.2.3 Windows XP Service Pack 2

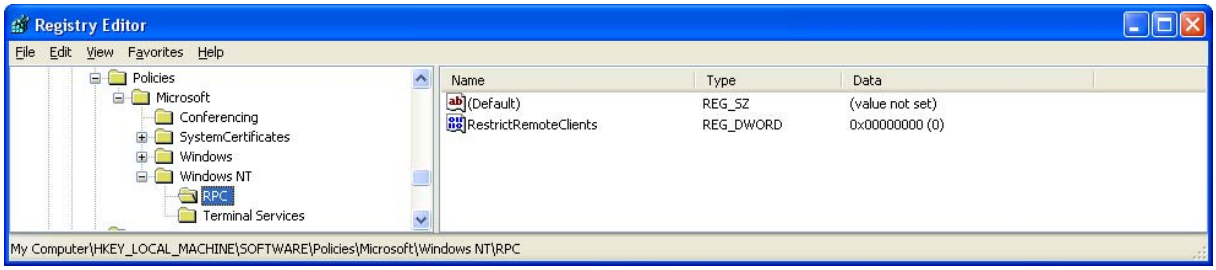
The installation procedure of NetDisturb version 4.4 creates the Registry entry if needed and set the value as explained below.

The **NetDisturb** Client and the **NetDisturb** Server use the RPC (Remote Procedure Call) mechanism to dialog.
Windows XP Service Pack 2 may deactivate the RPC service. To activate it and allow dialog between the **NetDisturb** Client and the **NetDisturb** Server, you need to modify the Registry.

The registry key is:
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\RPC

The value is:
RestrictRemoteClients REG_DWORD 0x00000000

If the value doesn't exist, you should create it. The result looks like the following figure:



PART 7 Using the NetDisturb Client

The **NetDisturb** Client is the main **NetDisturb** User Interface.
With **NetDisturb** Client you can:

- ⇨ Select packet stream to process and configure impairments to apply,
- ⇨ Run / Stop traffic following the configured impairments,
- ⇨ Open, save… contexts,
- ⇨ Configure the **NetDisturb** Server and **NetDisturb** driver.

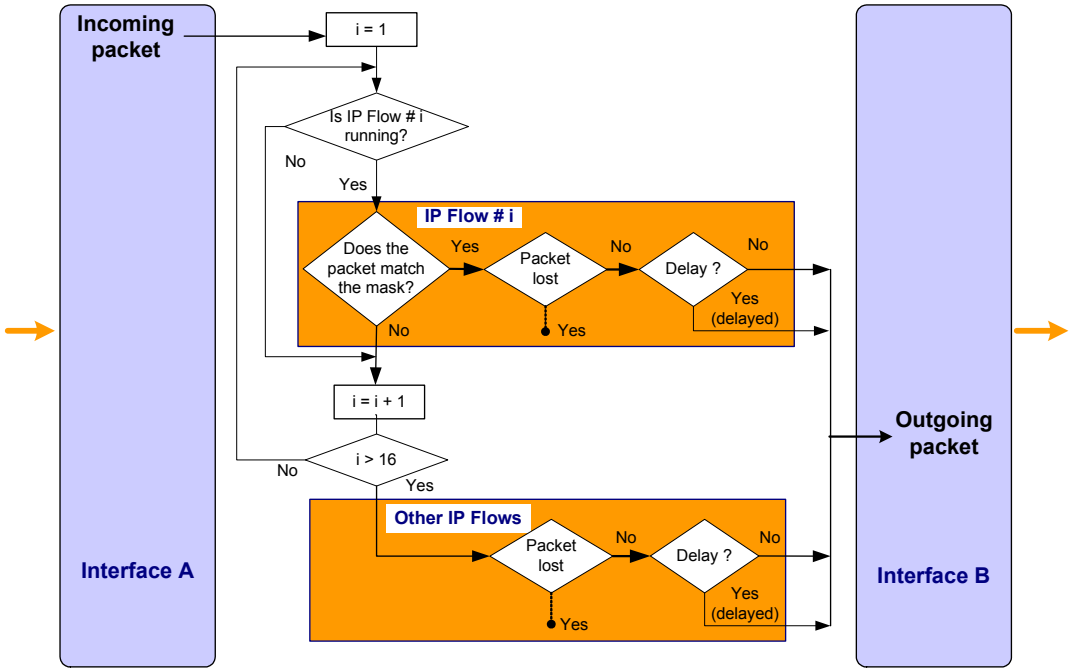
All parameters entered in the **NetDisturb** Client are automatically transmitted to the **NetDisturb** Server.

Important:

To use **NetDisturb**:
⇨ **First run NetDisturb Server**
⇨ **Then run NetDisturb Client**

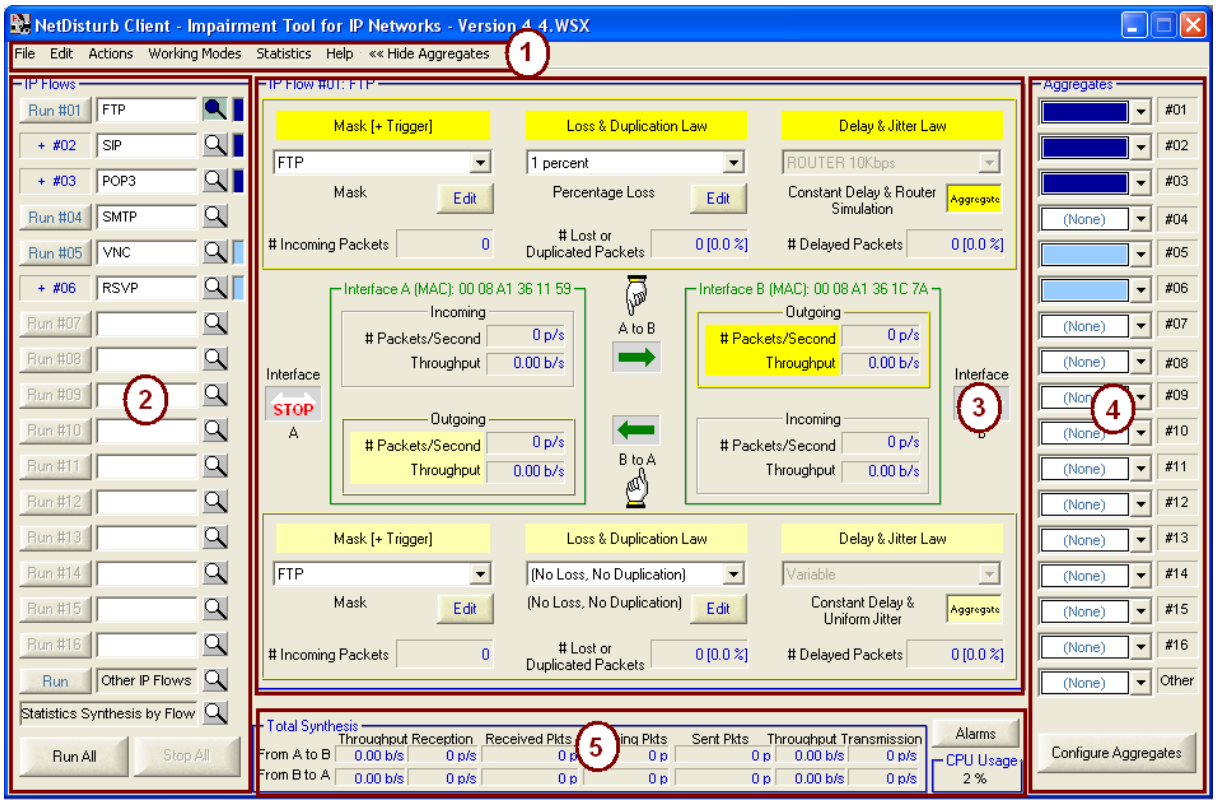
7.1 The NetDisturb Client Main Window

The **NetDisturb** Client main window is displayed after client identification. Traffic and impairment representation on the Client main window is based on the following scheme:



Treatments synoptic for selected packets in a flow from A to B
(B to A direction may be configured from the same manner, but isn't shown on this scheme)

The **NetDisturb** Client main window is composed of four areas:



The menu is a standard application menu. Items of the menu are detailed In reply to: paragraph 7.2 Menu Description.

The ‘IP Flows’ area lists mnemonic-names of flows. This area is used to start and to stop IP Flow unitary or all flows at the time. The loop button is used to selected flow individually. The last two flows have predefined behavior: The Other IP Flows allows applying specific loss and delay laws to non-previously filtered IP packets. The ‘Statistics Synthesis by Flow’ loop summarizes flows #1 to #16 and the ‘Other IP Flows’ as shown in paragraph 7.3.

This central-part presents traffic statistics on each IP Flow #1 to #16 or the ‘Other IP Flows’. It is used to create, delete and modify loss/duplication and delay/jitter laws, or IP masks.

The 'Aggregates' area allows defining up to 8 aggregates (an aggregate is a consecutive set of IP flows sharing the same Delay & Jitter law). An aggregate is defined with a color and a Delay & Jitter law can be defined for each direction (A → B and/or B→ A).

The total synthesis area is a reference area where global statistics information is presented. It includes ‘Alarms’ returned by the NIC drivers or by the **NetDisturb** driver when memory errors occur. The CPU usage value is provided for information.

7.2 Menu Description

7.2.1 The File Menu

In order to keep the parameters configuration for further tests sessions, the **NetDisturb** Client and Server use context files. Context files are saved with the **.wsx** extension. They are usually saved in the Script folder of the **NetDisturb** Server directory.

A context file contains:
- The impairment parameters (selected mask & laws),
- The configuration values.

The default context is opened at each run of the **NetDisturb** Client. The most recent files list is kept from sessions to sessions.

7.2.1.1 File/New

This command opens a new default context (no impairment parameters).

7.2.1.2 File/Open

This command allows opening an existing context file (.WSX files). The older version contexts are imported silently.

7.2.1.3 File/Save

This command allows saving the parameters and laws in a context file (.WSX file). The version 4.4 contexts can't be used by an older version of **NetDisturb**.

7.2.1.4 File/Save as…

This command allows saving parameters and laws in a context file, which name is requested in a standard dialog box. The version 4.4 contexts can't be used by an older version of **NetDisturb**.

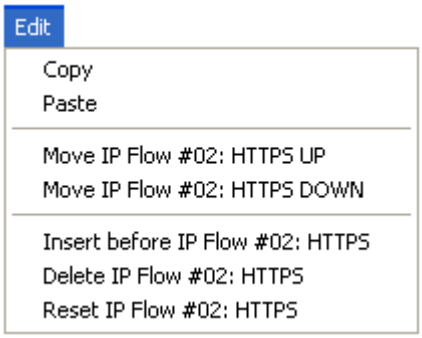
7.2.1.5 File/Recent Files

The 4 most recent files used are displayed at this place.

7.2.1.6 File/Exit

This command stops the **NetDisturb** Client. If changes where made you get the opportunity to save them into a context file.

7.2.2 The Edit Menu



The edit menu helps to handle the IP Flows.

7.2.2.1 Edit/Copy

The Copy item makes a copy of the current IP Flow into memory for further use. Copy includes the current selected Mask, Loss & Duplication Law and Delay & Jitter Law for the both directions. The IP Flow mnemonic is also concerned.

7.2.2.2 Edit/Paste

The Paste item changes the current IP Flow parameters by the previously memorized IP Flow parameters (use of the previous Copy command). It applies to the Mask, the Loss & Duplication Law and the Delay & Jitter Law for the both directions, and to the IP Flow mnemonic name.

7.2.2.3 Edit/Move xxx Up

The Move Up item moves the selected IP flow to one position up. The Move Up item includes the item's mnemonic on which the operation applies. For example 'Move IP Flow #03 Up' switches IP Flow #03 with IP Flow #02, where the content of IP Flow #03 is moved into the second item, while the content of IP Flow #02 is moved into the third position. The IP Flow mnemonic is also concerned.

7.2.2.4 Edit/Move xxx Down

The Move Down item moves the IP flow location to one position down. The Move Down item includes the item's mnemonic on which the operation applies. For example 'Move IP Flow #04 Down' switches IP Flow #04 with IP Flow #05, where the content of IP Flow #04 is moved into the fifth position, while the content of IP Flow #05 is moved into the fourth position. The IP Flow mnemonic is also concerned.

7.2.2.5 Edit/Insert before xxx

The ‘Insert before …’ item makes a room available at current item location, whose mnemonic is added. Items located after the current item move one position down; this includes the current item. The current item becomes empty. The 16th item is lost. If the current item is the 16th, no change appends to the 15th previous but the current – the 16th - is reset.

7.2.2.6 Edit/Delete xxx

The ‘Delete before …’ item deletes the current item and moves the lower items to one position up. The 16th item becomes empty.

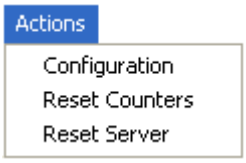
7.2.2.7 Edit/Reset xxx

The ‘Reset before …’ item set the content of the current item with default values. The IP Flow mnemonic is empty.

7.2.2.8 Edit menu and the Aggregates

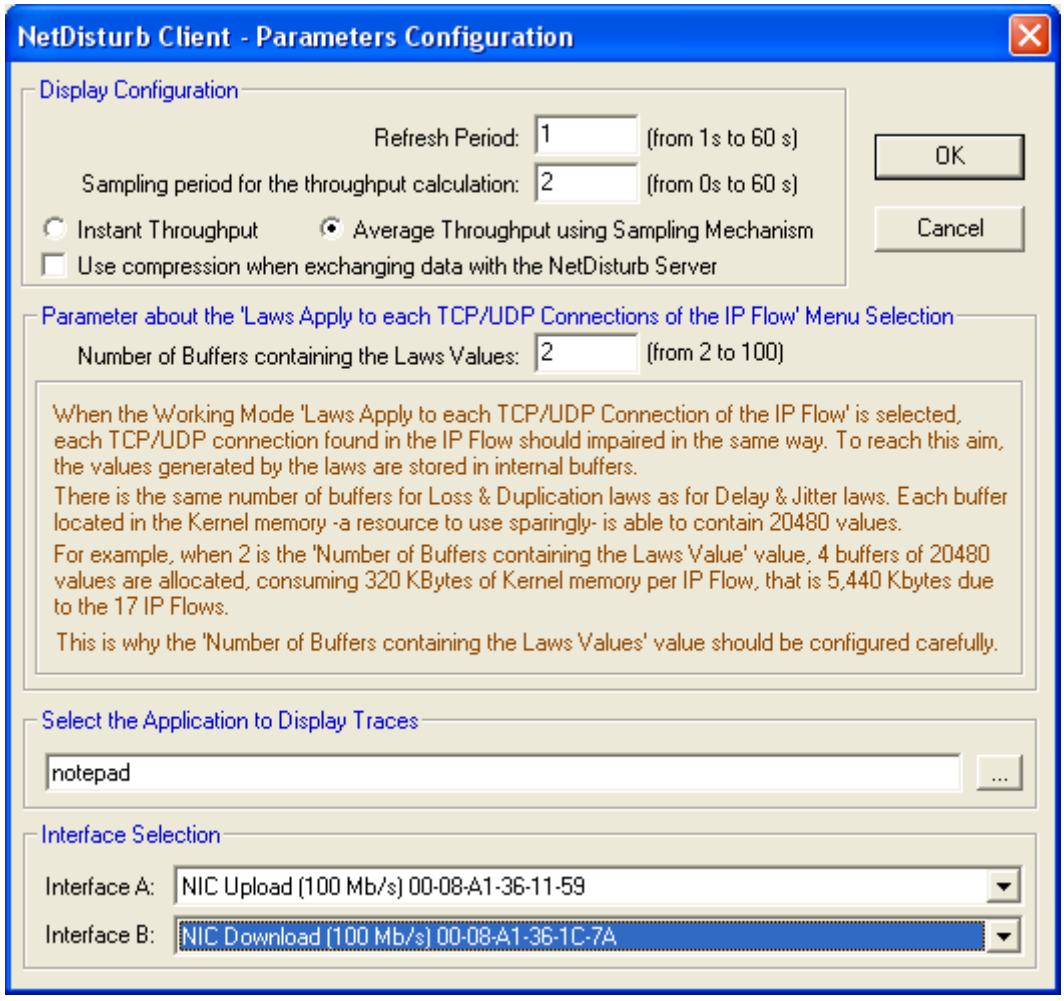
The aggregate configuration of the IP Flow is not changed by an action from the Edit menu.

7.2.3 The Actions Menu



7.2.3.1 Actions/Configuration

Select the "Configuration" item in the Actions menu to display the Parameters Configuration window:



This window is divided in four parts: Display configuration, Parameter about the 'Working Mode / Laws apply to each TCP/UDP connection of the IP Flow' selection, Select the Application to Display Traces and Interface Selection:

⇨ Display configuration

From this section you can:

- Define the refresh period for the display of GUI's counters
- Define the sampling period for the throughput calculation

- Define the way throughput will be computed (instant or average by using the sampling mechanism). Average throughput means computing statistics with values of the latest x seconds (x is the sampling period). Instant computing means computing with value of the latest second.
  Remark: Define an average throughput with a sampling period of 0 allows obtaining an average throughput on the whole period of the **NetDisturb** use (since the last Reset).
- Use compression when exchanging data between the Server and the Client.
  Remark: Data compression is useful when the **NetDisturb** Client and **NetDisturb** Server exchange traces and are connected via a ISDN or modem link. When the **NetDisturb** Client and **NetDisturb** Server are exchanging data on the same PC, the use of compression is not relevant.

⇨ Parameter about the 'Working Modes / Laws apply to each TCP/UDP connection of the IP Flow' selection

This parameter (number of buffers containing the law values) is used when the following working mode is selected: "Laws apply to each TCP/UDP connection of the IP Flow" (see paragraph 7.2.4.2), i.e. each TCP/UDP connection found in the IP Flow should be impaired in the same way.
To reach this goal, the values generated by the laws are stored in internal buffers.
There is the same number of buffers for Loss & Duplication laws as for Delay & Jitter laws. Each buffer located in the kernel memory of the NetDisturb Server machine – a resource to use sparingly, is able to contain 20480 values.

For example when 2 is the number of buffers, 4 buffers (2 per direction) of 20480 values are allocated, consuming 320 Kbytes of kernel memory per IP Flow, that is 5,440 Kbytes due to the 17 IP Flows handled by NetDisturb.

This is why the 'Number of Buffers containing the Laws values'
should be configured carefully.

⇨ Select the Application to display Traces

You can define the full path name of the program used to read the traces (word processor program). The Notepad application is entered by default.

⇨ Interface selection

This section allows selecting the Ethernet NIC to use for the interfaces A and B.

7.2.3.2 Actions/Reset Counter

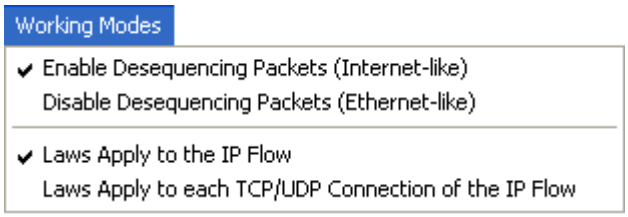
The Reset Counter impacts both the local Client and Server counters. All counters and percentages are set to 0. It doesn't reset Laws counters in use by lost and delay laws i.e. counters used in Associated Uniform Lost law or Fixed Throughput / Fixed Throughput Extended delay laws.

7.2.3.3 Actions/Reset Server

The Reset Server item stops the Server Part. When the Server stops, the **NetDisturb** driver is stopped too. Then the Client is closed and you should restart the **NetDisturb** Server and Client manually.

> To stop and free pending packets, you should reset the server.
> When you stop the IP Flow, pending packets remain in the output queue.

7.2.4 The Working Modes Menu



Impairment may introduce changes in the packet sequence. It is an option to keep the packet sequence or not.

The **NetDisturb** driver analyzes the IP packets to split them into the TCP or UDP connection they belong to. This mode makes possible to apply the same impairment values to each packet of each connection, e.g. to loose the third packet of each connection for example.

7.2.4.1 Working Modes/ Enable & Disable Desequencing Packets

One of the Ethernet characteristics is to keep packets received in order. Internet hasn't' this constraint regarding the packet ordering: some packets can use one way while others another one, with the consequence the receiver may get packets unordered.
The **NetDisturb** Driver can simulate an Internet network or can react as Ethernet does.

How **NetDisturb** creates an unordered case?

> It may append a delay applied to one packet makes this packet to be sent before previous ones, because the delay to apply to the latest packet is smaller than the inter-packet delay and the delay applied to older packets are reduced to be sent before the new packet.

7.2.4.2 Working Modes/Laws apply to the IP Flow or to each TCP/UDP Connection of the IP Flow

- Laws apply to the IP Flow

When the 'Laws Apply to the IP Flow' option is selected, every packets matching the masks requirements are considered belonging to the same flow. Processing is carried out in "continue". When you define to loose 1 packet on 3, the third received packet is lost, whatever the TCP/UDP connection it belongs to.

- Laws apply to each TCP/UDP connection of the IP Flow

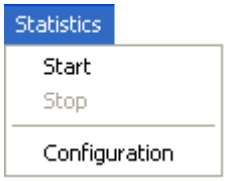
When this option is selected the **NetDisturb** driver analyses each IP packet trying to put the IP packet into a TCP or UDP connection by using the following parameters: protocol, IP addresses and port numbers.
If the connection doesn't exist, a new one is created.
Let's take the same example as above: loose 1 packet on 3.
In that case, the third packet of each TCP or UDP connection will be lost.

Up to 10000 connections can be handled simultaneously.
A flow disappears automatically when the TCP connection is closed and after a configurable timer for the UDP connections.
This timer is configurable in the Registry parameters of the **NetDisturb** driver.

⇨ Buffers

The number of buffers defines the number of values (delay or loss) kept by the **NetDisturb** driver and used for each Connection of an IP Flow.
One buffer contains 20480 values and the minimum number of buffers is 2.
With this working mode, the **NetDisturb** Server generates delay and loss values as much as the **NetDisturb** driver can keep.
When the **NetDisturb** driver detects a new flow, it gets its own pointer to loose and delay values exclusive of the other flows. This pointer starts at the beginning of the set of values. In case of connection with a large number of packets, the pointer increases fast; when connections have few packets their pointer increases slowly. When the pointer reached the latest value, it restarts at the beginning in a circular way.

7.2.5 The Statistics Menu



The **NetDisturb** Client statistics can be saved in a text file. Values saved are shown in the 'Statistics Synthesis' view (see 7.3 for more details). They are saved at the same rate they are visually refreshed.

Columns and IP Flows to save in the statistics file can be selected via the configuration dialog box.

7.2.5.1 Statistics/Start

Start to save statistics into the file. An abstract of each selected connection (Mask name, Lost and Delay law) is saved at the beginning of the file, followed by the list of statistics, one column per statistics.

Each following record gets the format:

Column separated by a tab	Comment
MM/DD/YYYY hh:mm:ss.mmm	Month/Day/Year Hour:Minute:Second.millisecond
#xx	Connection number
Statistic value	One value per selected statistic

When the statistics are saved, the file can be opened for reading but it can't be changed.

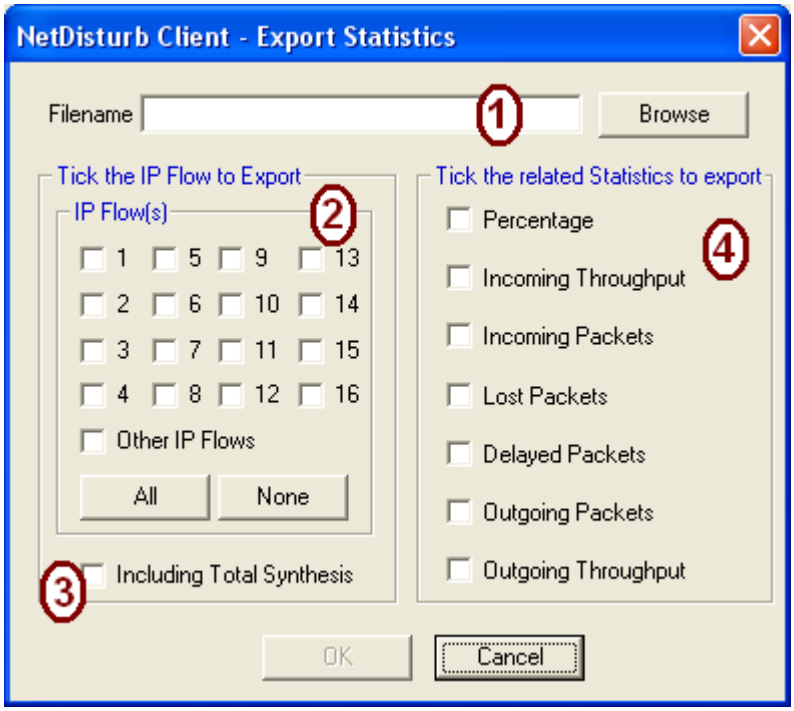
The throughput values are expressed in Kbps.

7.2.5.2 Statistics/Stop

Stop to save statistics in the file. The file can be renamed or copied.

7.2.5.3 Statistics/Configuration

This option allows defining various configuration parameters.



Statistics can start if at least the filename, one flow and one Statistics item are selected.

- Filename ❶

The filename edit box contains the target file name where statistics will be written. If the file still exists, new statistics are appended at the end of the file.

- Tick the IP Flow(s) to Export ❷

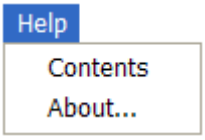
This section is used to select IP Flow to include in the statistics file. IP Flow #01 to IP Flow #16, plus the Other IP Flows can be selected. The Total Synthesis ❸ refers to the bottom part of the Client Windows (part 4 in the detailed description 7.1).

- Tick the related Statistics to export ❹

This section is used to select the statistic items to save:

- Rx (Receive) and Tx (Transmit) Throughput
  These statistics include the volume throughput (b/s, Kb/s, etc) and the packet throughput (packet per second).
- Packets Filtered, Lost or Delayed
  These statistics include the number of packets and the percentage.
- Packet Sent
  This statistic includes the number of packets.

7.2.6 The Help Menu



7.2.6.1 Help/Contents

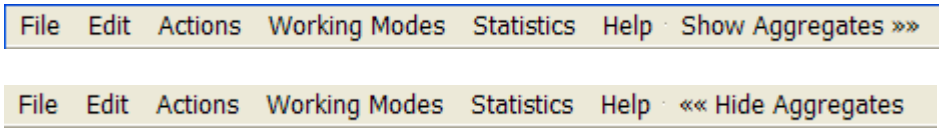
This command opens the **NetDisturb** User Guide as a PDF file. So you need a PDF reader to view the contents.

7.2.6.2 Help/About

This command displays the version number and copyright of the software.

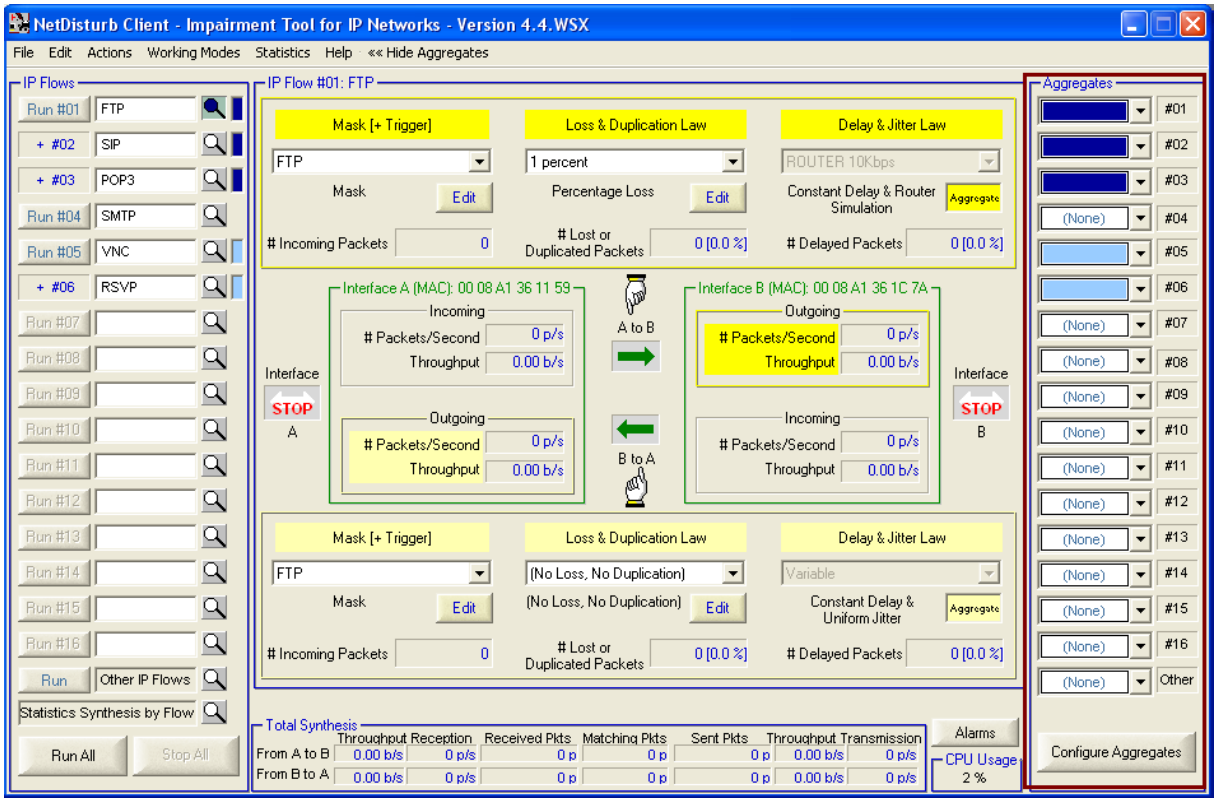
7.2.7 The Hide or Show Aggregates Menu

This menu has two states:

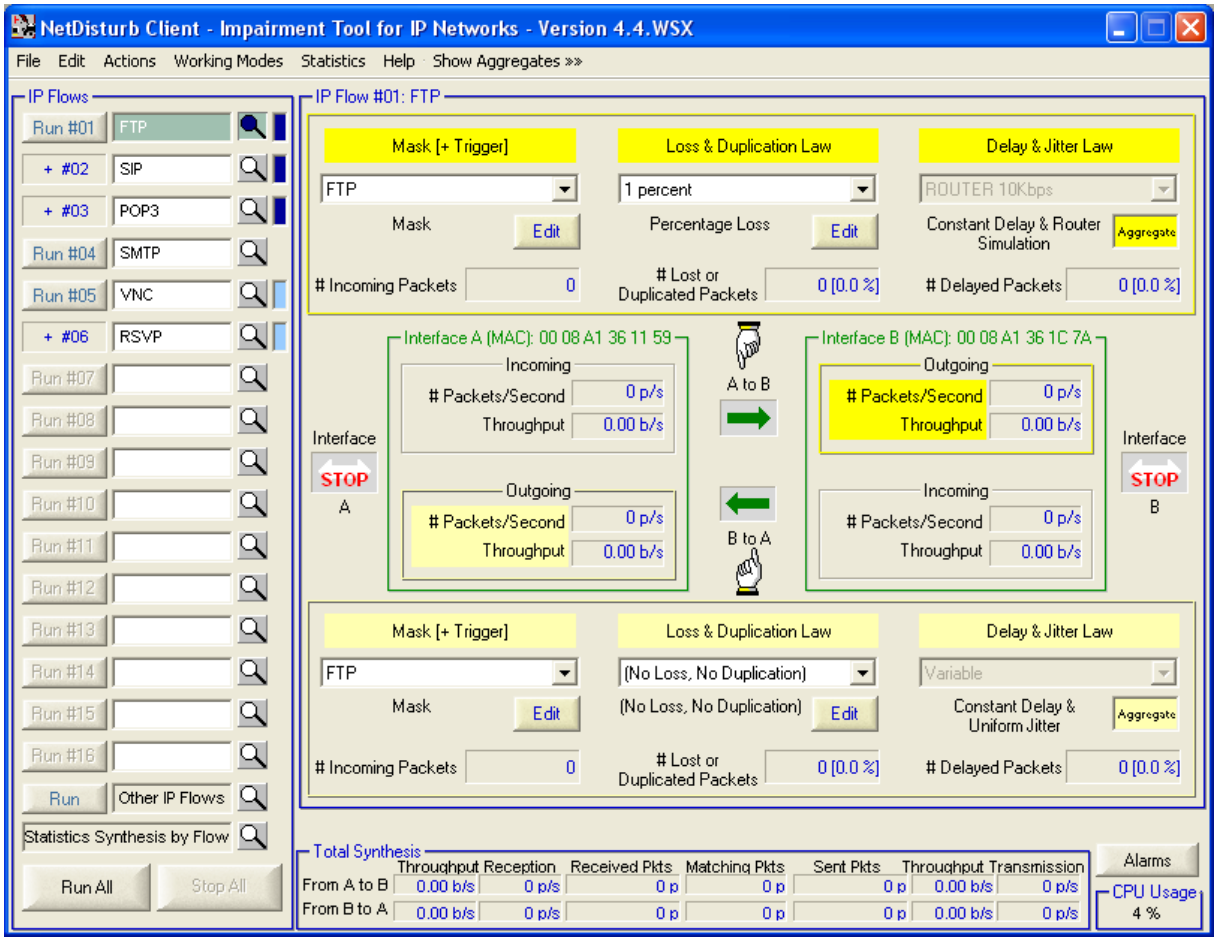


Or

By clicking on the 'Show Aggregates' menu, the **NetDisturb** Client window is enlarged on the right and shows the aggregates:



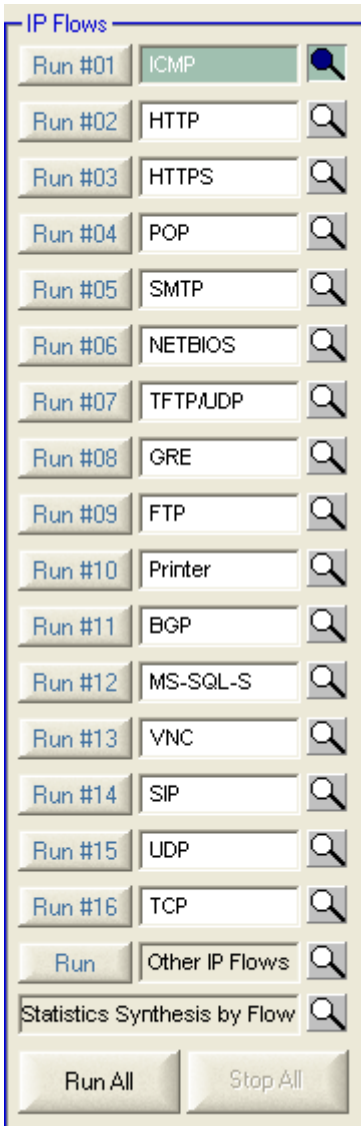
By clicking on the 'Hide Aggregates' menu, the **NetDisturb** Client window is reduced to hide the aggregates:



7.3 The IP Flows

This section describes the IP Flow Client part area.

7.3.1 General Description



Left buttons (Run #xx/Stop #xx) 

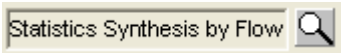
- Each IP Flow can be started or stopped unitary.
- The button 'Run/Stop #xx' indicates the status of the IP Flow will get if the button is pressed. This button is grayed when Interface A and B aren't defined.

Edit area 

- IP Flow #01 to IP Flow #16 can be named with a mnemonic that helps to remember impairment parameters or filter mask used.
- The Other IP Flows can't be renamed: it has specific characteristics described in paragraph 7.3.3.

Loop buttons 

- This button is used to access the details configuration and statistics of a specific IP Flow.
- The color changes to show the current status of the flow.

Statistics Synthesis flow 

- When selecting this view by pressing the loop button, the user can get an abstract of the activity of all flows. Details can be found in paragraph 7.3.4

Bottom buttons 

- The 'Run All' button starts all non-yet-started IP Flows, event IP Flows that don't have a filter defined.
- The 'Stop All' button stops all running IP Flows.

7.3.2 Status of the IP Flows

Idle status

The idle status is the default status of the IP Flow. It is indicated by the button 'Run #XX' and by the loop with a white color as shown:

If IP Flow details are shown, the edit part and loop button is pale green, whatever the status is:

Active Status

The active status is indicated by the button 'Stop #XX' pressed and the loop button orange as shown:

When the current IP Flow is in active state, the active status is indicated by the button 'Stop #XX' remains pressed but the label and the loop are orange, as shown:

7.3.3 The Other IP Flows

The Other IP Flows is in charge to handle IP packets that haven't been filtered by IP Flows #01 to #16. This is why the filter mask isn't available for this IP Flow.

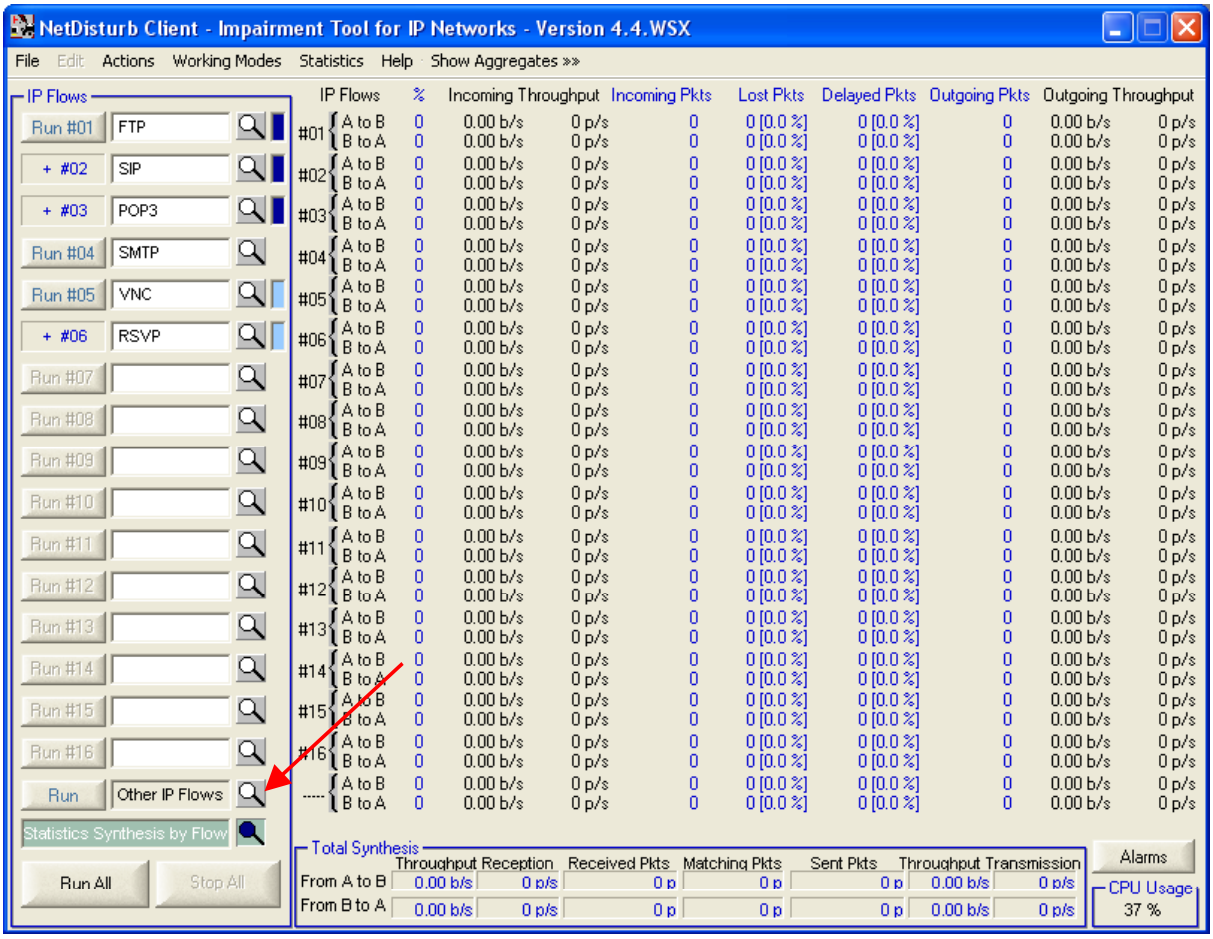
This flow can be used to filter other IP packets not defined by previous IP Flows.

The same operations apply to this '17th' flow as other flows (Run/Stop, Run All / Stop All, etc.)

The colored rules described in paragraph 7.3.2 are relevant to the Other IP Flows.

7.3.4 The Statistics Synthesis View

To get this view you have to press the loop button of the 'Statistics Synthesis by Flow' item.



On this screenshot no IP Flow is running.

Detailed Description:

IP Flows		%	Incoming Throughput		Incoming Pkts	Lost Pkts	Delayed Pkts	Outgoing Pkts	Outgoing Throughput	
#01	A to B	0	0.00 b/s	0 p/s	0	0 [0 %]	0 [0 %]	0	0.00 b/s	0 p/s
	B to A	0	0.00 b/s	0 p/s	0	0 [0 %]	0 [0 %]	0	0.00 b/s	0 p/s

Note: 'Pkts' means 'Packets'

There is one line per direction of the exchange. The upper line refers to the Interface A to Interface B direction. The second line is the opposite direction.

IP Flows

This column presents the flow number and the direction reference

%

This column presents the flow number and the direction reference

Incoming Throughput

This column presents the instant throughput, computed between two refresh periods. Both volume and packet throughputs are shown.

The Incoming Throughput shown in the upper line refers to data received by the 'Interface A' applying the IP Filter mask (or 'Interface B' for the second line respectively).

Incoming Pkts

This column presents the number of packet received. It is a cumulated value.

Lost Pkts

This column presents the number of packet lost, and the percentage of those packets regarding the global number of packets filtered, for the relevant direction.

Delayed Pkts

This column presents the number of packet delay, and the percentage of those packets regarding the global number of packets filtered (***Incoming Pkts*** column), for the relevant direction.

Outgoing Pkts

This column presents the number of packet sent from one interface to the other. It is the number of packets filtered (column ***Incoming Pkts***) minus the number of packets lost (***Lost Pkts*** column), for the relevant direction.

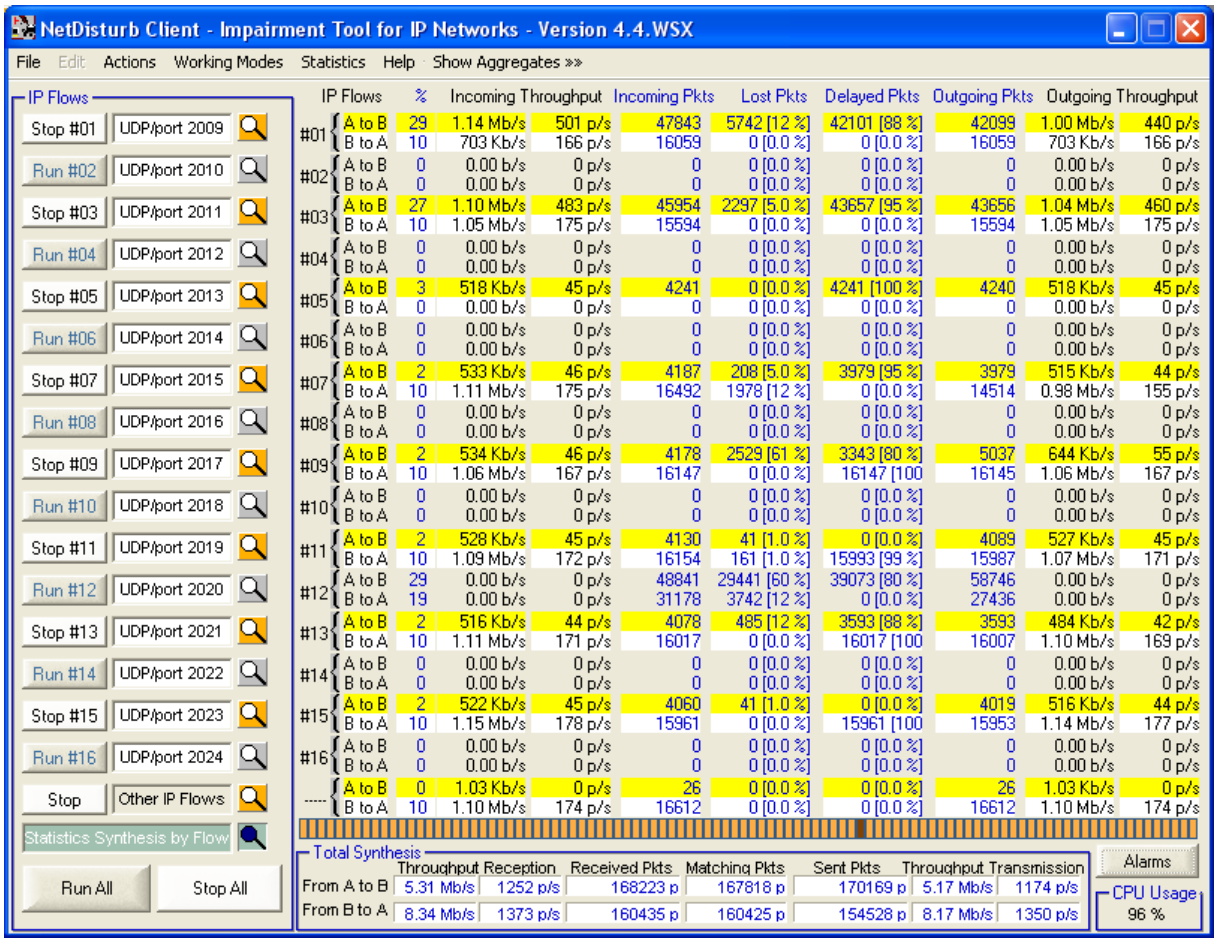
Outgoing Throughput

This column presents the instant throughput, computed between two refresh periods of packet sent to the outgoing Interface. Both volume and packet throughputs are shown.

The Outgoing Throughput column shown in the upper line refers to data sent to the Interface B (or Interface A for the second line respectively).

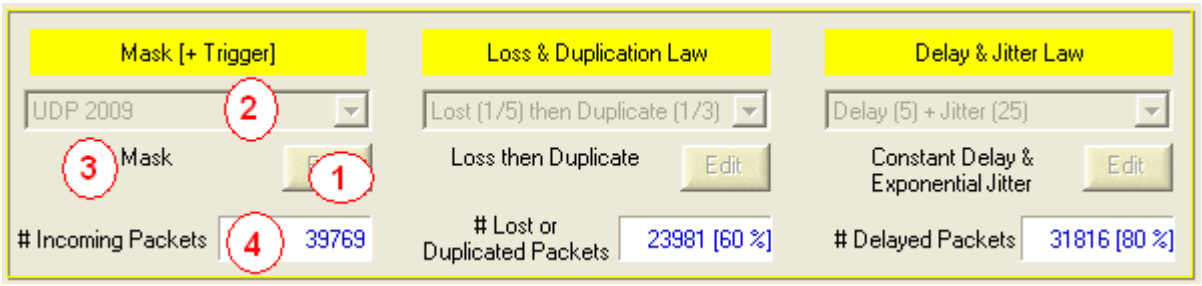
When some IP Flows are active, corresponding lines are colored as shown below:

- the yellow color is related to the A➔B direction
- the white color is related to the B➔A direction



7.4 The Impairment Parameters and associated Commands

The impairment parameters are defined by using a Loss & Duplication law and/or a Delay & Jitter law. These parameters can be modified from the top (for A to B direction) and bottom part (for B to A direction) of the **NetDisturb** Client main window.



Top area of the main Client window

This area is separated in three sections:

⇨ Mask [+ Trigger]
⇨ Loss & Duplication Law (enabled only if a Mask has been previously defined and selected)
⇨ Delay & Jitter Law (enabled only if a Mask has been previously defined and selected)

Each section is composed of 4 objects:

- ❶ the Edit button allows defining and modifying the mask or the impairment law.
- ❷ the combo-box allows selecting the defined mask or law
- ❸ the resume of the mask or law selected in the combo-box
- ❹ a label and a counter showing the number of packets processed

- **Mask [+ Trigger]**

On the left part, the IP Filter mask is presented. This parameter allows selecting packets to process and eventually a trigger to apply. The number of packets that meet the mask is displayed (# Incoming Packets) below the list box. The percentage displayed in parenthesis just besides is the number of filtered packets for that IP Flow on the total packets filtered.

- **Loss & Duplication Law**

This central part presents the loss and/or duplication law applied to the selected packets. It displays the number of lost packets and the ratio of packets lost on the number of filtered packets for the current IP Flow.

> There isn't any counter about the duplication but the number of outgoing packet includes the number of duplicated packets.

- **Delay & Jitter law**

The right part presents the delay law applied to the filtered packets that were not lost. The number of delayed packets and the percentage of delayed packets on number of filtered & no lost packets are displayed.

Once created a new mask or a new law, it will be available for applying to both directions (A ➔ B or B ➔ A).

7.4.1 Selection of a Filter Mask or Loss & Duplication Law or Delay & Jitter Law

To change the selection of the mask or the law, select the requested mask or law from the list displayed in the combo-box. The mask or the law is automatically selected.

7.4.2 The Mask [+ Trigger] Configuration

A mask is a set of parameters to select the packets that would be impaired.
A mask is composed of a combination of several items where each one of them is optional (except the IP version):

- IP version (IPv4 or IPv6 or IPv4 & IPv6): mandatory

- MAC header:
  - MAC Destination address (with the operators 'Equal' and 'Different')
  - MAC Source Address (with the operators 'Equal' and 'Different')
  - VLAN-ID list (802.1Q) (with the operators 'Equal' and 'Different')

- IP header:
  - IP Destination address (with the operators 'Equal' and 'Different')
  - IP Source Address (with the operators 'Equal' and 'Different')
  - Protocol (with the operators 'Equal' and 'Different')
  - Differentiated Services or ToS (with the operators 'Equal' and 'Different')

- Ports (only available with the UDP or TCP protocol):
  - Destination Port List (with the operators 'Equal' and 'Different')
  - Source Port List (with the operators 'Equal' and 'Different')

- Optional Trigger condition (only available if the IP Flow doesn't belong to an aggregate) with 3 parameters:
  - Offset (decimal value)
  - Pattern (hexadecimal string)
  - Result (hexadecimal string)

  The analysis starts at the Offset position of the Ethernet frame, where the content ANDed with the Pattern (up to the pattern length) should be equal to the Result to set the trigger condition.
  When the Trigger condition occurs the following parameters are considered before applying the impairments:

- Trigger parameters:
  - Delay before applying the impairments
  - Impair or not the frame that has triggered
  - Duration of the impairments
  - Number of cycles for the trigger

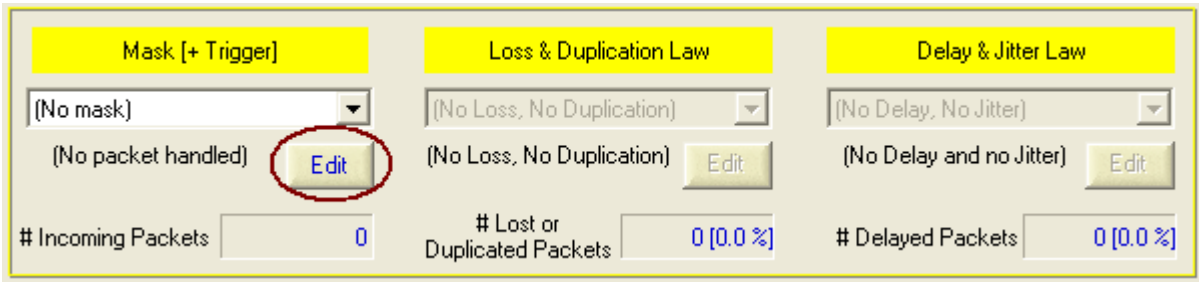
The format for the parameters with a **list** (i.e. VLAN list or Port list) is detailed in the paragraph 7.4.2.5.

By default, the following IPv4 masks are included with the default context called 'Default.wsx':

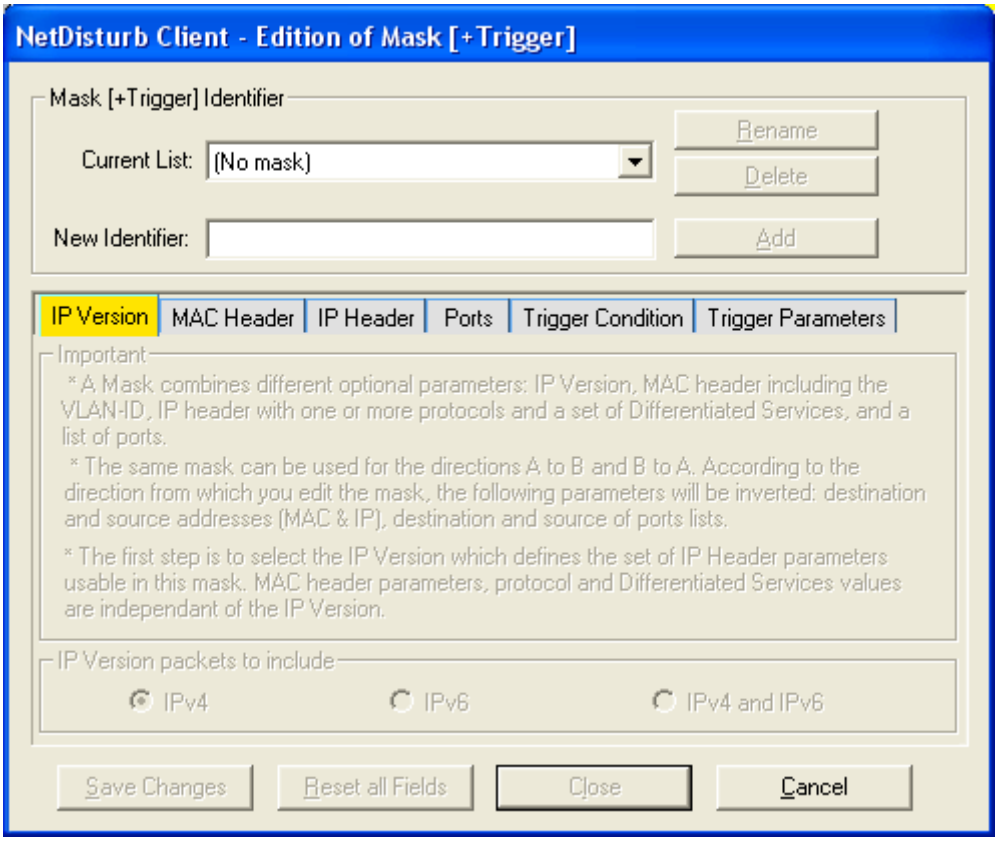
Combo-box	Comment area	Description
(No mask)	*No parameter*	This mask disables the IP Flow because no packet can match a Mask without selection criteria.
TCP	Mask	This filter considers only IP packets with a protocol set to TCP.
UDP	Mask	This filter considers only IP packets with the UDP protocol.
HTTP	Mask	This filter considers IP packets with the TCP protocol and the destination ports 80 or 8080.
FTP	Mask	This filter considers IP packets with the TCP protocol and the destination ports 20 or 21.
SMTP	Mask	This filter considers IP packets with the TCP protocol and the destination port 25.
POP3	Mask	This filter considers IP packets with the TCP protocol and the destination port 110.
VNC	Mask	This filter considers IP packets with the TCP protocol and the destination port 5900.
HTTPS	Mask	This filter considers IP packets with the TCP protocol and the destination port 435.
TFTP	Mask	This filter considers IP packets with the UDP protocol and the destination port 69.
NTP	Mask	This filter considers IP packets with the TCP protocol and the destination port 123.
TELNET	Mask	This filter considers IP packets with the TCP protocol and the destination port 23.
GRE	Mask	This filter considers IP packets with the GRE (x2F) protocol.
RSVP	Mask	This filter considers IP packets with the RSVP (x2E) protocol.
ICMP	Mask	This filter considers only IP packets with ICMP (01) protocol.
NETBIOS	Mask	This filter considers IP packets with the **TCP** protocol and destination ports 137, 138 or 139.
Printer/Port	Mask	This filter considers IP packets with the TCP protocol and destination port 9100.
VLAN	Mask	This filter considers IP packets when the VLAN ID is included between 1 and 5.

Note: supplementary masks may be added depending of the product release.

To define or edit an existing mask, press the Edit button as indicated below:

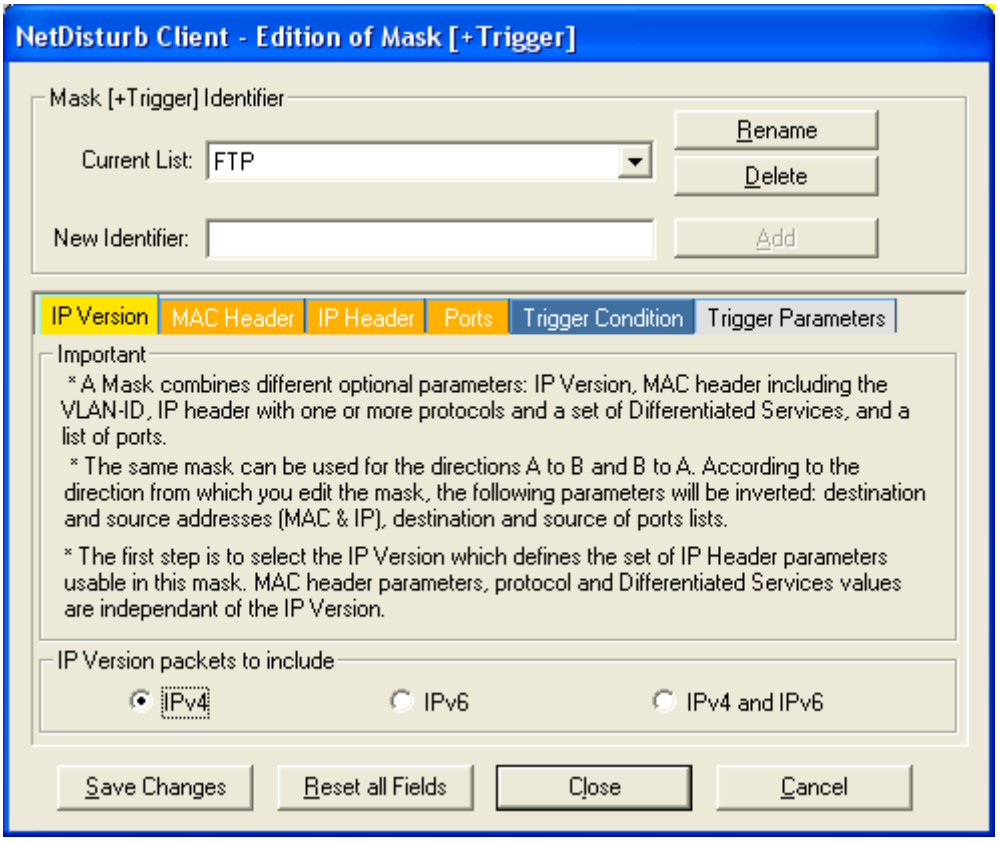


Then the configuration window will appear:



This window allows defining a new mask by first specifying a new mask identifier.
As no mask is yet defined, only the New Identifier field is enabled and the tabs to define the parameters are disabled.

If you select a preexisting mask in the current list of the combo-box, then the parameters of this mask can be viewed and the first "IP version" tab is enabled as in the example below:



This window is composed of two areas:

- Mask [+ trigger] Identifier to select or rename an existing mask, or to define a new mask.
- 6 tabs to define the parameters of the mask and the parameters of the related trigger.

The first 4 tabs concern the mask and the two last tabs are related to the trigger:

- (mask) IP Version
- (mask) MAC Header
- (mask) IP Header
- (mask) Ports
- (trigger) Trigger Condition
- (trigger) Trigger Parameters *(only accessible if a Trigger Condition has been defined)*

7.4.2.1 The Mask [+ Trigger] Identifier

The mask identifier is used to select an existing mask in the “Current List” combo-box. An existing mask can be deleted by pushing the “Delete” button.

You create a new mask by entering the name in the “New Identifier“ field and clicking on the “Add” button”.
Up to 100 masks can be created.

To manage the Mask list, various buttons are available:

Rename: This button should be used to change the Mask identifier.

Delete: This button should be used to remove a Mask from the current list.

Add: This button should be used to insert a new Mask Identifier into the current Mask Identifier list.

7.4.2.2 Six tabs to define the parameters of the Mask and the Trigger

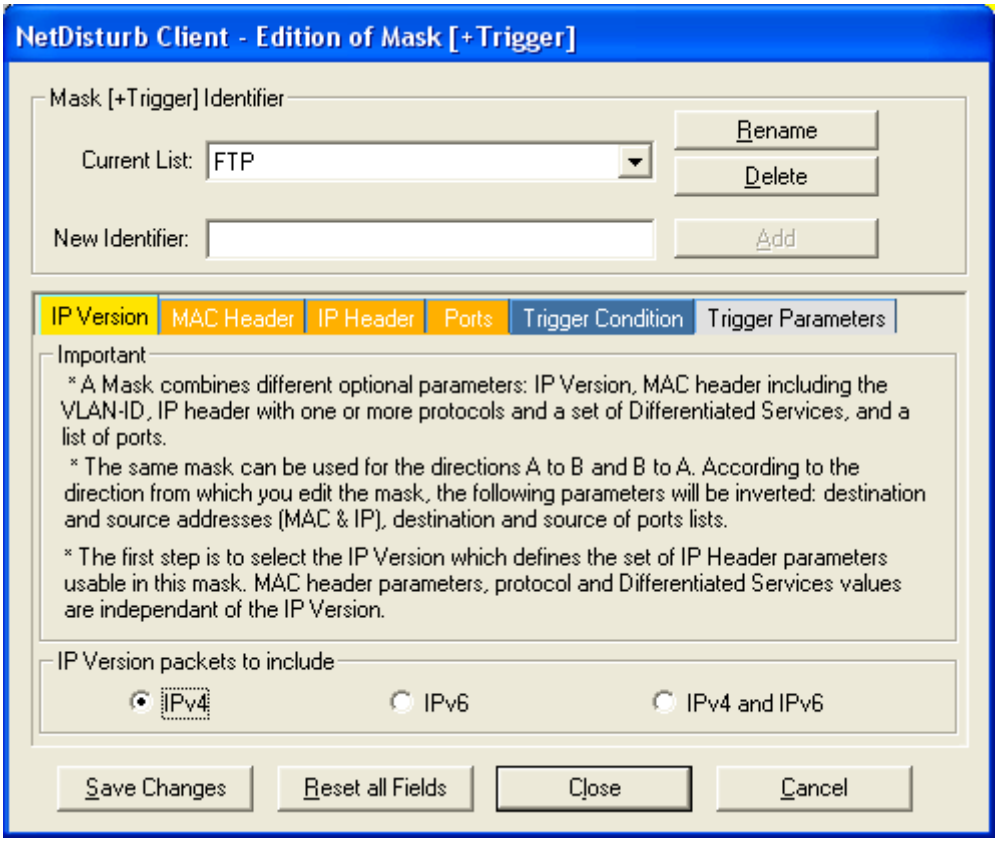
Once a mask has been created, then you can define the parameters of the mask and the related trigger by using the following tabs:

- (mask) IP Version
- (mask) MAC Header
- (mask) IP Header
- (mask) Ports
- (trigger) Trigger Condition
- (trigger) Trigger Parameters *(only accessible if a Trigger Condition has been defined)*

A mask is defined by the combination of four types of parameters: IP version, MAC header, IP header and Ports.
Each parameter of a mask is optional. When a parameter is set then the parameter should be present in the IP Frame to match the mask.

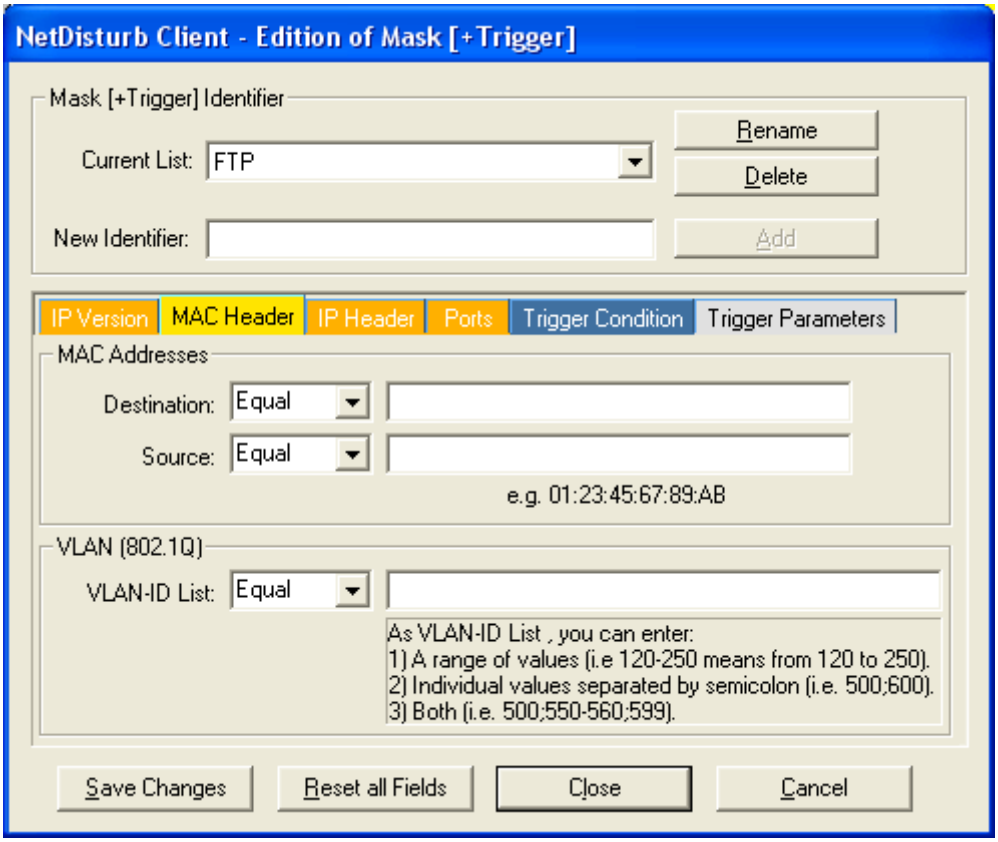
Each mask is defined in reference to a direction in order to identify to which interface the source and destination addresses belongs to. Eventually, if processing is applied to the other direction, the **NetDisturb** driver reverses automatically the source and destination addresses and ports.

7.4.2.2.1 Mask: the "IP Version" tab



The first step is to select the IP version: **IPv4** or **IPv6** or **IPv4 and IPv6**.
The choice of an IP version will affect the parameters handled by **NetDisturb** in the "IP Header" tab.

7.4.2.2.2 Mask: the "MAC Header" tab



MAC Addresses

- Destination with the Equal or Different operator
- Source with the Equal or Different operator

A destination or source MAC address has the following format (12 hexadecimal digits grouped by 2 and separated by the colon character):

XX:XX:XX:XX:XX:XX

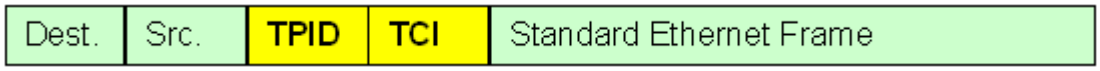
Here is an example:

Destination Equal (to) **00:0B:DB:95:3D:BF**
and
Source Different (of) **00:80:C8:81:37:66**

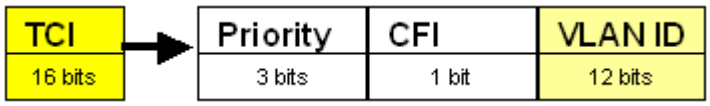
The IP packets having a MAC destination address equal to **00:0B:DB:95:3D:BF** or a MAC source address different of **00:80:C8:81:37:66** will belong to this IP flow.

VLAN-ID list

- VLAN-ID List with the Equal or Different operator (enter a decimal value or a list – see 7.4.2.5 for more details).
  The VLAN-ID can be used only with Ethernet type 8100 frames. In that case, the IEEE 802.1Q format is assumed.



TPID means **T**ag **P**rotocol **Id**entifier. It is equal to 8100.
TCI means **T**ag **C**ontrol **I**nformation. It includes the VLAN-ID as shown:

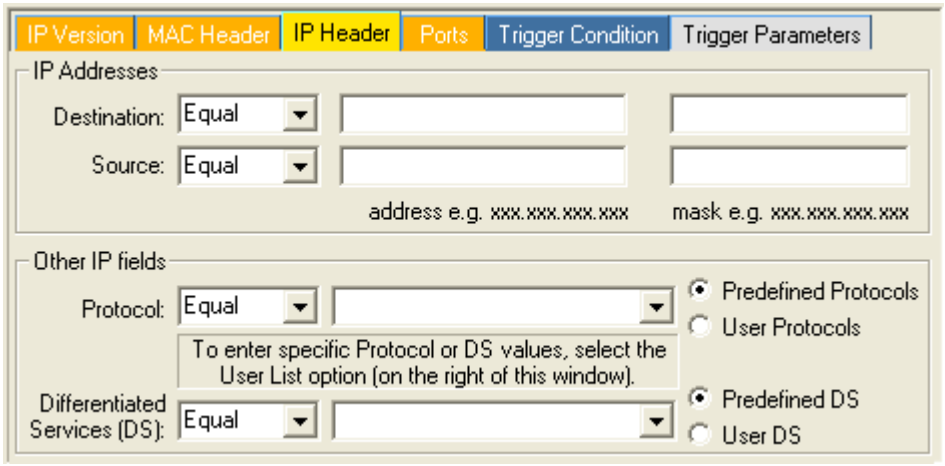


Note: you can input individual values separated by a semicolon or a range or values or a mix of individual values and range of values.

7.4.2.2.3 Mask: the "IP Header" tab

Depending of the IP version (**IPv4**, **IPv6** or **IPv4 and IPv6**) previously selected in the IP version tab, the format of certain fields of this tab will be different.

- If IP version = **IPv4** then the IP Header tab is as follows:



⇨ IP Addresses

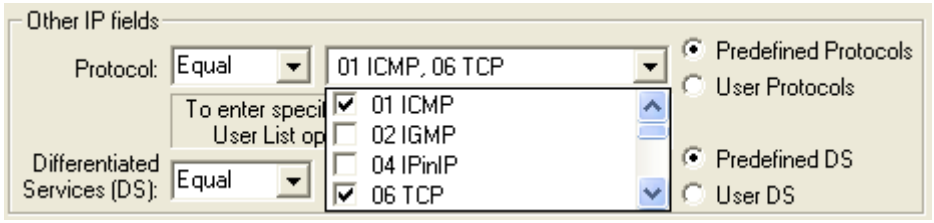
Three objects are defined for the Destination or Source address:
- Operator: Equal or Different
- IP address (enter a decimal value: ex. 192.168.0.17)
- Mask for the IP address (enter a decimal value: ex. 192.168.0.17)

	*The reference for the Source and Destination addresses depend on the original Interface 'Edit selection. In case the 'Edit' button from the Interface A was pressed, the direction from Interface A to Interface B is the reference direction. If the mask is re-edited from the Interface B, then the Source and Destination fields are inverted automatically by the **NetDisturb** Client to match the new direction.*

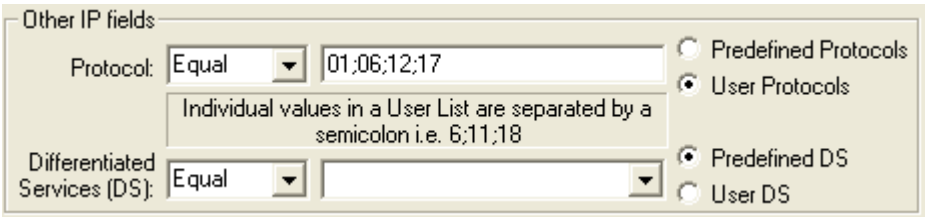
⇨ Other IP fields

Protocol with three objects defined:

- Operator: Equal or Different
- The combo-box allows selecting one or several predefined protocols, or input a list of values separated by a semicolon you want to use
- The option to choose "Predefined Protocols" or "User Protocol List"



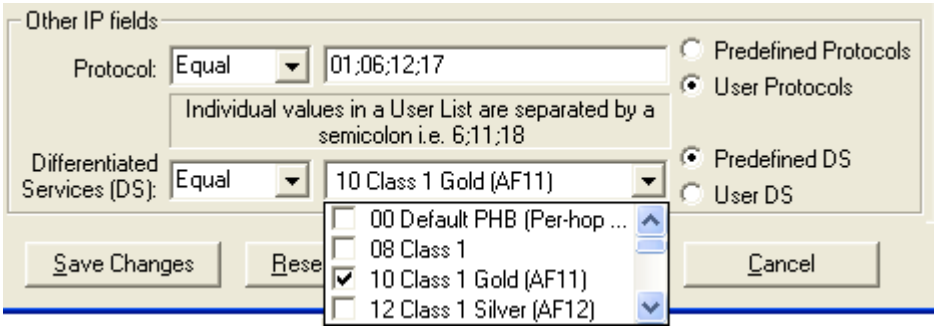
Example by selecting the "Predefined Protocols" option



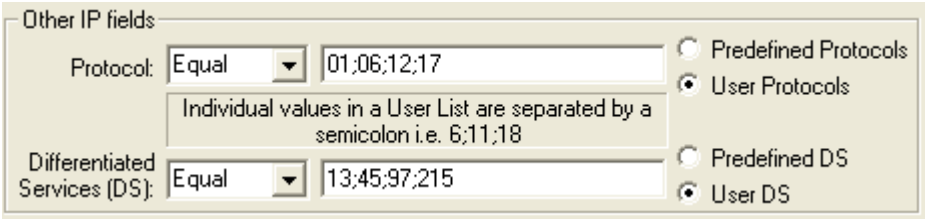
Example by selecting the "User Protocol List" option

Differentiated Services (DS) (TOS byte) with three objects defined:

- Operator: Equal or Different
- The combo-box allows selecting one or several predefined values, or input a list of values separated by a semicolon you want to use
- The option to choose "Predefined DS" or "User DS List"

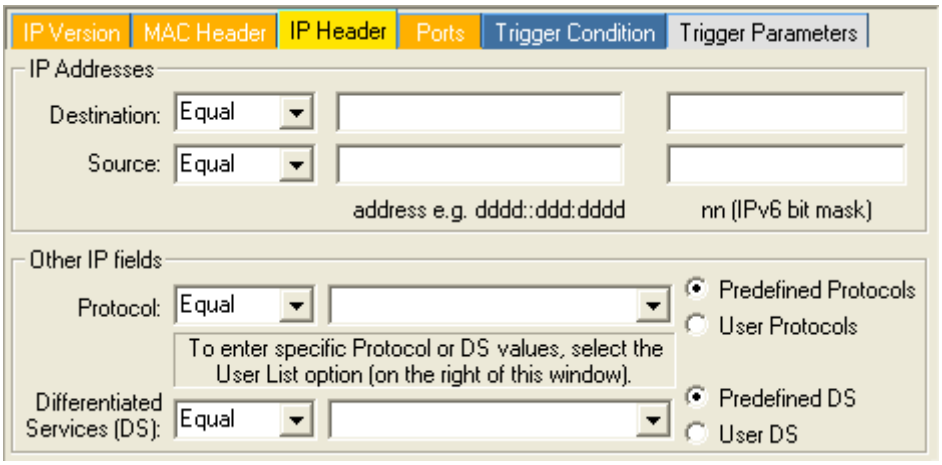


Example by selecting the "Predefined DS" option



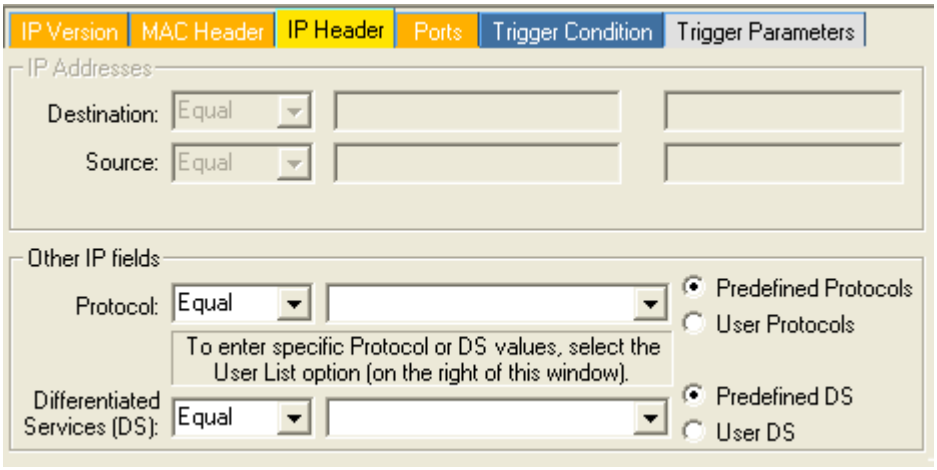
Example by selecting the "User DS List" option

- If IP version = **IPv6** then the IP Header tab is as follows:



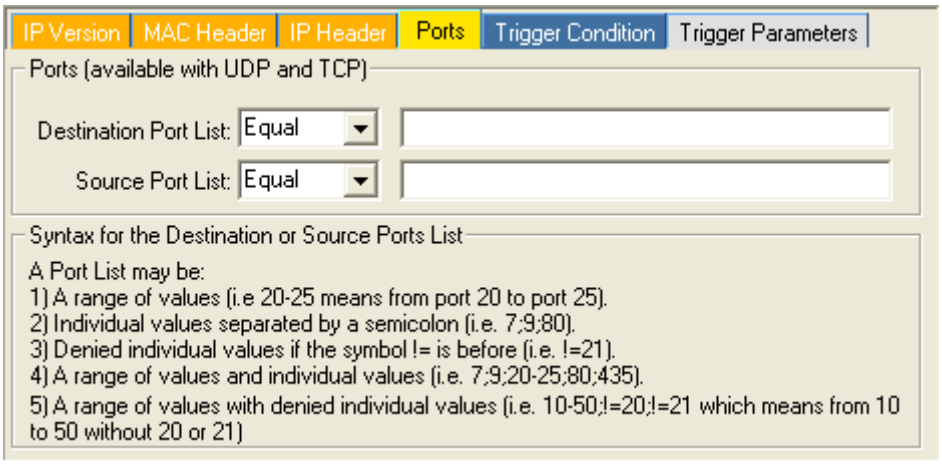
The main difference with the previous case concerns the format of the IPv6 address and the IPv6 mask.
Refer above for details on the other fields.

- If IP version = **IPv4 and IPv6** then the IP Header tab is as follows:



In this case, the IP address fields are disabled because no sense..
Refer above for details on the other fields.

7.4.2.2.4 Mask: the "Ports" tab

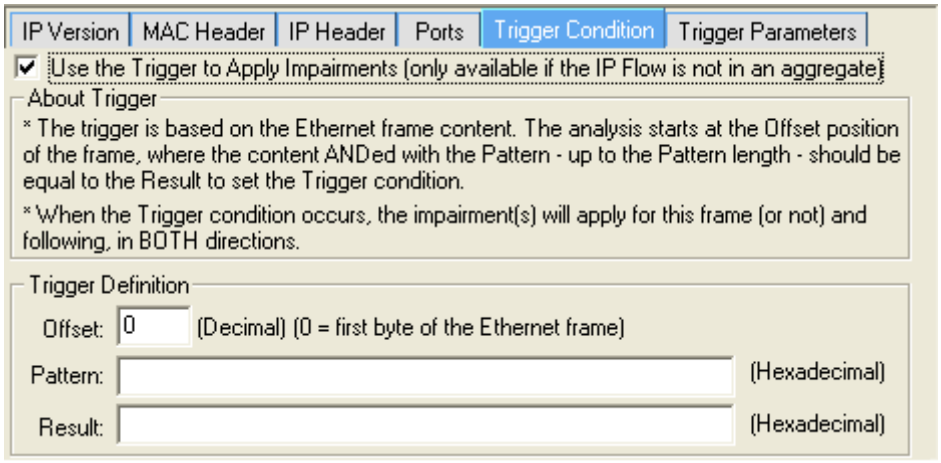


Ports (apply only to the TCP or UDP protocol)

- Destination Port List (enter a decimal value or a list – see 7.4.2.5 for more details)
- Source Port List (enter a decimal value or a list – see 7.4.2.5 for more details)

*The reference for the Source and Destination port depend on the original Interface 'Edit selection. In case the 'Edit' button from the Interface A was pressed, the direction from Interface A to Interface B is the reference direction. If the mask is re-edited from the Interface B, then the Source and Destination fields are inverted automatically by the **NetDisturb** Client to match the new direction.*

7.4.2.2.5 Trigger: the "Trigger Condition" tab



The Trigger is designed to associate the beginning of the impairment(s) to the content of the Ethernet frames and to limit the duration of the impairment.

The Trigger is activated when the content of an Ethernet frame matches a given result. To check if the content of the Ethernet frame matches the expected result, a logical AND operation is made between the content of the Ethernet frame and a given Pattern. The Ethernet frame analysis starts at the given Offset value of the

frame up to the length of the given Pattern. The result of the logical AND operation is compared to the given Result: the Trigger is activated when both are equal.

When the Trigger is activated, the beginning of the impairment(s) refers to both Interfaces (A to B, B to A).

When a Trigger is set in the mask of each direction, the Trigger activated starts impairment(s) on both directions. Additional parameters of the Trigger apply also to both directions.

NetDisturb doesn't check if the Trigger is relevant to the parameters of the Mask.

The following example assumes that the impairment should start when a FTP connection is requested. The definition of the pattern is:

- The protocol should be TCP
- The port number should be 21 (FTP)

To define a Trigger that fulfills this requirement, the Pattern analysis starts at the protocol field of the IP Header that is located at the 23rd bytes of the Ethernet frame. The port number is located at the 37th bytes of the frame. The bytes between the protocol and the port number are not significant.

The definition of this trigger is:

Offset = 23

Pattern = `FF 00 00 00 00 00 00 00 00 00 00 00 00 00 FF`

Result = `06 00 00 00 00 00 00 00 00 00 00 00 00 00 15`

The Pattern parameter and the Result parameter should be entered in hexadecimal.

In this example, a VLAN can't be used with these values because it adds 2 bytes before the IP Header. When a VLAN is used the protocol field is the 25th byte of the Ethernet frame.

With this simple Trigger, let's see what's happening with three common Ethernet frames.

The analysis of the Ethernet frames made by NetDisturb is described below. In this example, the first frame is an ARP Request frame, the second frame is the ARQ Reply and the third frame is the TCP SYN. The part of the frame under analysis is highlighted.

Frame #1 = ARQ Request

Offset	Content
0000	ff ff ff ff ff ff 00 02 55 54 ce 6f 08 06 00 01
0010	08 00 06 04 00 01 00 02 55 54 ce 6f c0 a8 00 78
0020	00 00 00 00 00 00 c0 a8 00 17 00 00 00 00 00 00
0030	00 00 00 00 00 00 00 00 00 00 00 00

The analysis process is the following:

```
Frame part to analyze:      02 55 54 ce 6f c0 a8 00 78 00 00 00 00 00 00
                                    Logical AND with the Pattern
Pattern:                    FF 00 00 00 00 00 00 00 00 00 00 00 00 00 FF

Frame after the AND :       02 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Result expected:            06 00 00 00 00 00 00 00 00 00 00 00 00 00 15
```

The Result is not equal to the frame after the AND operation. The Trigger isn't activated.

Frame #2 = ARQ Reply

Offset	Content
0000	00 02 55 54 ce 6f 00 04 76 9f f8 a7 08 06 00 01
0010	08 00 06 04 00 02 00 04 76 9f f8 a7 c0 a8 00 17
0020	00 02 55 54 ce 6f c0 a8 00 78

The analysis process is the following:

```
Frame part to analyze:      04 76 9f f8 a7 c0 a8 00 17 00 02 55 54 ce 6f
                                    Logical AND with the Pattern
Pattern:                    FF 00 00 00 00 00 00 00 00 00 00 00 00 00 FF

Frame after the AND :       04 00 00 00 00 00 00 00 00 00 00 00 00 00 6F
Result expected:            06 00 00 00 00 00 00 00 00 00 00 00 00 00 15
```

The Result is not equal to the frame after the AND operation. The Trigger isn't activated.

Frame #3 = TCP Syn

Offset	Content
0000	00 02 55 54 ce 6f 00 04 76 9f f8 a7 08 00 45 00
0010	00 2c f7 27 40 00 40 06 c1 c4 c0 a8 00 17 c0 a8
0020	00 78 07 e2 00 15 3f 9b 13 93 00 00 00 00 60 02
0030	ff ff bb 21 00 00 02 04 05 b4

The analysis process is the following:

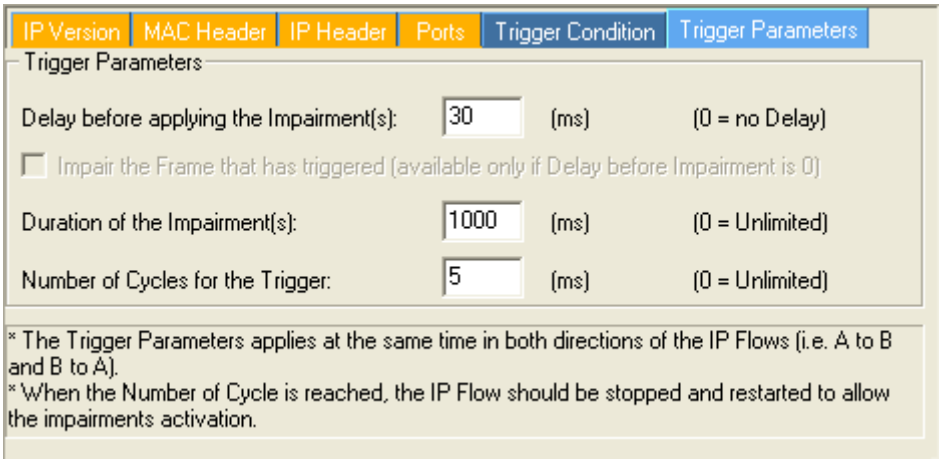
```
Frame part to analyze:      06 c1 c4 c0 a8 00 17 c0 a8 00 78 07 e2 00 15
                                    Logical AND with the Pattern
Pattern:                    FF 00 00 00 00 00 00 00 00 00 00 00 00 00 FF

Frame after the AND :       06 00 00 00 00 00 00 00 00 00 00 00 00 00 15
Result expected:            06 00 00 00 00 00 00 00 00 00 00 00 00 00 15
```

The Result is equal to the frame after the AND operation: the Trigger is activated!

7.4.2.2.6 Trigger: the "Trigger Parameters" tab

This tab is accessible only if the Trigger Definition has been made into the "Trigger Condition" tab.



The Trigger can be configured:

- To add an initial delay before the impairment.
- To include the Ethernet frame that matches the Result in the list of frames to impair or to leave this frame without impairment.
- To limit the impairment in time.
- To limit the number of loops of the impairment based on the Trigger analysis.

These configuration parameters are detailed below.

Delay before applying the Impairment(s)

When a frame has activated the Trigger, an additional delay can be added before **NetDisturb** starts the impairment(s).
The unit of this delay's value is the millisecond. By default, this delay value is 0, which means that **NetDisturb** starts the impairment(s) immediately.
When this delay's value is greater than zero, the impairment(s) will start after the given delay. In the meanwhile, the frames are relayed without being impaired.
When this delay's value is zero, the parameter 'Impair the Frame that has triggered' can be configured.
A delay greater than zero is needed in some application protocols i.e. video, when there is some information to exchange before to start to exchange the most important frames.

Impair the Frame that has triggered

When the' **Impair the Frame that has triggered**' is checked, the frame that has activated the Trigger is included in the set of frames to impair. By default, the frame isn't included.
As example, it is useful to leave the first frame without impairment when this frame starts the connection to impair i.e. a TCP frame with the SYN flag for any TCP connection. In this case, if the TCP SYN frame had been lost, the connection would not have been able to start so there wouldn't be any frame to impair for this TCP connection.

Duration of the Impairment(s)

The **Duration of the Impairment(s)** parameter limits the impairment(s) in time. By default, this duration is zero so the impairment(s) continues until the IP Flow is stopped.

When this duration is greater than zero, the beginning of the Impairment(s) starts when the first frame is impaired. The **Delay before applying the Impairment(s)** isn't included in this duration. When the impairment's duration reach the **Duration of the Impairment(s)** value, the impairment(s) stops except if the **Number of Cycles for the Trigger** parameter is not zero.

When the impairment(s) stops, the next frames matching the mask are transferred from the incoming Interface to the outgoing Interface immediately.

Number of Cycles for the Trigger

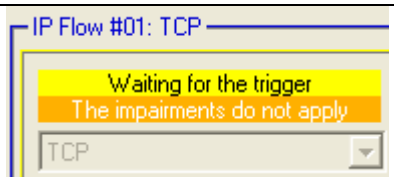
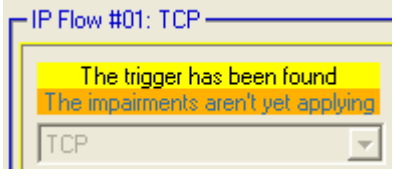
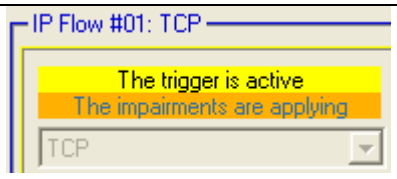
The **Number of Cycles for the Trigger** parameter is available when the **Duration of the Impairment(s)** is not zero.
By default, the number of cycles is 0 i.e. the Trigger cycle is unlimited.

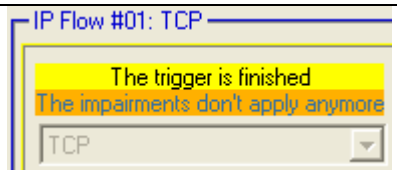
When a **Number of Cycles for the Trigger** is specified, the Trigger cycle restarts at the beginning of the frame analysis process until the number of cycles is reached.

When the **Number of Cycles for the Trigger** is reached, the impairment(s) stops: he next frames matching the mask are transferred from the incoming Interface to the outgoing Interface immediately.

7.4.2.2.7 The Trigger Dynamic

This paragraph gives some examples of the use of Trigger parameters to explain the 4 states of a Trigger:

1. Trigger is waiting. A Trigger gets this state before it has been found. There are 2 cases when a Trigger gets this state: either the IP Flow has just been started or the Duration of Impairment(s) has been reached and the Number of Cycles hasn't been reached. Ethernet frames are relayed without impairment.	
2. Trigger has been found. A Trigger gets this state when the Ethernet frame has been found and a Delay before Impairment(s) has not yet expired. Ethernet frames are relayed without impairment.	
3. Trigger is active. A Trigger gets this state when the Impairment(s) applies, either after the Ethernet frame has been found without Delay before Impairment(s) or when the Delay before Impairment(s) has expired. Ethernet frames are impaired.	

4. Trigger has been stopped. A Trigger gets this state when the Number of Cycles has been reached. This state is a permanent state until the IP Flow is stopped. Ethernet frames are relayed without impairment.	

The Figure 1 illustrates the configuration of a Trigger without Duration. When the Trigger has been reached, the Impairment(s) remains active until the IP Flows is stopped.

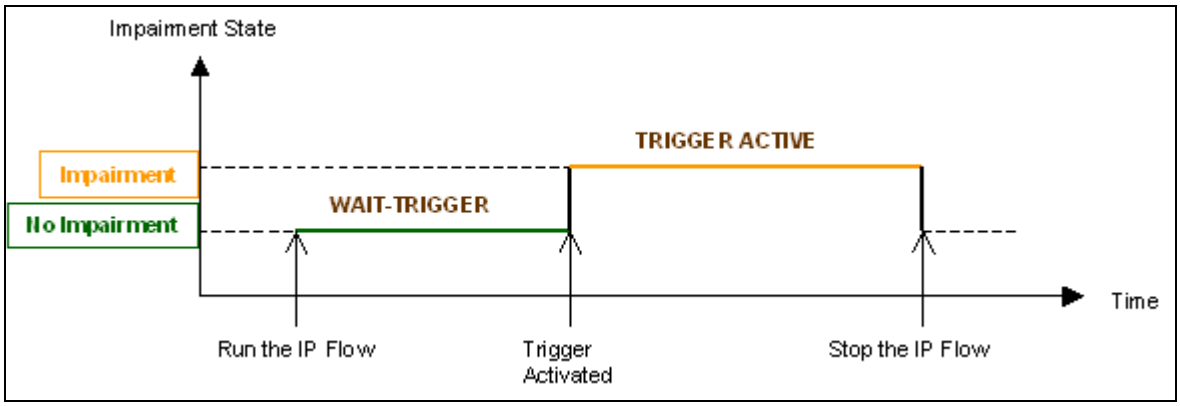


Figure 1 - Trigger without Duration

The Figure 2 illustrates the configuration of a Trigger with an Delay and no Impairment Duration. When the Trigger has been reached, the Impairment(s) starts after the Delay and remains active until the IP Flows is stopped.

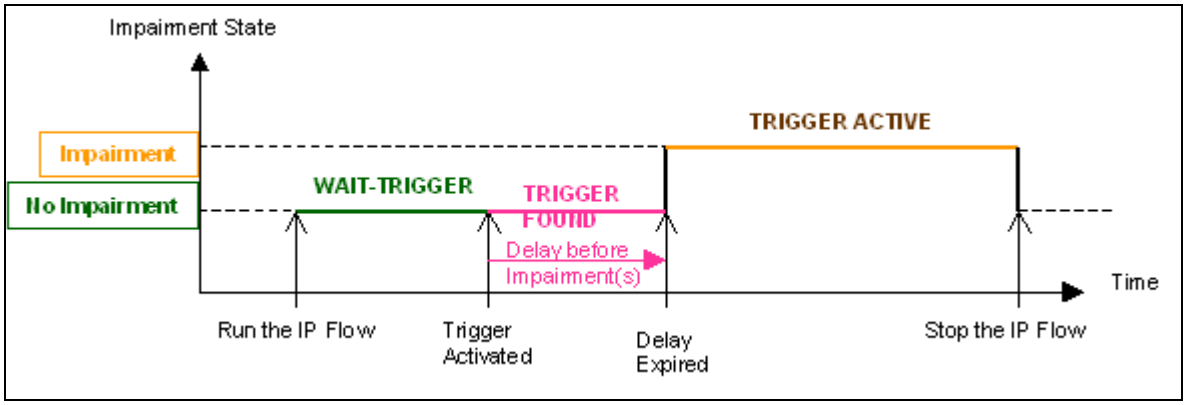


Figure 2 - Trigger with Delay and no Impairment Duration

The Figure 3 illustrates the configuration of a Trigger with the Duration of the Impairment(s) not zero and an unlimited Number of Cycles.

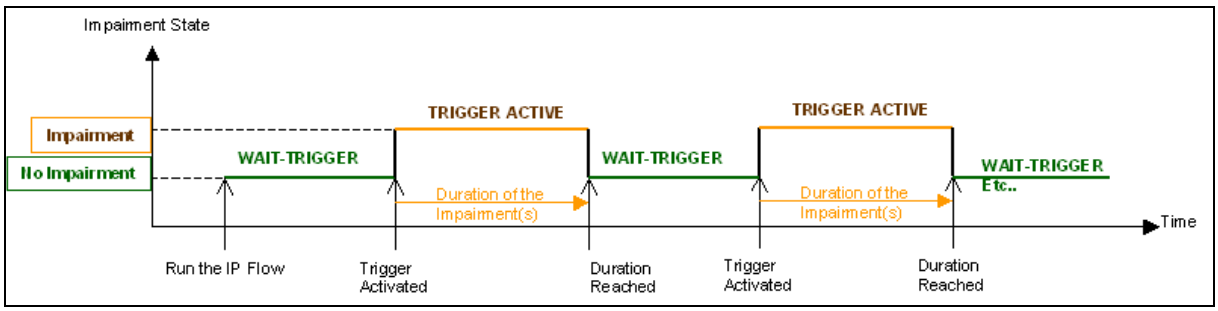


Figure 3 - Trigger with the Impairment's Duration

The Figure 4 - Trigger with the Impairment's Duration and 1 Cycle illustrates the configuration of a Trigger with the Duration of the Impairment(s) not zero and a Number of Cycles limited to 1.

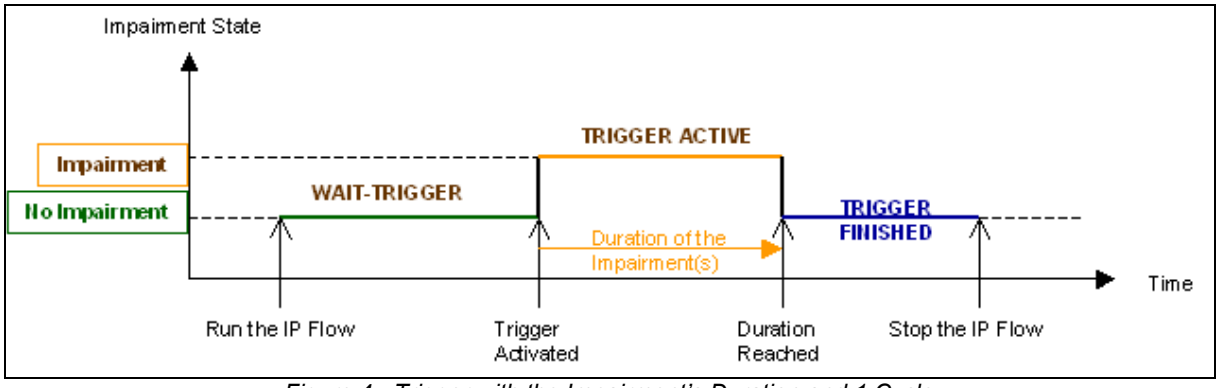


Figure 4 - Trigger with the Impairment's Duration and 1 Cycle

The Figure 5 illustrates the configuration of a Trigger with a Delay Applying before the Impairment not zero, the Duration of the Impairment(s) not zero and a Number of Cycles set to 2.

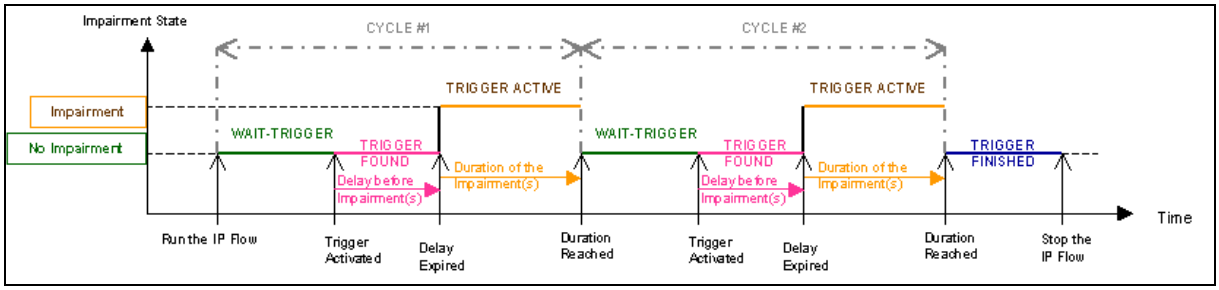


Figure 5 - Trigger with a Delay, the Impairment's Duration and a limited number of Cycles

7.4.2.3 The Action Buttons

To handle the Mask and Trigger parameters different buttons are available at the bottom of the window:

Save Changes: to save the values for the current mask, and insert the new Mask Identifier if the Identifier was not already in.

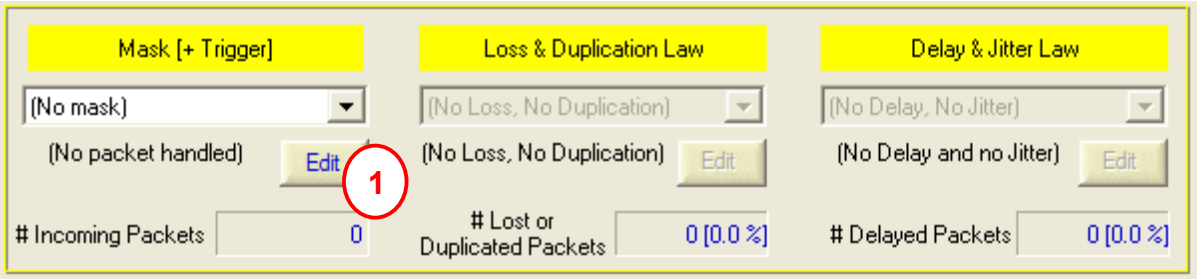
Reset all Fields: to reset all the values previously entered..

Close: to save in the current context all modifications made i.e. new Mask identifiers as well as changes for the existing masks.

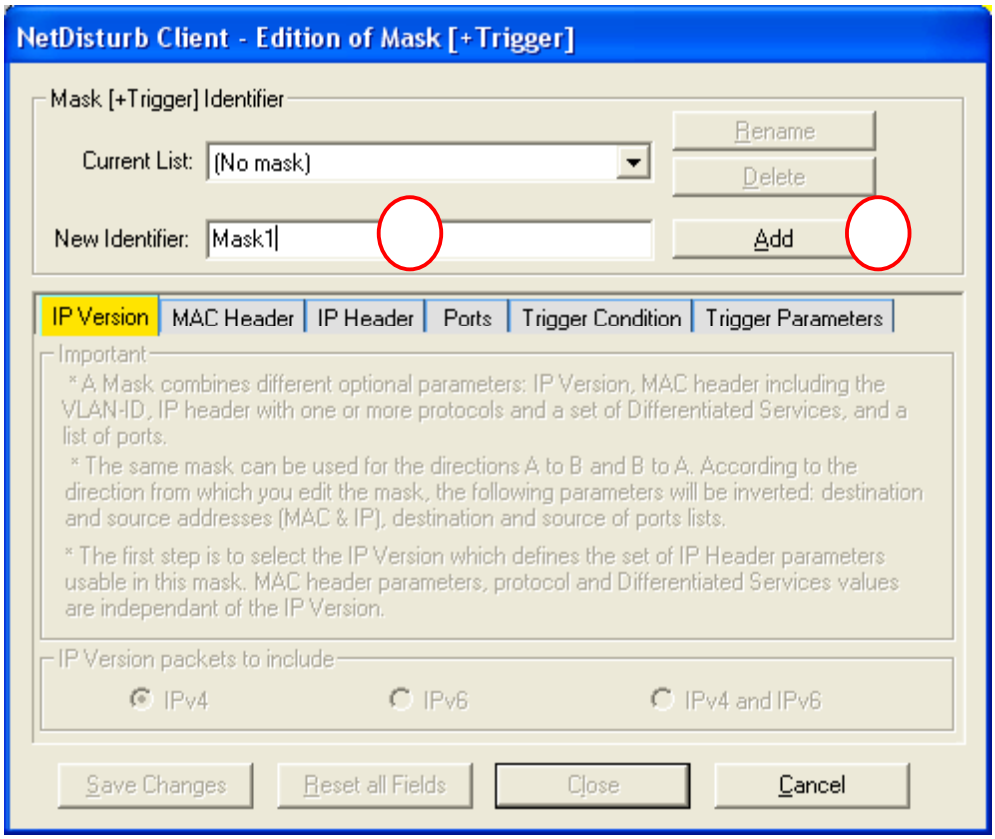
Cancel: to ignore all modifications made i.e. new Mask identifiers as well as changes for the existing masks.

7.4.2.4 To Create a New Mask with its parameters in a few steps

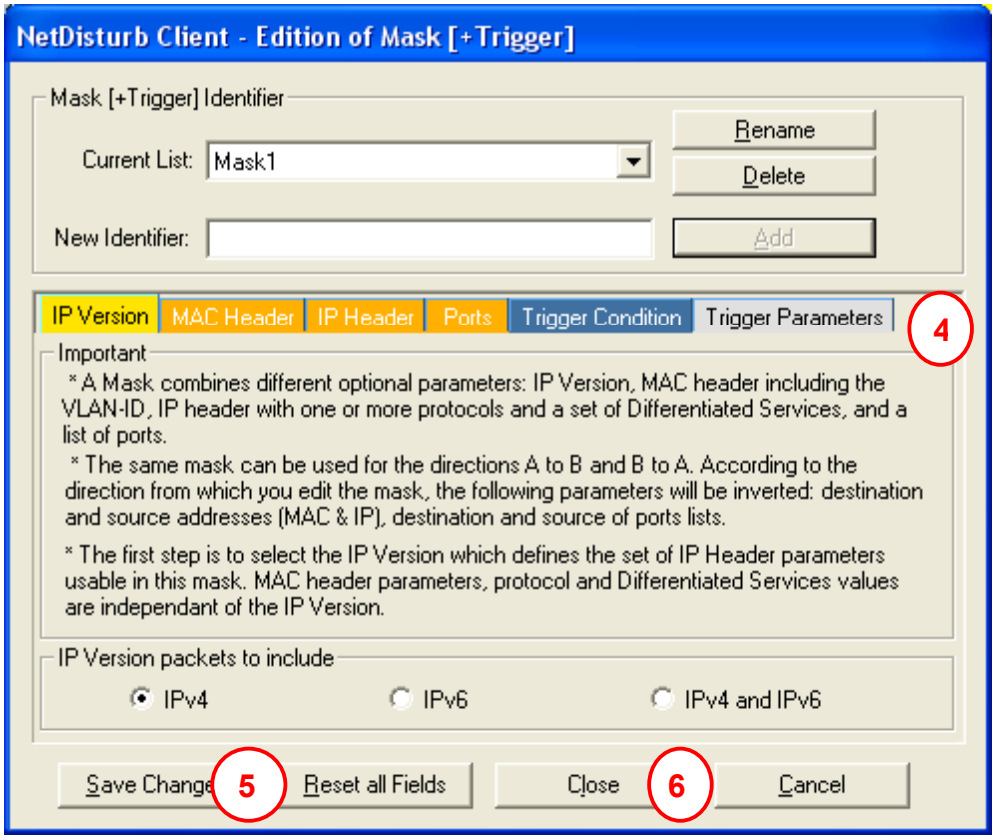
1) Click on the "Edit" button as shown below:



2) The Mask window is displayed and input a name in the New Identifier field (for example Mask1):



3) Click on the "Add" button to add this new identifier in the list of the masks. You have then the following window displayed with the enabled tabs.



4) Now the tabs are enabled and you can define the parameters for this mask (IP version, MAC Header, IP Header and Ports) and if needed the Trigger condition and parameters.

5) Press "Save Changes" to save the parameters and return at step 1 if you want continue to edit or create a new mask.

6) Click "Close" to quit the "Edition of Mask" window.

7.4.2.5 List of Values

Some parameters in the Mask can be a list of values. To match the mask, the IP packet should include one value from the list. The syntax of the list allows a set of individual values or ranges of values. Both individual values and ranges can be mixed. **Values are expressed in decimal**.

The separator between individual values or range of values is the (;) semicolon character. The syntax used is very near the syntax of the printer for a set of pages.

7.4.2.5.1 Individual Value

An individual value is one and only one value.

Example: 135

7.4.2.5.2 List of Individual Values

A list of values is multiple individual values, each value separated by a semi-coma.

Example: 25;80;110;435

7.4.2.5.3 Range of Values

A range of values is a set of values indicated by the first and the last of the range (first and last included). The first value is separated from the last value by a dash.

Example: 2009-2020;3000-3100

7.4.2.5.4 Complex List

Here is an example including individual values and range of values.

List:	**12; 13; 25-30; 50-100;120**
Values matching:	12, 13, 25 to 30 included, 50 to 100 included, and 120
Values not matching:	< 12, 14 to 24, 31 to 49, 101 to 119, > 121

7.4.3 The Loss & Duplicate Law Configuration

NetDisturb is able to loose and/or duplicate packets. Three modes are available:

- **NetDisturb** losses the selected IP packets following the mathematical law configured: a percentage, a 1 on N law or discrete values extracted from a user file.
- **NetDisturb** is able to duplicate IP packets following the Uniform mathematical law configured by User, following a percentage or a 1/N law.
- **NetDisturb** is able to loose packets and duplicate the non-lost packets following a 1/M law.

Up to 100 Loss & Duplication laws can be created.

By default the following laws are defined in the Default.wsx context file:

Combo-box (law identifier)	Comment area	Description
(No Duplication, No Loss)	(No duplication, No Loss)	With this option, no duplication and no loss apply to the IP Flow.

Loss Law		
Constant Loss	Button "To Lose 12 packets"	12 packets are lost each time the user activates this button.
Uniform Loss	Uniform Loss	Domain values [1 to 100] Threshold = 30
Burst Uniform Loss	Burst Uniform Loss	Domain values [10 to 1000] Threshold (n) = 350 Threshold (n+x) = 380 Depth = 2
OnePerTen	User-defined Loss File	Sample file: OnePerTen.txt Loss of 1 packet per 10 packets
Percentage Loss	Percentage Loss	Percentage: 15
One each 10 Loss	Range Loss	Range (N): 10

Duplication Law		
Percentage Duplication	Percentage Duplicate	Percentage = 10 % Minimal Duplication = 1 Maximal Duplication = 3
Duplication 1 Packet every 20	Range Duplication	Range (N): 20 Minimal Duplication = 1 Maximal Duplication = 3
Uniform duplication	Uniform Duplicate	Alpha: 1 – Beta: 50 Threshold: 10 Minimal Duplication = 1 Maximal Duplication = 1
Duplicate if Not Loss	Loss then Duplicate	Loss Range (N): 100 Duplication Range (M): 50 Minimal Duplication = 1 Maximal Duplication = 3

7.4.3.1 Loss & Duplication Law and the Working Mode

Working Mode: Laws apply to the IP Flow

When a Loss & Duplication law is selected on a given IP Flow, the law applies to all packets matching the mask. For each new packet, a new value is extracted from the law or from the file, depending on the type of law selected. These values are stored in a global table by **NetDisturb**. When the table is empty, **NetDisturb** Server provides a new table to the **NetDisturb** driver with new values depending on the law.
This value is compared to the Threshold: if the value is greater or equal than the Threshold, the packet is lost. Otherwise, the packet may then be delayed.

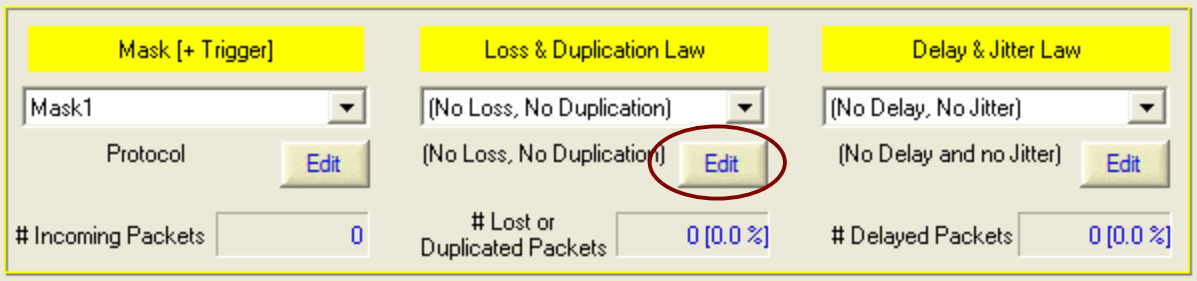
Working Mode: Laws apply to each TCP/UDP connection of the IP Flow

When a Loss & Duplication law is selected for a given IP Flow, the law applies to all packets matching the mask.
These values are stored in a table maintained by **NetDisturb**.
The **NetDisturb** Server provides once a table to the **NetDisturb** driver with values depending on the law. **NetDisturb** loops on values from this table:

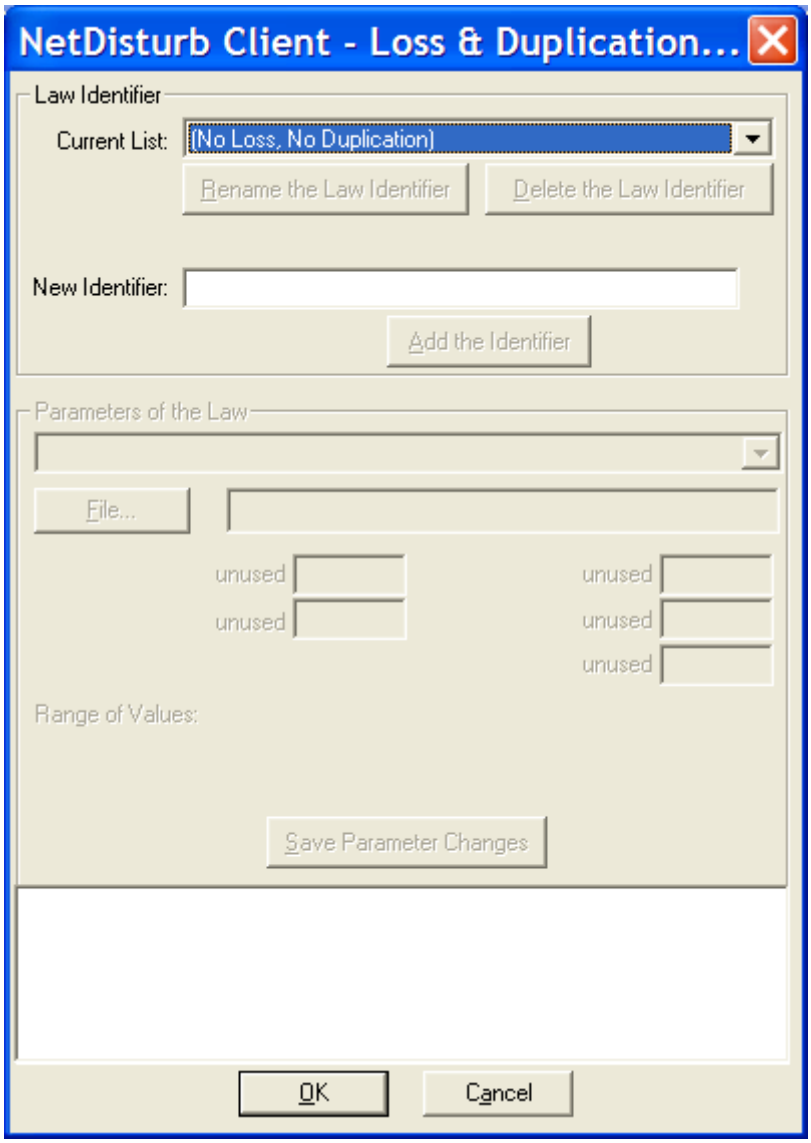
when the end of the table is reached, the **NetDisturb** driver restarts at the beginning.
If the packet is TCP or UDP, the 5-tuple IP addresses, protocol and ports is used to classify the packet. Else the IP addresses and protocol are only used.
For each packet, a loss value is extracted from the loss value buffer, at the current index of the packet of the given connection. When the end of the table is reached, values extracted restart at the beginning.
This value is compared to the Threshold: if the value is greater or equal than the Threshold, the packet is lost. Otherwise, the packet continue to be handled and may be delayed.

7.4.3.2 How to create or edit the Loss & Duplication Law

To create or configure a Loss & Duplication Law click on the “Edit” button at the top or bottom part of the main window.



The following window is then displayed:



The Loss & Duplication Law window is composed of three sections:

⇨ Law identifier

It is used to choose an existing law from the “Current List” combo-box. An existing law can be deleted by clicking on the “Delete the Law Identifier” button.
You can also add a new law by entering a name in the “New Identifier“ field and by clicking on the “Add the Identifier” button”.

Button	*Action*
Rename the Law Identifier	Change the law identifier.
Delete the Law Identifier	Remove the law from the temporary list.
Add the Identifier	Add the Identifier in the temporary list.

⇨ Parameters of the Law

This section is composed of a list box to select the law to apply, and different edit fields may be enabled in order to input parameters.

The "Value range" allows seeing the range of values generated by the law for the user-defined parameters. It applies to the Uniform Loss law and Burst Uniform Loss law.

A list box allows selecting a law among the following pre-defined laws:

- Loss: Constant law
  Parameter: number of packets
- Loss: Uniform law: $f(x) = dx/(\beta - \alpha)$
  Parameters: alpha, beta, threshold
- Loss: Burst Uniform law: $f(x) = dx/(\beta - \alpha)$
  Parameters: α, β, threshold(n), threshold(n + x), depth
- Loss: User-defined File
  Parameters: file name, threshold
- Loss: Percentage
  Parameter: percentage
- Loss: 1/N (1 packet is lost every N received packets)
  Parameter: range(N)
- Duplication: Percentage (send n times the received packet)
  Parameters: percentage, Min ≤ n ≤ Max
- Duplication: 1/M (duplicate 1 packet n times every M received packets). Parameters: range(M), Min ≤ n ≤ Max
- Duplication: Uniform $f(x) = dx/(\beta - \alpha)$
  Parameters: alpha, beta, threshold
- Loss (1/N) then Duplication (1/M): the loss law (1/N) is used first before the duplication law (1/M)

Button	*Action*
Save Parameter Changes	Temporary saves the parameters of the current law.

⇨ Action buttons

The "Loss & Duplication Laws" window handles a temporary list of laws until the user press the **OK** or **Cancel** button.

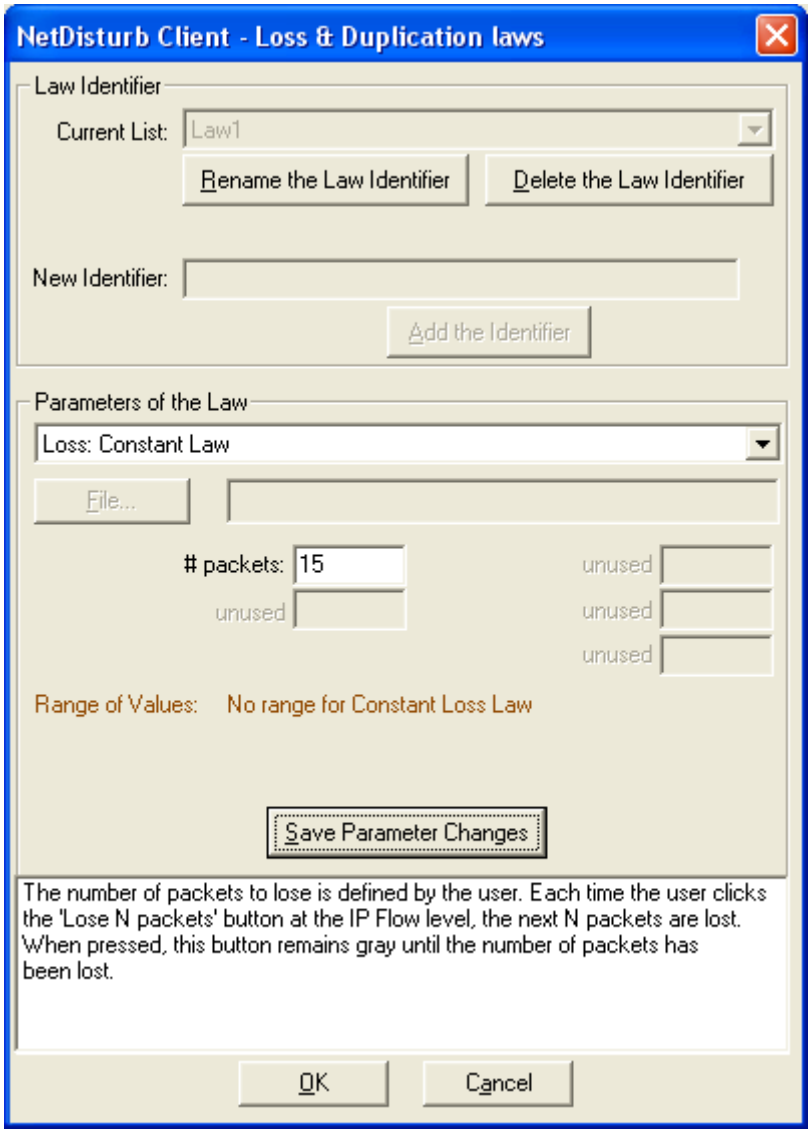
Button	*Action*
OK	Permanently saves changes (addition, deletion and parameters changes) and closes the window.
Cancel	Allows ignoring all modifications made since the window has been opened.

How to create a new Loss & Duplication Law:

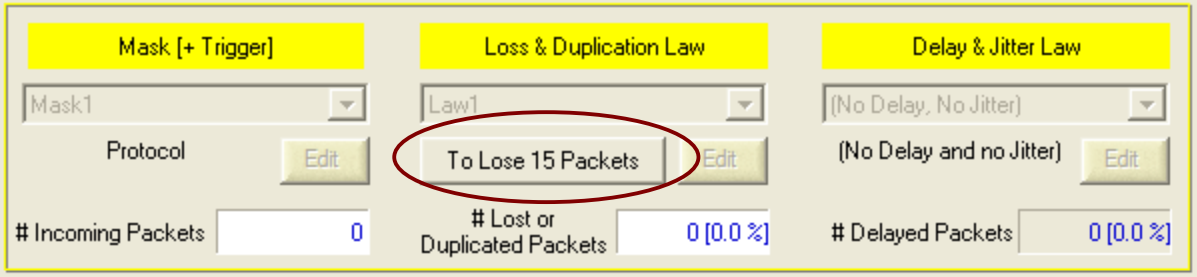
1. Enter a name in the “Add a new Law Identifier” edit field,
2. Then click on the “Add the Identifier” button.
3. Select one kind of law in the ‘Parameters of the Law’,
4. Enter law parameter(s),
5. Press the “Save Parameters Changes“ button.
6. Press “OK” to quit the “Loss & Duplication Laws” window and to save new Identifiers and changes.

7.4.3.3 Loss: Constant Law

When this law is selected, the **NetDisturb** driver will lose the number of packets defined.

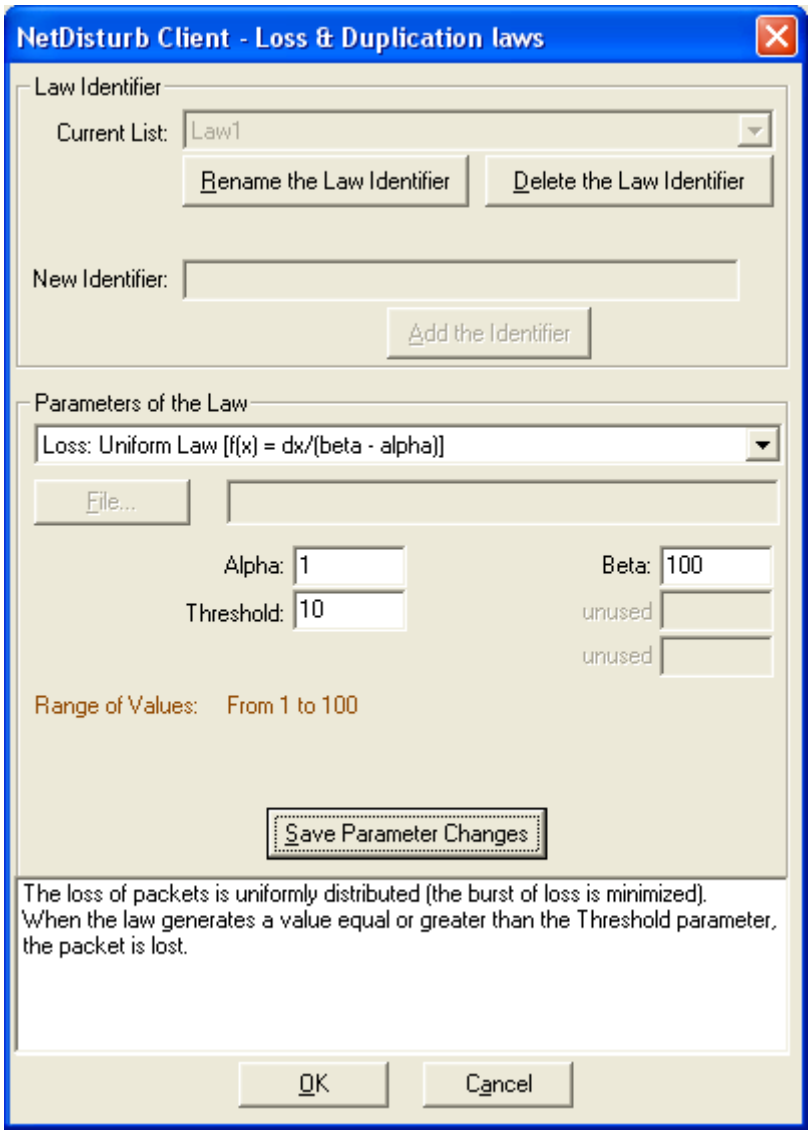


For this law, only one parameter must be defined: **# packets**
A button «Lose xx packets» replaces the summary area in the main window as shown below when the IP Flow is running.
Each time this button is enabled, you can press it to lose #n packets.



7.4.3.4 Loss: Uniform Law [f(x) = dx/(beta – alpha)]

When this law is selected, a uniform distribution of numbers contained between the **Alpha** and **Beta** values is computed and stored in a table. This table and the threshold (also supplied by the user) are then transmitted to the **NetDisturb** driver.



The **NetDisturb** driver picks a number in the table (see also 7.4.3.1) for each selected packet. If this number is greater or equal than the threshold, then the packet is lost.

The mathematical function used is (see the Uniform Law in PART 9 for more information):

Uniform law on (α, β) range

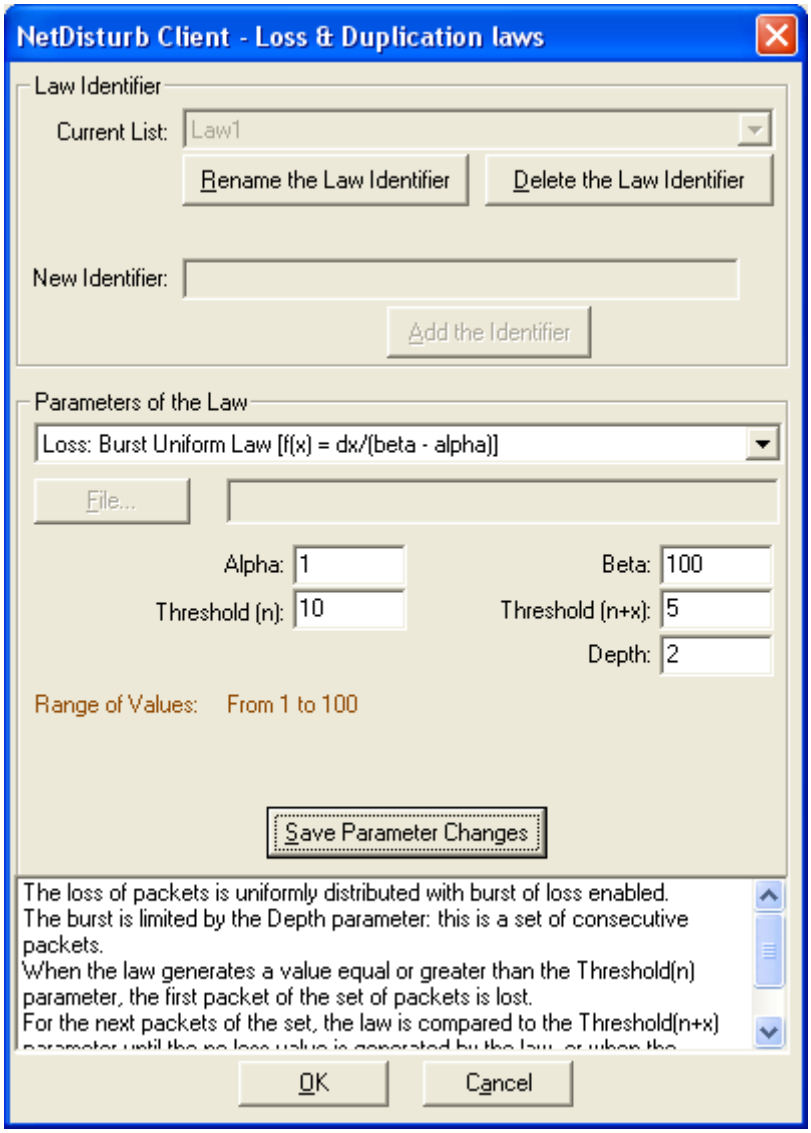
$f(x) = 1/(\beta - \alpha)$ if $\alpha < x < \beta$

$f(x) = 0$ else

For this law, three parameters are defined:

Alpha: min value of the range
Beta: max value of the range
Threshold: if the number calculated by the law is greater or equal than the Threshold value, the packet is lost.

7.4.3.5 Loss: Burst Uniform Law [f(x) = dx/(beta – alpha)]



When this law is selected, the loss of packets is uniformly distributed with burst of loss enabled.
The burst is limited by the **Depth** parameter: this is a set of consecutive packets.

When the law generates a value equal or greater than the **Threshold(n)** parameter, the first packet of the set of packets is lost.
For the next packets of the set, the value of the law is compared to the **Threshold(n+x)** parameter until the no loss value is generated by the law, or when the number of lost packets equals the **Depth** value.

As for the Uniform Law, the Burst Uniform Law calculates a table of numbers uniformly distributed between **Alpha** and **Beta**. This table is transmitted to the

NetDisturb driver with two thresholds T1 (**Threshold (n)**) and T2 (**Threshold (n+x)**) and the **Depth** value (D).
The T1 threshold is the first loss factor.
The T2 threshold is the second loss factor, used in correlation with T1 and for a maximum number of packets defined by the D parameter. T2 may be greater or lower than T1.
This law allows generating burst losses. Processing is applied as follows:

- ⇨ The **NetDisturb** driver picks a number from the table for each packet (see also 7.4.3.1)
- ⇨ For the packet n, the **NetDisturb** driver picks one number from the table (current number) and loses it if this number is greater or equal than T1.
- ⇨ If the packet n is lost, the following packets (up to n+D) will be lost if the picked up number is superior to T2. This threshold (T2) is used to process the following D (depth) packets with the following rules:
  - If the packet n+i (with i < D) is not lost, the threshold comes back to T1 (the burst loss is stopped).
  - If the packets (from n+1 up to n+D) are all lost, the threshold comes back to T1 (the burst loss is stopped).

The mathematical function used is (see the Uniform Law in PART 9 for more information):

Uniform law on (α, β) range

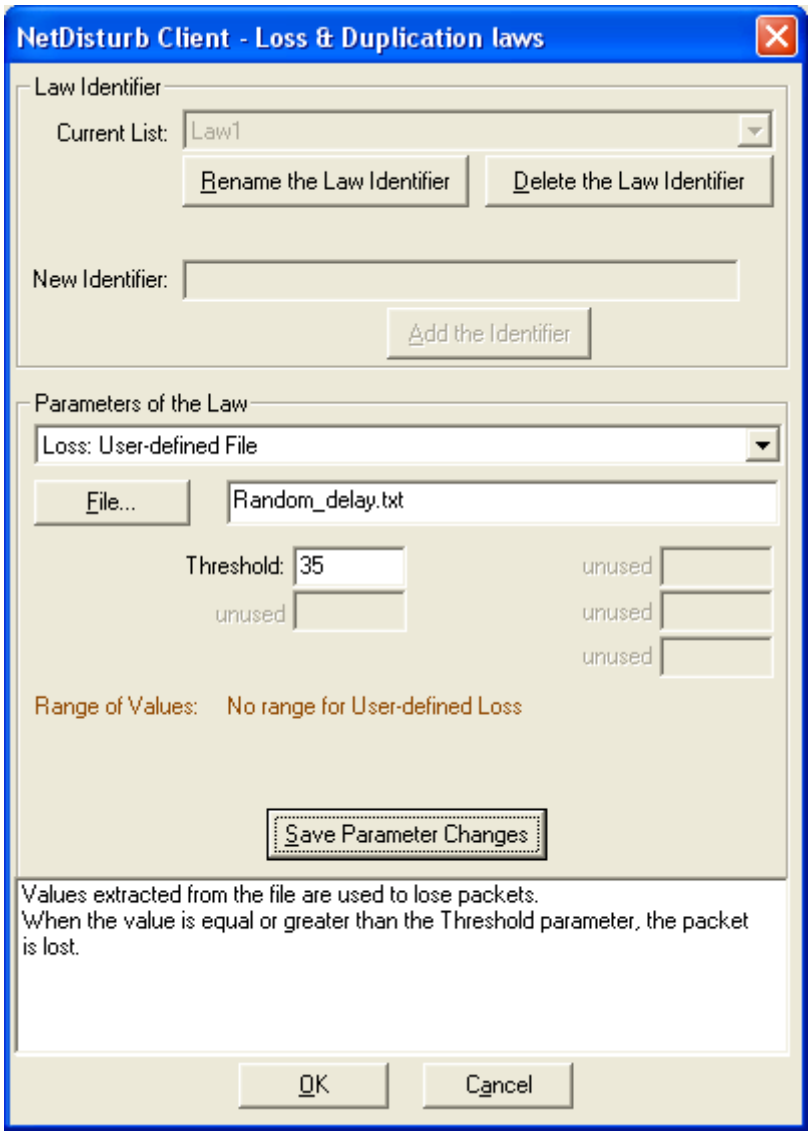
$f(x) = 1/(\beta - \alpha)$ if $\alpha < x < \beta$

$f(x) = 0$ else

For this law, three parameters are defined:

Alpha: min value of the range
Beta: max value of the range
Threshold: if the number calculated by the law is greater or equal than the Threshold value, the packet is lost.

7.4.3.6 Loss: User-defined File



When this law is selected, the loss values are extracted from the user-defined file. This file must be a text file.
Losses are expressed in integer positive number. Separators used for decoding are End of Line (CR or CR-LF), semicolon, coma, and tab or space characters.

If a line starts with a sharp (#) character, the rest of the line is ignored i.e. up to the End of Line (CR or CR-LF).

To assure performance, the file is read in one shot and stored in memory at law selection time. The values of the file are used to load the table transmitted to the **NetDisturb** driver. In order to not overload the memory resources, maximum read number of loss values is limited to 40 960.

If the file size exceeds the table size, only the part that can be memorized will be read. The rest of the file will be used for the next loading.

If the file size is too small to fulfill the table, the file is read again from the beginning to complete the table.

The **NetDisturb** driver picks a number in the table (see also 7.4.3.1) for each selected packet. If this number is greater or equal than the threshold, then the packet is lost. When the end of the file is reached, the **NetDisturb** driver restarts with the first values of the file.

The file sample (OnePerTen.txt) illustrates a loss of 1 packet for 10 packets sent when the Threshold value τ is $0 < \tau < 100$.
(The content of the file OnePerTen.txt is: `0 0 0 0 0 0 0 0 0 100)`

- For any Threshold value greater than 1 and smaller or equal than 100, only the 10th packet is lost.
- If the Threshold value is greater than 100, no packet is lost.
- If the Threshold value is 0, all packets are lost.

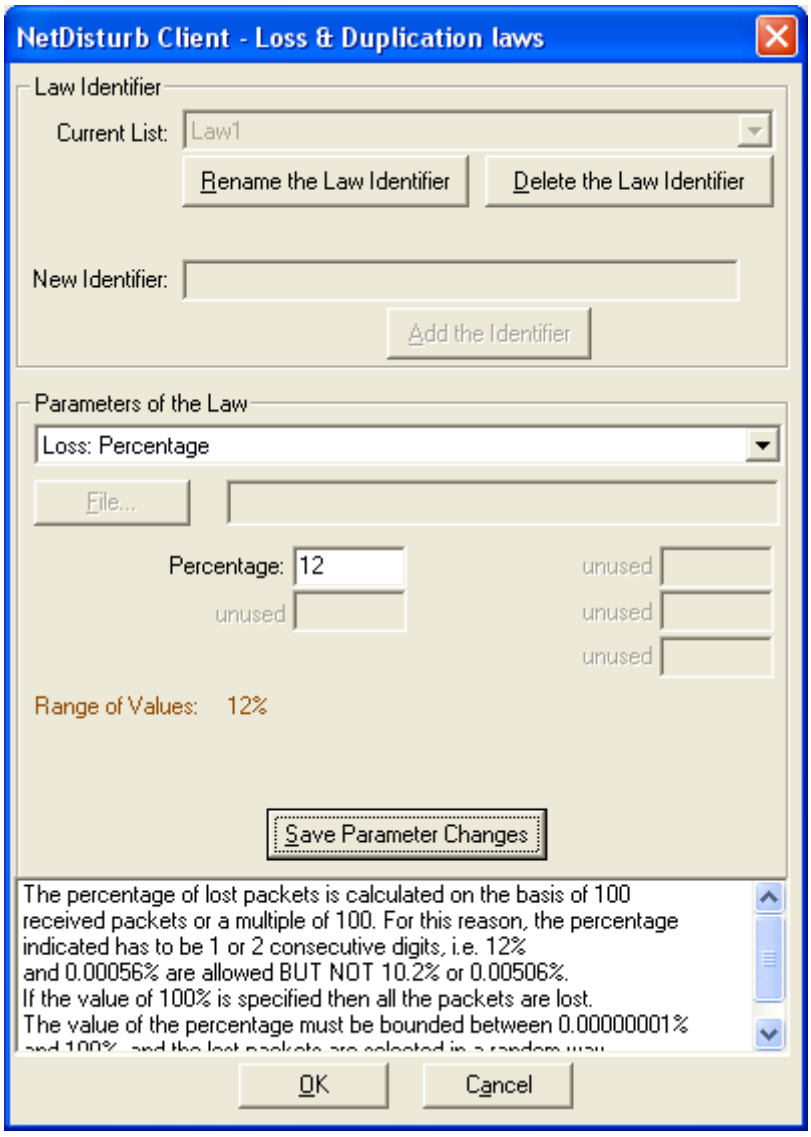
Here is another example of the impact of the threshold value. The content of the file is: `10 20 30 40 50 60 70 80 90 100`

Packet #	Value extracted	Lost result with Threshold = 95	Value extracted	Lost result with Threshold = 50	Value extracted	Lost result with Threshold = 15
1	10	*Continue*	10	*Continue*	10	*Continue*
2	20	*Continue*	20	*Continue*	20	**LOST**
3	30	*Continue*	30	*Continue*	30	**LOST**
4	40	*Continue*	40	*Continue*	40	**LOST**
5	50	*Continue*	50	**LOST**	50	**LOST**
6	60	*Continue*	60	**LOST**	60	**LOST**
7	70	*Continue*	70	**LOST**	70	**LOST**
8	80	*Continue*	80	**LOST**	80	**LOST**
9	90	*Continue*	90	**LOST**	90	**LOST**
10	100	**LOST**	100	**LOST**	100	**LOST**
11	10	*Continue*	10	*Continue*	10	*Continue*
12	20	*Continue*	20	*Continue*	20	**LOST**
13	30	*Continue*	30	*Continue*	30	**LOST**
14	40	*Continue*	40	*Continue*	40	**LOST**
15	50	*Continue*	50	**LOST**	50	**LOST**
16	60	*Continue*	60	**LOST**	60	**LOST**
17	70	*Continue*	70	**LOST**	70	**LOST**
18	80	*Continue*	80	**LOST**	80	**LOST**
19	90	*Continue*	90	**LOST**	90	**LOST**
20	100	**LOST**	100	**LOST**	100	**LOST**
21	10	*Continue*	10	*Continue*	10	*Continue*

Note: *Continue* means the packet is not lost and may be handled by the Delay & Jitter Law if defined.

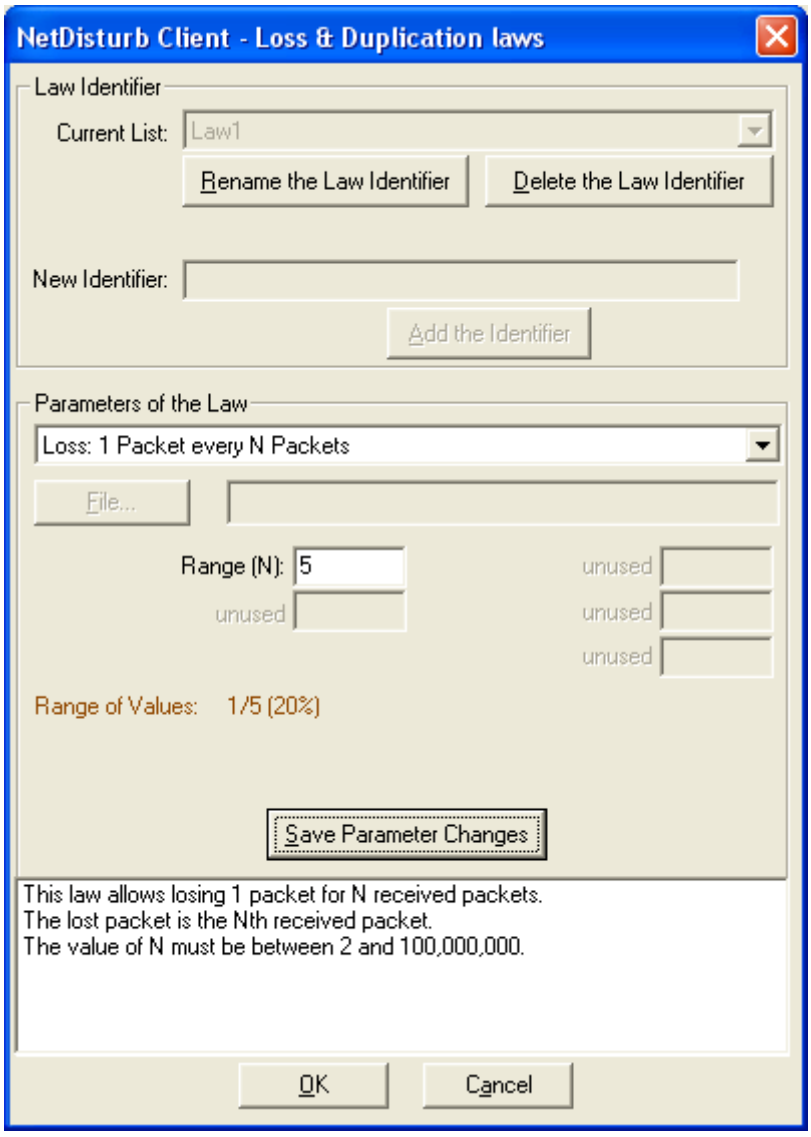
A more detailed description with delays and jitters is also available in the paragraphs 7.4.5 and 7.4.6.

7.4.3.7 Loss: Percentage



When this law is selected, a percentage of packets are lost and the packets to lose are randomly selected.
The percentage of lost packets is calculated on the basis of 100 received packets or a multiple of 100. For this reason, the percentage indicated has to be 1 or 2 consecutive digits, i.e. 12% and 0.00056% are allowed BUT NOT 10.2% or 0.00506%.
If the value of 100% is specified then all the packets are lost.
The value of the percentage must be bounded between 0.00000001% and 100%, and the lost packets are selected in a random way.

7.4.3.8 Loss: 1 Packet every N Packets



This law allows loosing 1 packet for N received packets. It affects a same packet based on its order.

The lost packet is the Nth received packet, i.e. considering N is 12, then the 12th, 24th, 36th packet and so on are lost.

The value of N must be between 2 and 100,000,000.

7.4.3.9 General Rules concerning the Duplication of Packets

This paragraph details some general terms used to describe the Duplication of packets.

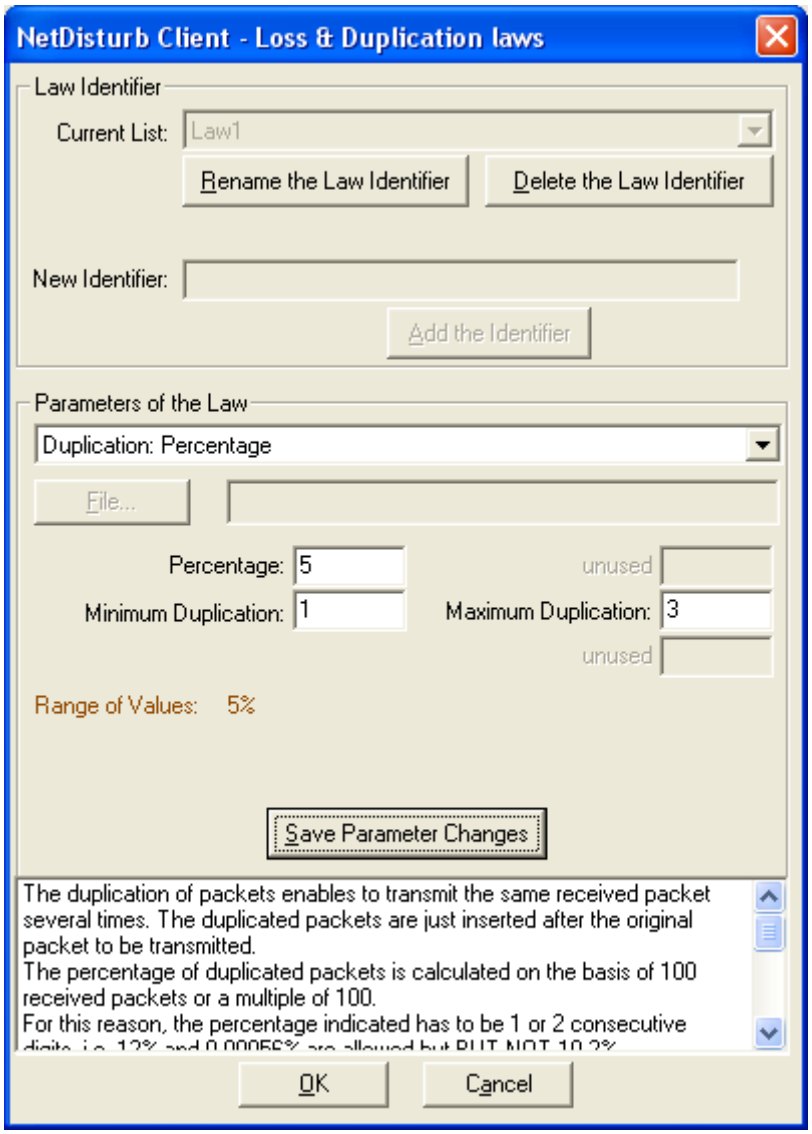
7.4.3.9.1 What does Duplication mean with **NetDisturb**

The duplication refers to the action to send more than once the same packet. If the packet N should be duplicated, the packet N is send at least twice consecutively.

7.4.3.9.2 How many times is a packet duplicated

The Minimal Duplication and Maximal Duplication parameters help to select the number of times the packet should be duplicated. When those parameters have the same value, the number of duplications is constant. Otherwise, the number of duplications is randomly selected, where the smallest value is “Minimal Duplication” and the highest value is “Maximal Duplication”.

7.4.3.10 Duplication: Percentage



The duplication of packets enables to transmit the same received packet several times. The duplicated packets are just inserted after the original packet to be transmitted.

The **Percentage** of duplicated packets is calculated on the basis of 100 received packets or a multiple of 100.
For this reason, the percentage indicated has to be 1 or 2 consecutive digits, i.e. 12% and 0.00056% are allowed BUT NOT 10.2% or 0.00506%. If the value of 100% is specified then all the packets are duplicated.
The value of the percentage must be bounded between 0.00000001% and 100%, and the packets to duplicate are selected in a random way.

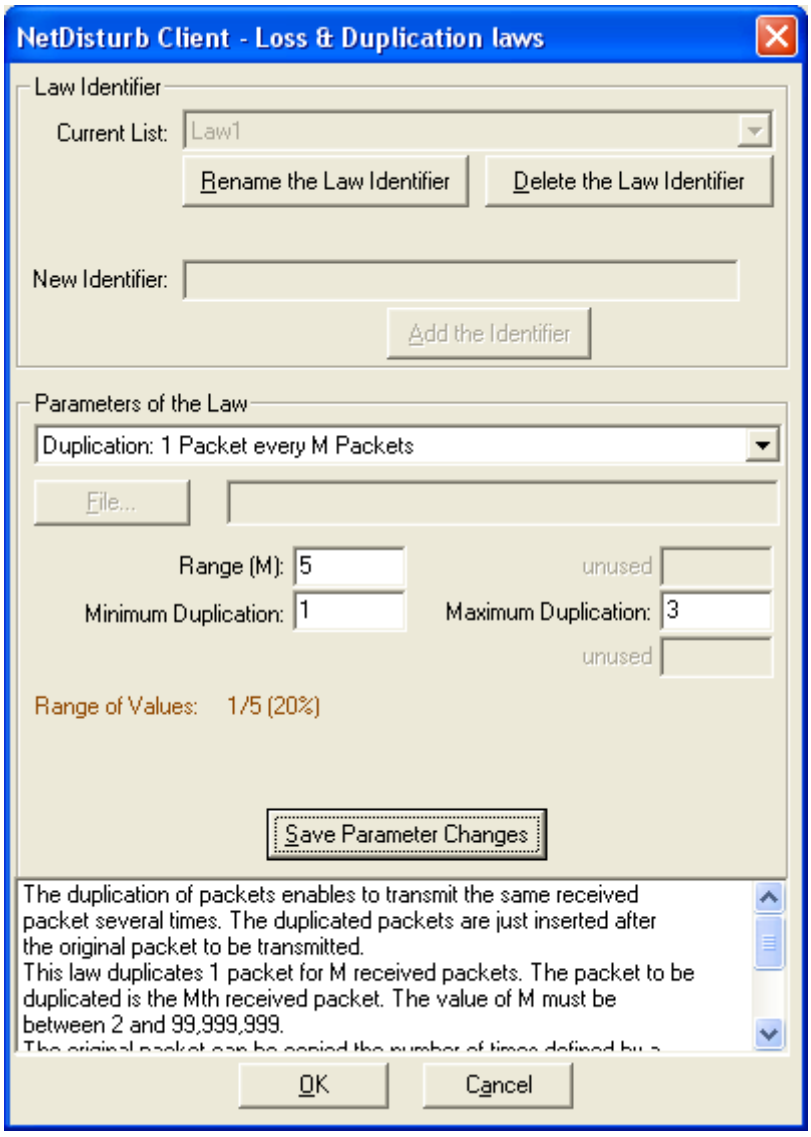
The original packet can be duplicated for a number of times defined by a random selection in the '**Minimum Duplication**' and '**Maximum Duplication**' range parameters.
When '**Minimum Duplication**' equals '**Maximum Duplication**', all the eligible packets for the process of duplication are copied the same number of times.

Here are a few examples:

- If the Percentage is 10, 10 packets are duplicated each 100 received packets.
- If the Percentage is 5, 5 packets are duplicated each 100 received packets.
- If the Percentage is 2.5, 25 packets are duplicated each 1,000 received packets.
- If the Percentage is 0.012, 12 packets are duplicated each 100,000 received packets.

See also paragraph 7.4.3.9 for the general rules and terms relevant to the duplication of packets.

7.4.3.11 Duplication: 1 Packet every M Packets



This duplication of packets enables to transmit the same received packet several times. The duplicated packets are just inserted after the original packet to be transmitted.
This law duplicates 1 packet for M received packet and the packet to be duplicated is the Mth received packet.
The value of M must be between 2 and 99,999,999.
The original packet can be copied the number of times defined by a random selection in the '**Minimum Duplication**' and '**Maximum Duplication**' range parameters.
When '**Minimum Duplication**' equals '**Maximum Duplication**', all the eligible packets for the process of duplication are copied the same number of times.

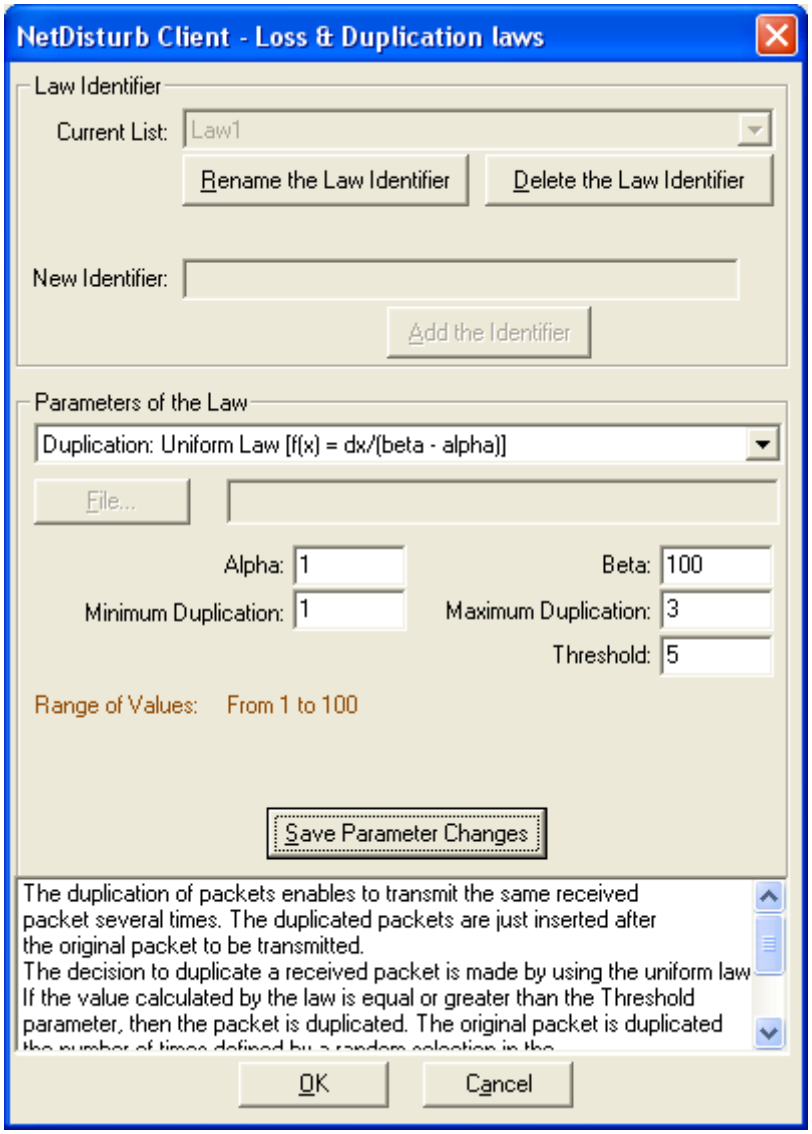
The **Range (M)** indicates which packet is going to be duplicated i.e. considering M is 9, then the 9^{h}, 18^{th}, 27^{th} packet and so on are duplicated.

See also paragraph 7.4.3.9 for the general rules and terms relevant to the duplication of packets.

7.4.3.12 Duplication: Uniform Law [f(x) = dx/(beta – alpha)]

The duplication of packets enables to transmit the same received packet several times. The duplicated packets are just inserted after the original packet to be transmitted.
The decision to duplicate a received packet is made by using the uniform law.

If the value calculated by the law is equal or greater than the **Threshold** parameter, then the packet is duplicated.
The original packet is duplicated the number of times defined by a random selection in the '**Minimum Duplication**' and '**Maximum Duplication**' range parameters.
When '**Minimum Duplication**' equals '**Maximum Duplication**', all the eligible packets for the process of duplication are copied the same number of times.



When this law is selected, a uniform distribution of numbers between the **Alpha** and **Beta** values is computed and stored in a table. This table and the **Threshold** value are then transmitted to the **NetDisturb** driver.

The **NetDisturb** driver picks a number in the table for each selected packet. If this number is greater or equal than the **Threshold**, then the packet is duplicated.

The mathematical function used is (see the Uniform Law in PART 9 for more information):

Uniform law on (α, β) range

$f(x) = 1/(\beta - \alpha)$ if $\alpha < x < \beta$

$f(x) = 0$ else

For this law, three parameters are defined:

Alpha: min value of the range
Beta: max value of the range
Threshold: if the number calculated by the law is equal or greater than the Threshold value, the packet is duplicated.

See also paragraph 7.4.3.9 for the general rules and terms relevant to the duplication of packets.

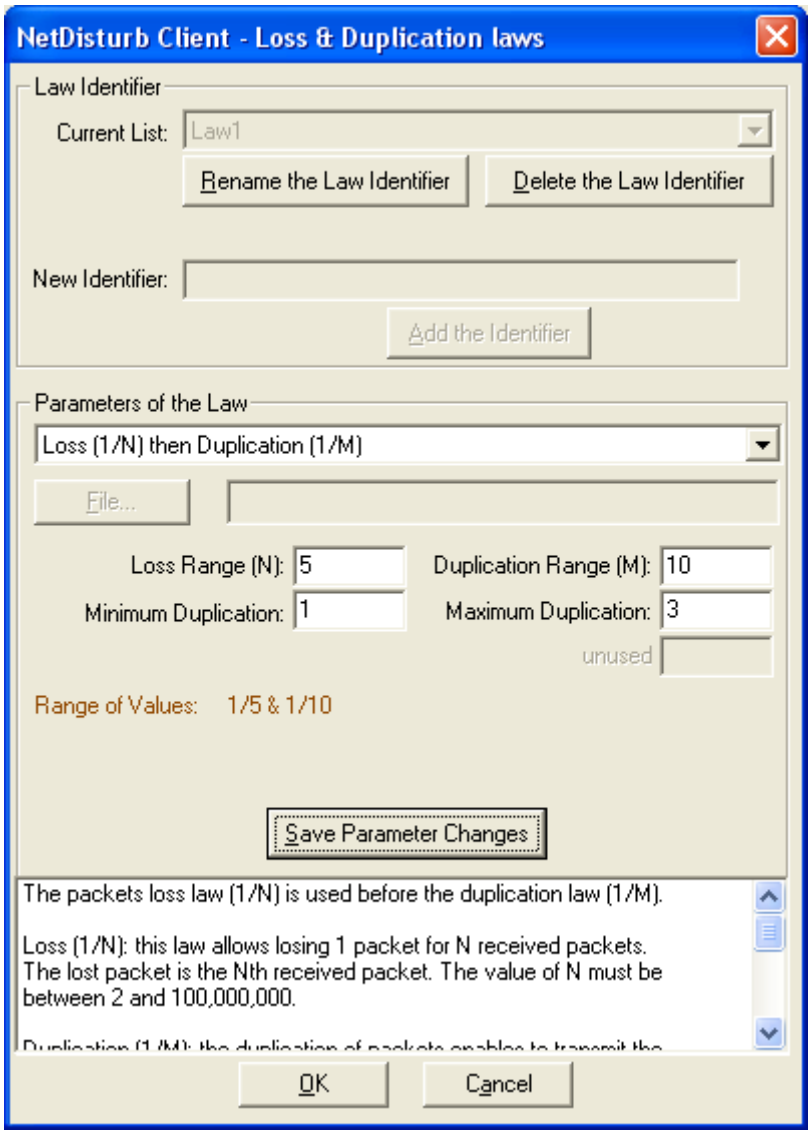
7.4.3.13 Loss (1/N) then Duplication (1/M)

The packets loss law (1/N) is used before the duplication law (1/M).

Loss (1/N): this law allows losing 1 packet for N received packets. The lost packet is the Nth received packet. The value of N must be between 2 and 100,000,000.
For this law, 1 parameter is used: **Loss Range (N)**
Refer to paragraph 7.4.3.8 Loss: 1 Packet every N Packets for more details.

Duplication (1/M): the duplication of packets enables to transmit the same received packet several times. The duplicated packets are just inserted after the original packet to be transmitted. The value of M must be between 2 and 100,000,000.
The original packet is duplicated the number of times defined by a random selection in the '**Minimum Duplication**' and '**Maximum Duplication**' range parameters.
When '**Minimum Duplication**' equals '**Maximum Duplication**', all the eligible packets for the process of duplication are copied the same number of times.
For this law, 3 parameters are used: **Duplication Range (M)**, **Minimum Duplication** and **Maximum Duplication**.

Refer to paragraph 7.4.3.11 Duplication: 1 Packet every M Packets for more details.



Let's take an example of 100 packets received with **Loss Range(N)** = 10 and **Duplication Range(M)** = 20.
The lost packets are the 10th, the 20th, the 30th, the 40th … the 100th.
The duplicated packets are the 22nd (because packet #10 and #20 have been lost), the 44th (because packet #30 and #40 have been lost and because the first packet of the next set of 20 none lost packets is the 23rd), the 66th (because packet #50 and #60 have been lost, and the first packet of this 20 packet set is the 45th) and the 88th (because packet #70 and #80 have been lost with a 20 packets set starting at 67th).

See also paragraph 7.4.3.9 for the general rules and terms relevant to the duplication of packets.

7.4.4 The Delay & Jitter Law Configuration

NetDisturb can delay the IP packets following a mathematical law configured by the user or using values extracted from an input file. These values apply to the IP packets matching to the selected mask, if a loss law hasn't previously lost the packets.

If the value is constant, it is a Delay. When values vary, that is the case with mathematical laws, it is a Delay & Jitter value.

Up to 100 Delay & Jitter Laws can be created.

By default the following laws are defined in the Default.wsx context file:

Combo-box	Comment area	Description
(No Delay, No Jitter)	(No Delay, No Jitter)	With this option, no delay or jitter is applied to the IP flow.
Constant delay	Constant Delay	A 20 ms delay is applied to IP packets
Exponential jitter	Constant Delay & Exponential Jitter	Delay & Jitter to apply: from 20 to 124 ms. The delay is 20 ms and the jitter varies from 0 to 104 ms.
Uniform jitter	Constant Delay & Uniform Jitter	Delay & Jitter to apply: from 3 to 102 ms. The delay is 2 ms and the jitter varies from 1 to 100 ms.
Constant Delay & User File Jitter	Constant Delay & User File	The file Random_delay.txt contains jitter values to add to the constant 10 ms delay.
User File Delay & Jitter	User File with Constant Delay & Jitter	The file RandomValues.txt contains values used as Delay & Jitter.
Router Simulation with Delay	Constant Delay & Router Simulation	Constant delay = 20 ms IP Throughput = 1000 Kb/s Max memory = 500 Ko
Router Simulation & User File	Router Simulation & User File with Delay and Jitter	IP Throughput = 1000 Kb/s Max memory = 250 Ko Delay & Jitter values are extracted from a user file (RandomValues.txt).
Delay & Throughput with Duration in a File	Constant Delay & File with Throughput with Duration	Constant delay = 250 ms Throughput values and Duration of the Throughput values are extracted from a user file (ThroughputAndDurationSample.txt).

7.4.4.1 Delay & Jitter Law and the Working Mode

Working Mode: Laws apply to the IP Flow

When a Delay & Jitter Law is selected for a given IP Flow, the law applies to all packets matching the mask that haven't been lost. For each packet, a new Delay & Jitter value is extracted from the law or from the file, depending on the type of law selected. These values are stored in a global table by the **NetDisturb** driver. When the table is empty, the **NetDisturb** Server provides a new table to the **NetDisturb** driver with new values provided by the law or the file.
The value is the number of milliseconds the packet is delayed.

Working Mode: Laws apply to each TCP/UDP connection of the IP Flow

When a Delay & Jitter Law is selected for a given IP Flow, the law applies to all packets matching the mask that haven't been lost.
These values are stored in a table maintained by the **NetDisturb** driver.
The **NetDisturb** Server provides the table once to the **NetDisturb** driver with values provided by the law or extracted from the file. The **NetDisturb** driver loops on values from this table: when the end of the table is reached, **NetDisturb** driver restarts at the beginning.
If the packet is TCP or UDP, the 5-tuple IP addresses, protocol and ports is used to classify the packet. Else the IP addresses and protocol are only used.
For each packet, a Delay & Jitter value is extracted from the buffer at the current index of the packet for the connection i.e. the n^{th} packet received for the given connection is delayed by the n^{th} value of the table. When ***n*** reaches the end of the table, the values extracted restart at the beginning of the table.

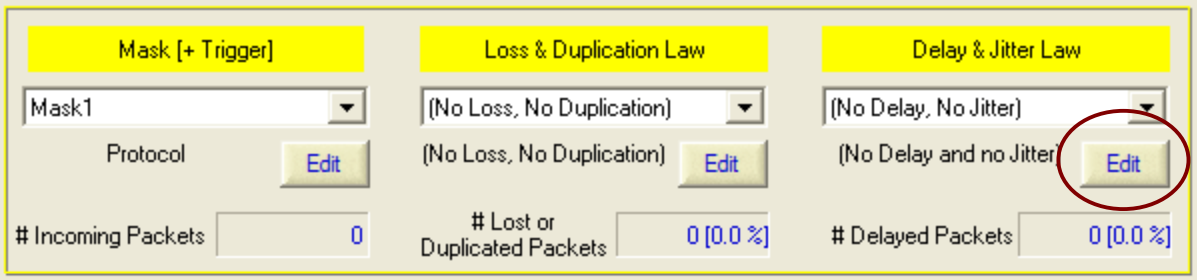
7.4.4.2 Delay & Jitter Accuracy

The **NetDisturb** driver accuracy is ±2 milliseconds.
It means that a delay variation of one millisecond between two packets can't be taken into account. With such Delay & Jitter, the result is either no Delay & Jitter or a Delay & Jitter of 2ms at least.

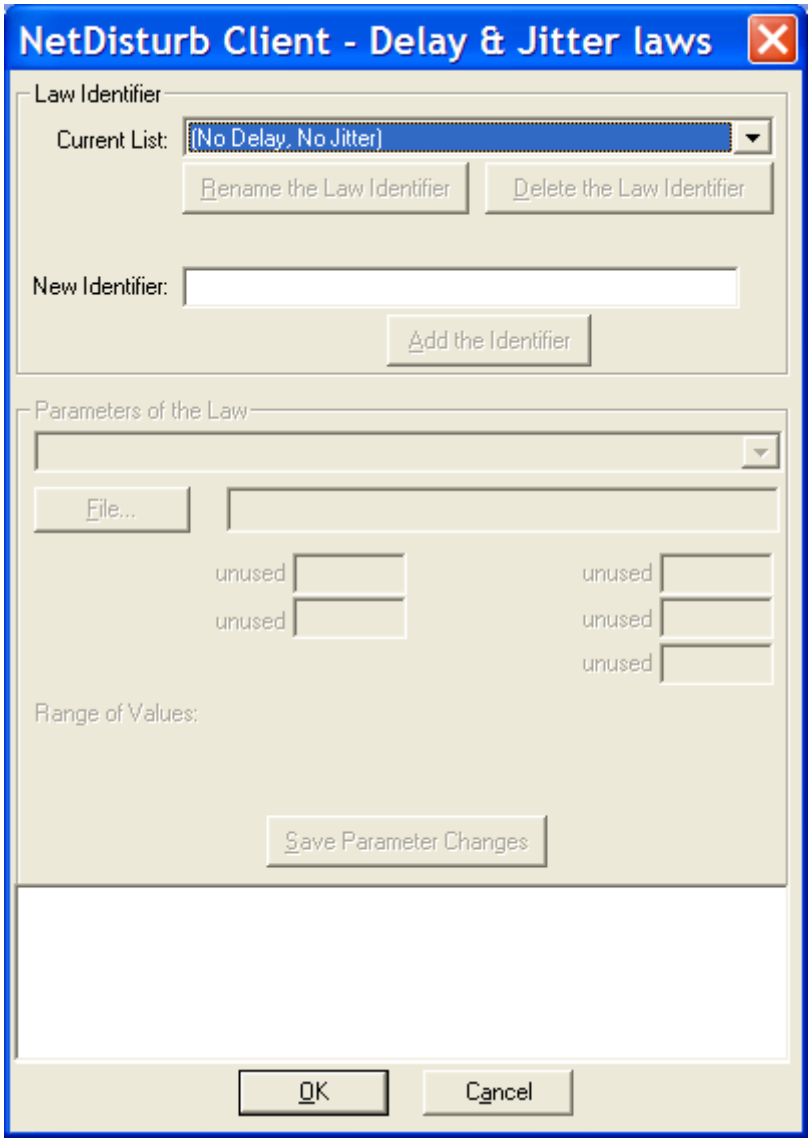
Note: the **NetDisturb** driver uses the OS timer accuracy to delay the packets. Because Windows is not a real-time OS, it may append Windows is not able to wake up the **NetDisturb** driver in the timely manner. In such case, the delay and/or jitter value is increased unexpectedly.

7.4.4.3 How to create or edit the Delay & Jitter Law

To create or configure a Delay & Jitter Law click on the “Edit” button at the top or bottom part of the main window.



The following window is then displayed:



The Delay & Jitter Law window is composed of three sections:

⇨ Law identifier

It is used to choose an existing law from the “Current List” combo-box. An existing law can be deleted by clicking on the “Delete the Law Identifier” button.
You can also add a new law by entering a name in the “New Identifier“ field and by clicking on the “Add the Identifier” button”.

Button	*Action*
Rename the Law Identifier	Change the law identifier.
Delete the Law Identifier	Remove the law from the temporary list.
Add the Identifier	Add the Identifier in the temporary list.

⇨ Parameters of the Law

This section is composed of a list box to select the law to apply, and different edit fields may be enabled in order to input parameters.

The “Value range” allows seeing the range of values generated by the law for the user-defined parameters. It applies to the Uniform Loss law and Burst Uniform Loss law.

A list box allows selecting a law among the following pre-defined laws:

- Constant Delay & No Jitter
  Parameter = constant delay
- Constant Delay & Exponential Jitter law: $f(x) = \lambda e^{-\lambda x} dx$
  Parameters: constant delay, λ
- Constant Delay & Uniform Jitter law: $f(x) = dx/(\beta - \alpha)$
  Parameters: constant delay, alpha, beta
- Constant Delay & User File with Jitter values
  Parameters: constant delay, user file
- User File with Delay & Jitter values
  Parameter: user file
- Router Simulation & Constant Delay
  Parameters: IP throughput, max memory, constant delay
- Router Simulation & User File with Delay and Jitter values
  Parameters: IP throughput, max memory, user file
- Constant Delay & User File with Throughput and Duration values
  Parameters: constant delay, user file

Button	*Action*
Save Parameter Changes	Temporary saves the parameters of the current law.

⇨ Action buttons

The "Loss & Duplication Laws" window handles a temporary list of laws until the user press the **OK** or **Cancel** button.

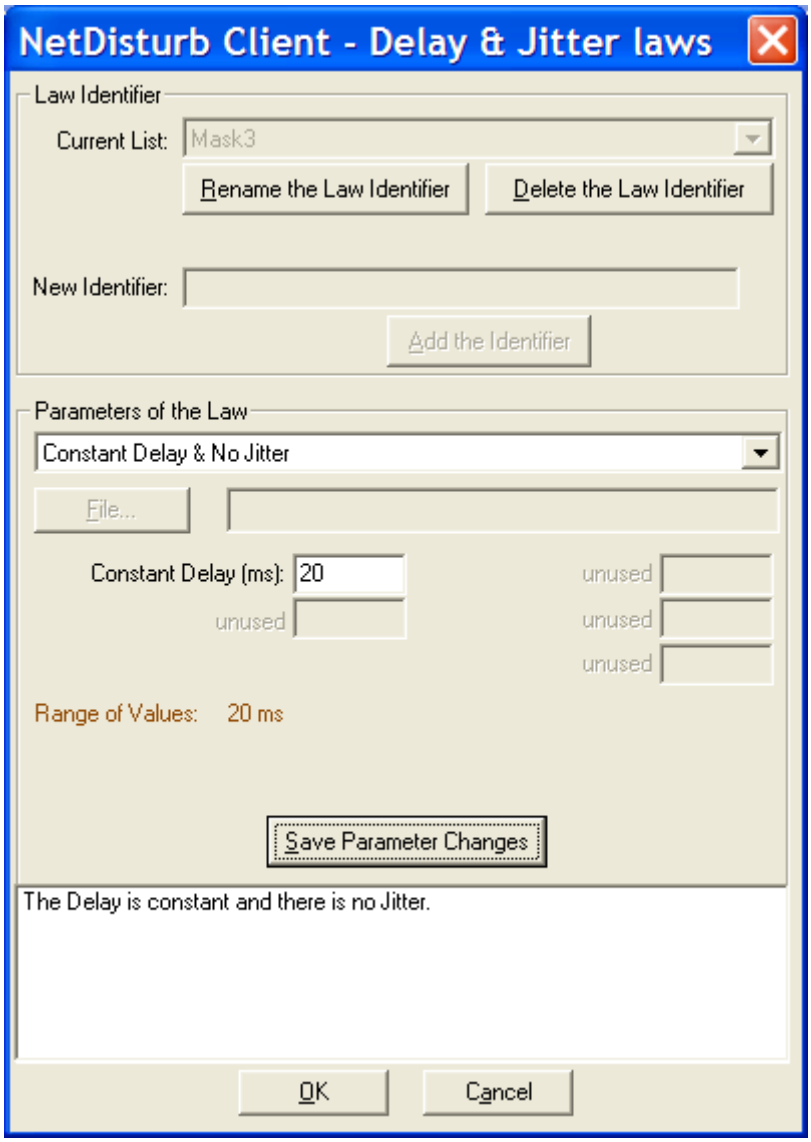
Button	*Action*
OK	Permanently saves changes (addition, deletion and parameters changes) and closes the window.
Cancel	Allows ignoring all modifications made since the window has been opened.

How to create a new Delay & Jitter Law:

1. Enter a name in the “Add a new Law Identifier” edit field,
2. Then click on the “Add the Identifier” button.
3. Select one kind of law in the ‘Parameters of the Law’,
4. Enter law parameter(s),
5. Press the “Save Parameters Changes“ button.
6. Press “OK” to quit the “Delay & Jitter Laws” window and to save new Identifiers and changes.

7.4.4.4 Constant Delay & No Jitter

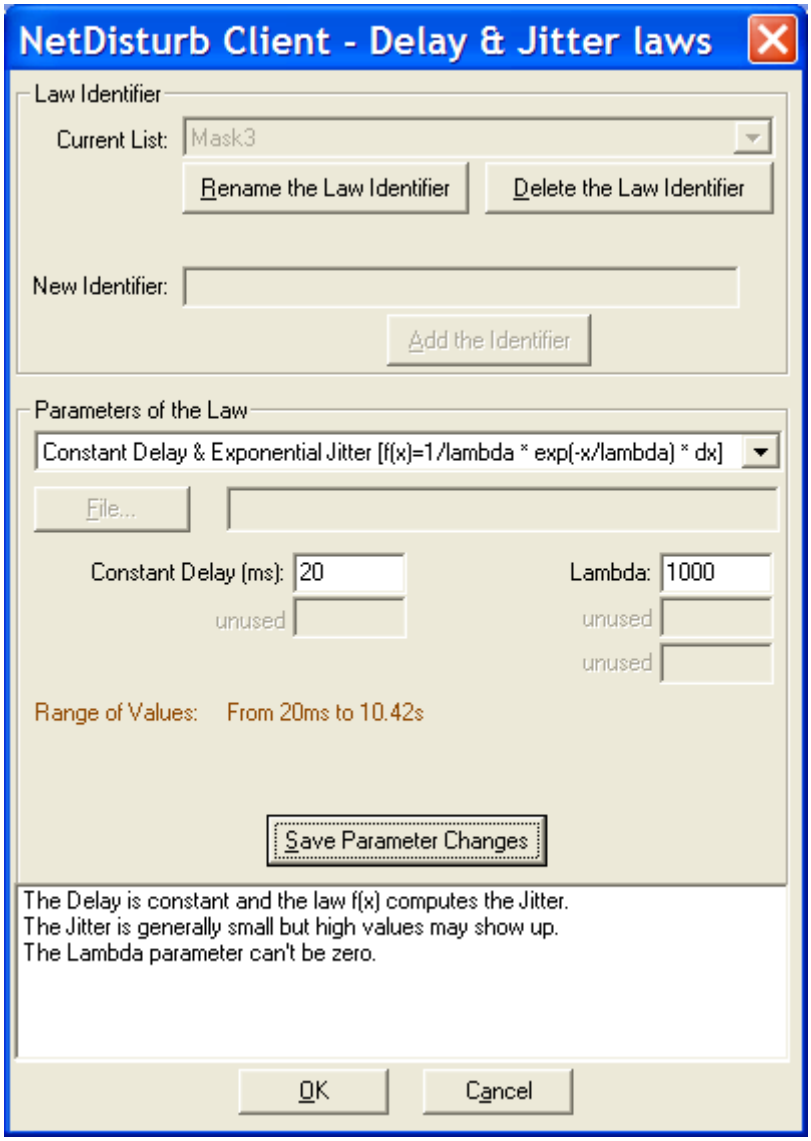
A constant delay is applied to all packets relevant to the IP Flow not previously lost.



The “Constant Delay (ms)” parameter must be only defined, and all packets will be delayed in a constant manner.

7.4.4.5 Constant Delay & Exponential Jitter [f(x) = 1/lambda * exp (-x/lambda) * dx]

When this law is selected, an exponential distribution of the jitter is computed from the Lambda parameter. This distribution is stored in a table. This table is then transmitted to the **NetDisturb** driver, finally coupled with a **Constant Delay** (expressed in ms) that will be added to the calculated jitter.



The mathematical function used is (see the Exponential Law in PART 9 for more information):

Exponential law $(\lambda > 0)$

$f(x) = (1/\lambda)e^{-x/\lambda}$ if $x \geq 0$

$f(x) = 0$ if $x < 0$

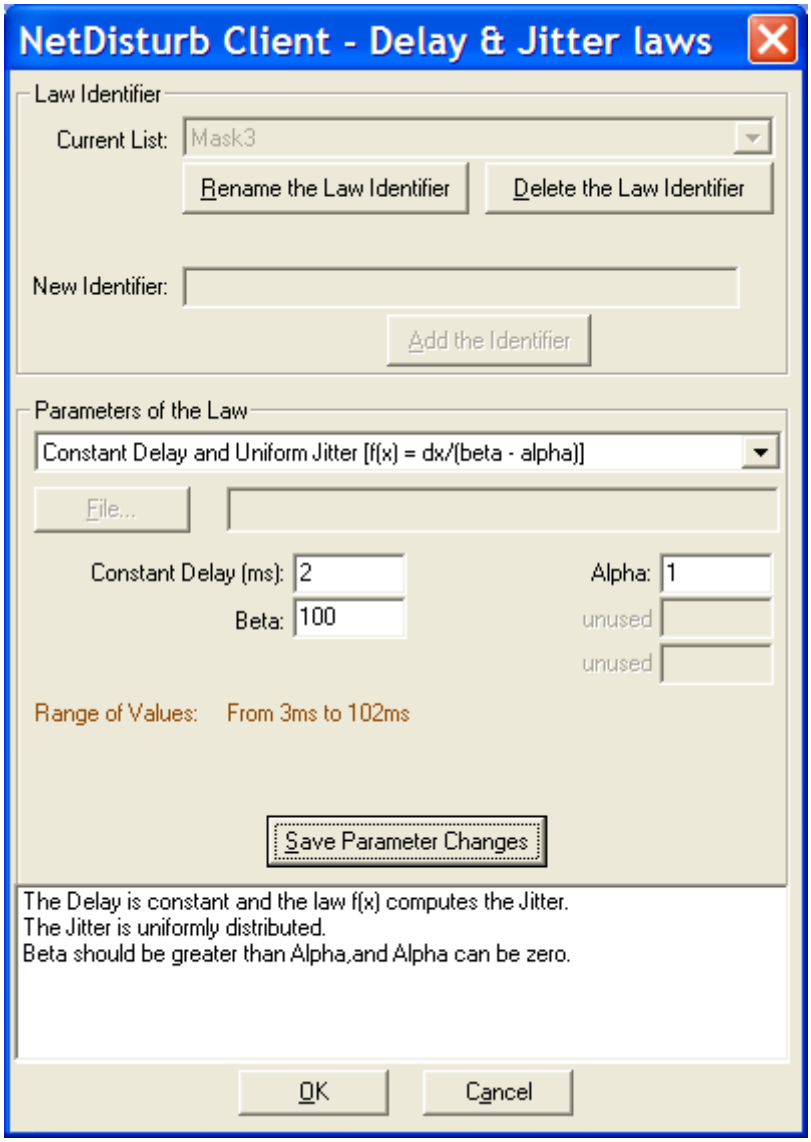
For this law, one parameter is defined:

Lambda parameter of the law

The **Range of Values** area presents the domain of values calculated after entering the Lambda parameter.

7.4.4.6 Constant Delay & Uniform Jitter [f(x) = dx/(beta – alpha)]

When this law is selected, a uniform distribution of jitter values is calculated from the **Alpha** and **Beta** parameters.
This distribution is stored in a table. This table is then transmitted to the **NetDisturb** driver, finally coupled with a **Constant Delay** (expressed in ms) that will be added to the calculated jitter.



The mathematical function used is (see the Uniform Law in PART 9 for more information):

Uniform law on (α, β) range

$f(x) = 1/(\beta - \alpha)$ if $\alpha < x < \beta$

$f(x) = 0$ else

For this law, two parameters are defined:

Alpha min value of the range
Beta max value of the range

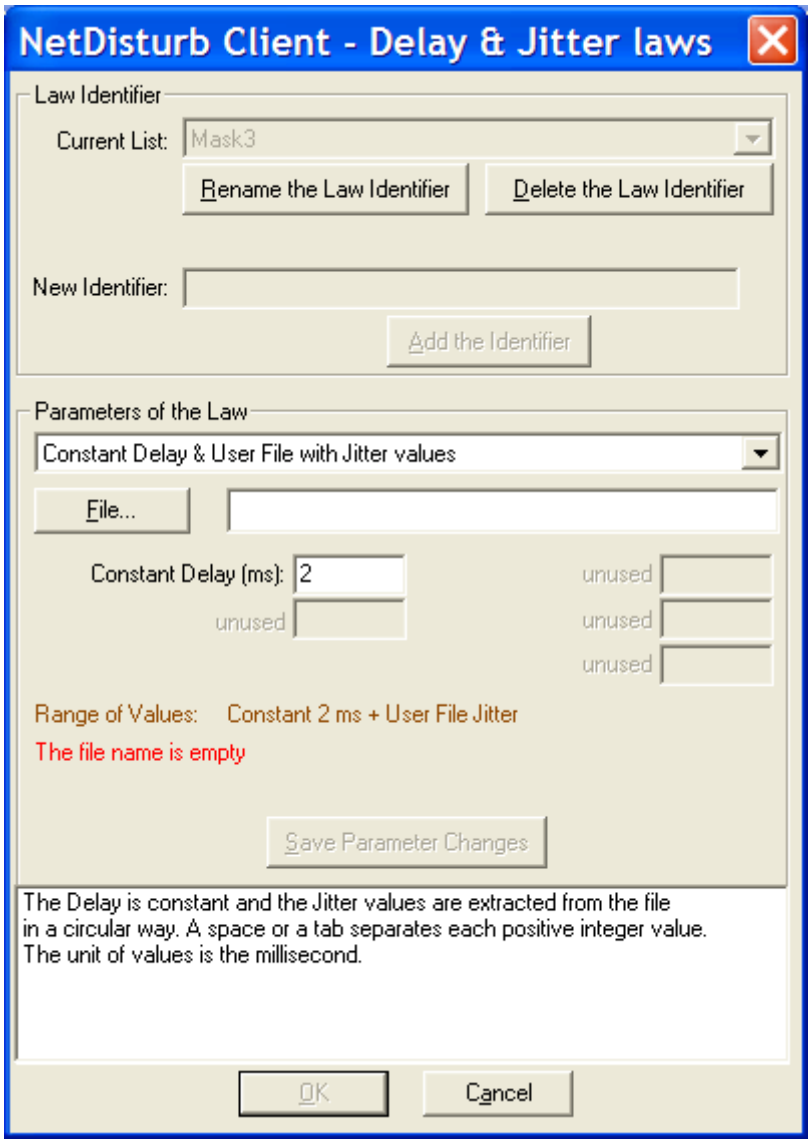
The **Range of Values** area presents the domain of values calculated after entering the **Alpha** and **Beta** parameters.

7.4.4.7 Constant Delay & User File with Jitter Values

When this law is selected, the delay rate is obtained from a file.

Total delay applied to the packet = Constant Delay (expressed in ms) + delay read from the file for this packet.
The Jitter values file must be a text file.
Delays are expressed with an integer positive number. The unit is the millisecond. The separators used for decoding are: end of line (CR or CR-LF), semicolon, coma, and tab or space characters.
If a line starts with a sharp (#) character, the rest of the line is ignored i.e. up to the End of Line (CR or CR-LF).

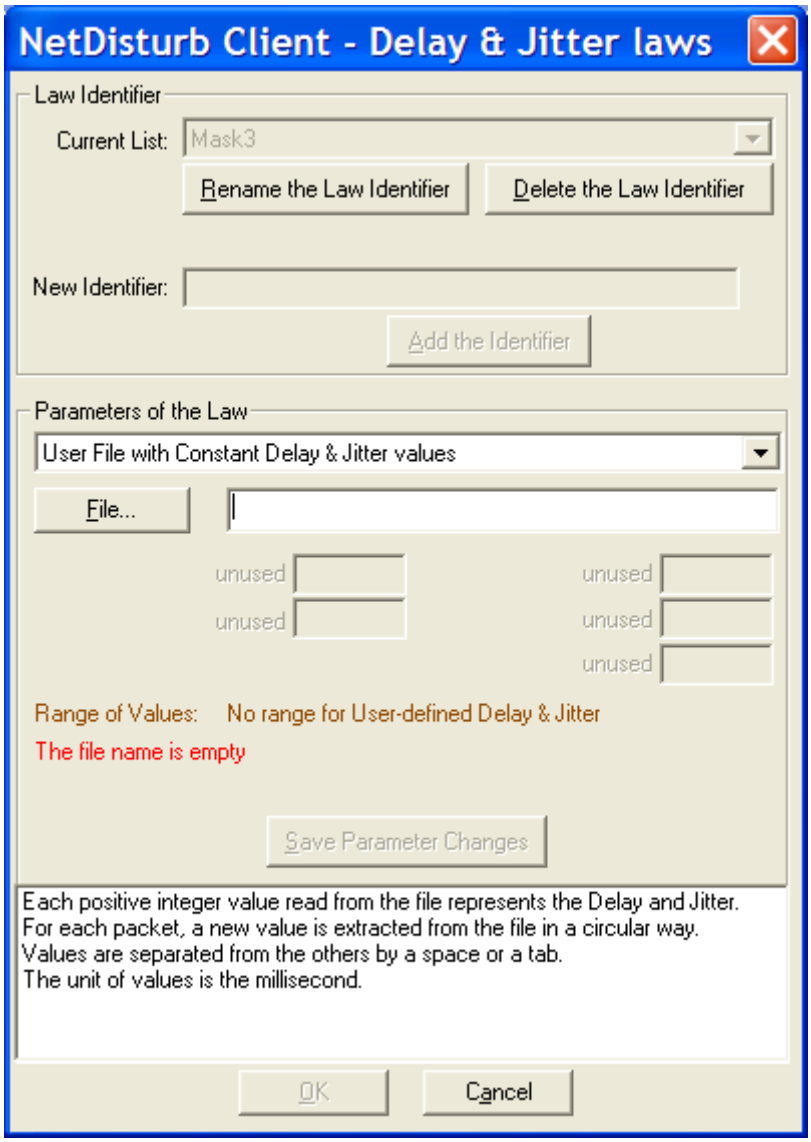
One Jitter value is picked for each packet handled. When the end of the file is reached, the **NetDisturb** driver restarts with the first values of the file.



For performance reasons the file is read in one shot, and stored in memory when the law IP Flow is set in the Run state. The values are used to load the table transmitted to the **NetDisturb** driver. In order to not overload the memory resources, the maximum number of delays read is limited to 40,960.
If the file size exceeds the table size, only the part that can be memorized will be read. The rest of the file will be used for the next loading. If the file size is too small to fulfill the table, fulfillment is done by read back the file from its beginning.

7.4.4.8 User File with Constant Delay & Jitter Values

When this law is selected, the total delay to apply to the packet is read from the user file.
This file containing the Delay and Jitter values must be a text file.
Delays are expressed with an integer positive number. The unit is the millisecond. The separators used for decoding are: end of line (CR or CR-LF), semicolon, coma, and tab or space characters.
If a line starts with a sharp (#) character, the rest of the line is ignored i.e. up to the End of Line (CR or CR-LF).
One Delay/Jitter value is picked for each packet handled. When the end of the file is reached, the **NetDisturb** driver restarts with the first values of the file.



The first packet initializes the T0 time. Then the value T is calculated: T = T0 + D1 (with D1 = first delay read in the user file). T is the time when the second packet must be transmitted on the output interface. The second IP packet is received at the T1 time.

If T1 < T then this second packet is queued with a delay defined as: T0 + D1 – T1
If T1 > or = T then this second packet is sent immediately on the outgoing interface.

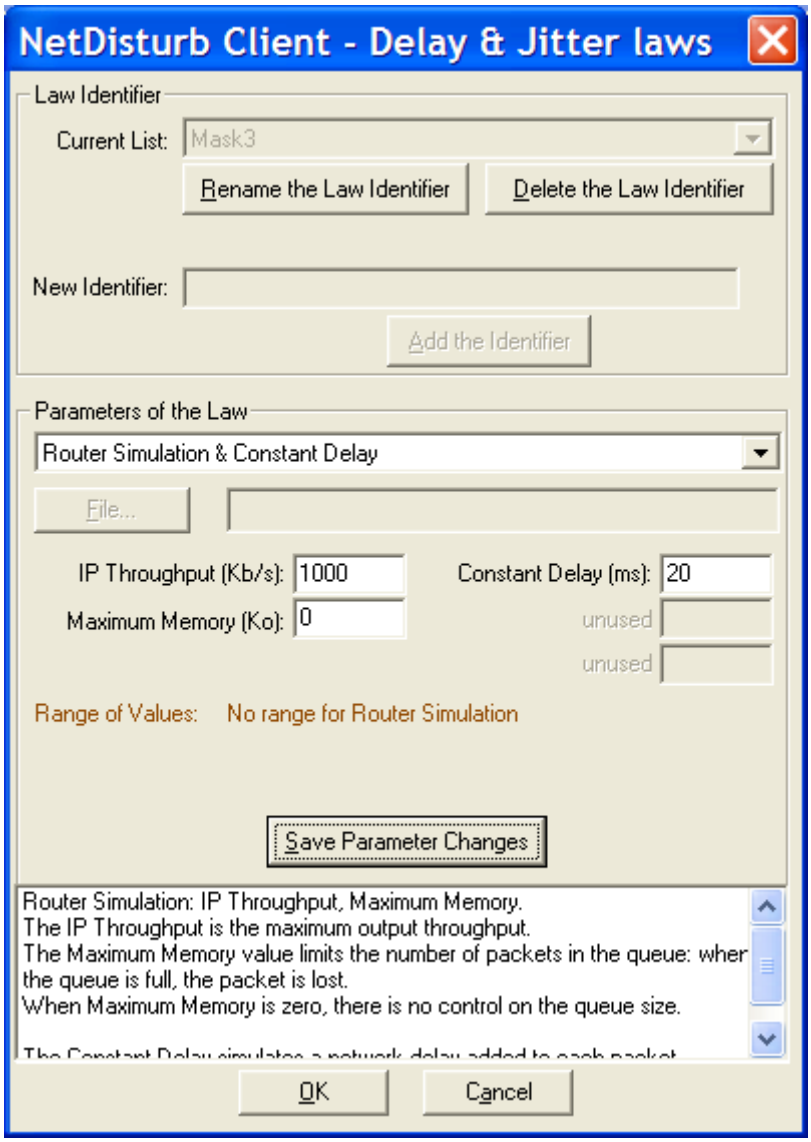
Then the new value T is calculated for the third packet: T = T0 + D1 + D2 (with D2 = second delay read in the user file). T is now the time when the third packet must be transmitted. And the process continues ... when the end of file is reached, the process continues by the beginning of the file and it loops. So the values defined in the user file correspond to inter packet delays.

7.4.4.9 Constant Delay & Router Simulation

This law is used to simulate a real network by offering:

- A limited **IP Throughput** on the output interface in Kbps.
- A **Constant Delay** (to simulate a network transit delay)
- The loss of packets as soon as the virtual output queue is full (the **Maximum Memory** parameter expressed in Kb/s is the virtual output queue size). When the output queue is virtually full, all new incoming packets are not transmitted to the output interface.

The example displayed below illustrates the 3 parameters used by the “Router Simulation & Constant Delay” law: **IP Throughput**, **Maximum Memory** and **Constant Delay**.



The output queue is a virtual queue because there isn’t any real queue associated to the IP Flow.

(continue)

When the IP Flow is started i.e. when the 'Run' button is pressed, the internal remaining size is the Maximum Memory parameter value.
Each time a packet is received, the internal remaining size parameter is decreased by the packet size. When the remaining size parameter is 0, the queue is marked as full.
Any new packet is lost until the remaining size becomes positive. When the packet is sent, the relevant queue size parameter is increased.

In the meantime each packet to send is first moved in the **global output queue** and if needed, the number of packets delayed is increased.

This is why there may be packets not yet send when the IP Flow is stopped. Those packets continue to be sent until the **global output queue** is free.

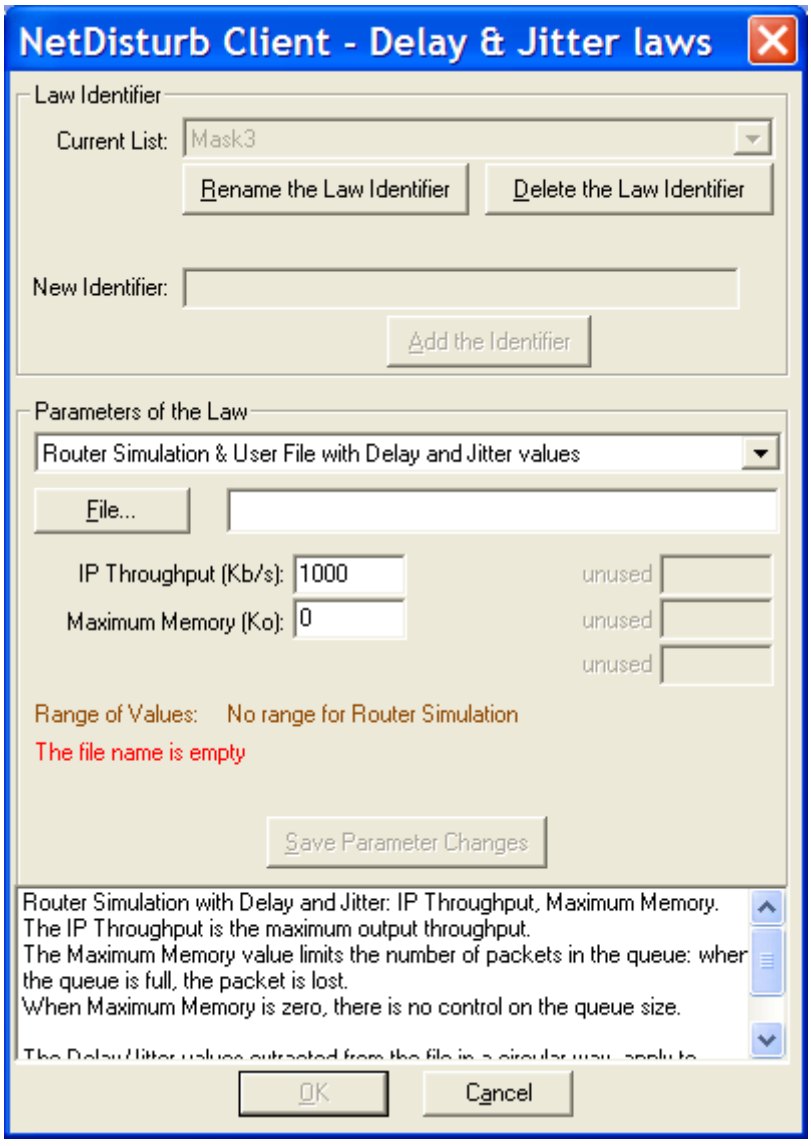
You shouldn't be surprised if packets continue to be sent even if no packet has been received: it is in most cases the **global output queue** that is not yet empty.

7.4.4.10 Router Simulation & User File with Delay and Jitter Values

This law is used to simulate a real network by offering:

- A limited **IP Throughput** on the output interface in Kbps.
- A loss of packets as soon as the output queue is full (the **Maximum Memory** parameter expressed in Kb/s is the output queue size). When the output queue is full, all new incoming packets will not be transmitted to the output interface.
- A Constant Delay & Jitter value read from the text file (to simulate a real network transit delay). The values are expressed with an integer positive number. The unit is the millisecond. The separators used for decoding are: end of line (CR or CR-LF), semicolon, coma, and tab or space characters. One Delay & Jitter value is picked for each packet handled. When the end of the file is reached, the **NetDisturb** driver restarts with the first values of the file. If a line starts with a sharp (#) character, the rest of the line is ignored i.e. up to the End of Line (CR or CR-LF)

The example displayed below illustrates the 3 parameters used by the "Router Simulation & Constant Delay" law: **IP Throughput**, **Maximum Memory** and the user defined **file** containing the **Delay & Jitter values**.



7.4.4.11 Constant Delay & User File with Throughput and Duration Values

This law is used to change the output throughput from time to time. It is a throughput simulation law where the throughput varies.

The throughput and the duration of the throughput are positive and integer values. The values are extracted from the user-defined file. This file must be a text file.
Separators used for decoding are End of Line (CR or CR-LF), semicolon, coma, tab or space characters.

If a line starts with a sharp (#) character, the rest of the line is ignored i.e. up to the End of Line (CR or CR-LF)

There is a couple of values to read:
- The first value is the throughput. The unit of the throughput is the Kbps.
- The second value read is the duration of the throughput. The duration unit is the millisecond.

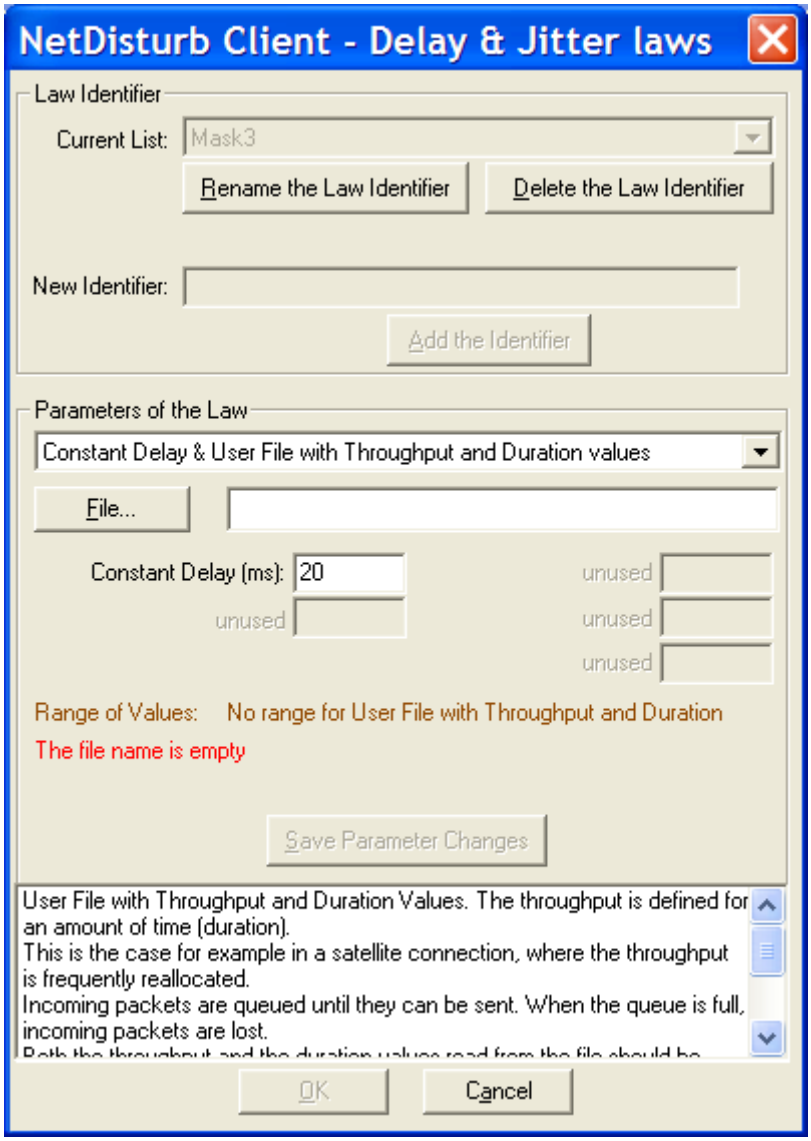
To assure performance, the file is read in one shot and stored in memory at law selection time. The values extracted from the file fill a table transmitted to the **NetDisturb** driver. In order to not overload the memory resources, the maximum read number of values is limited to 40 960 i.e. 20480 couples.

If the file size exceeds the table size, only the part that can be memorized will be read. The rest of the file will be used for the next loading.

If the file size is too small to fulfill the table, the file is read again from the beginning to complete the table.

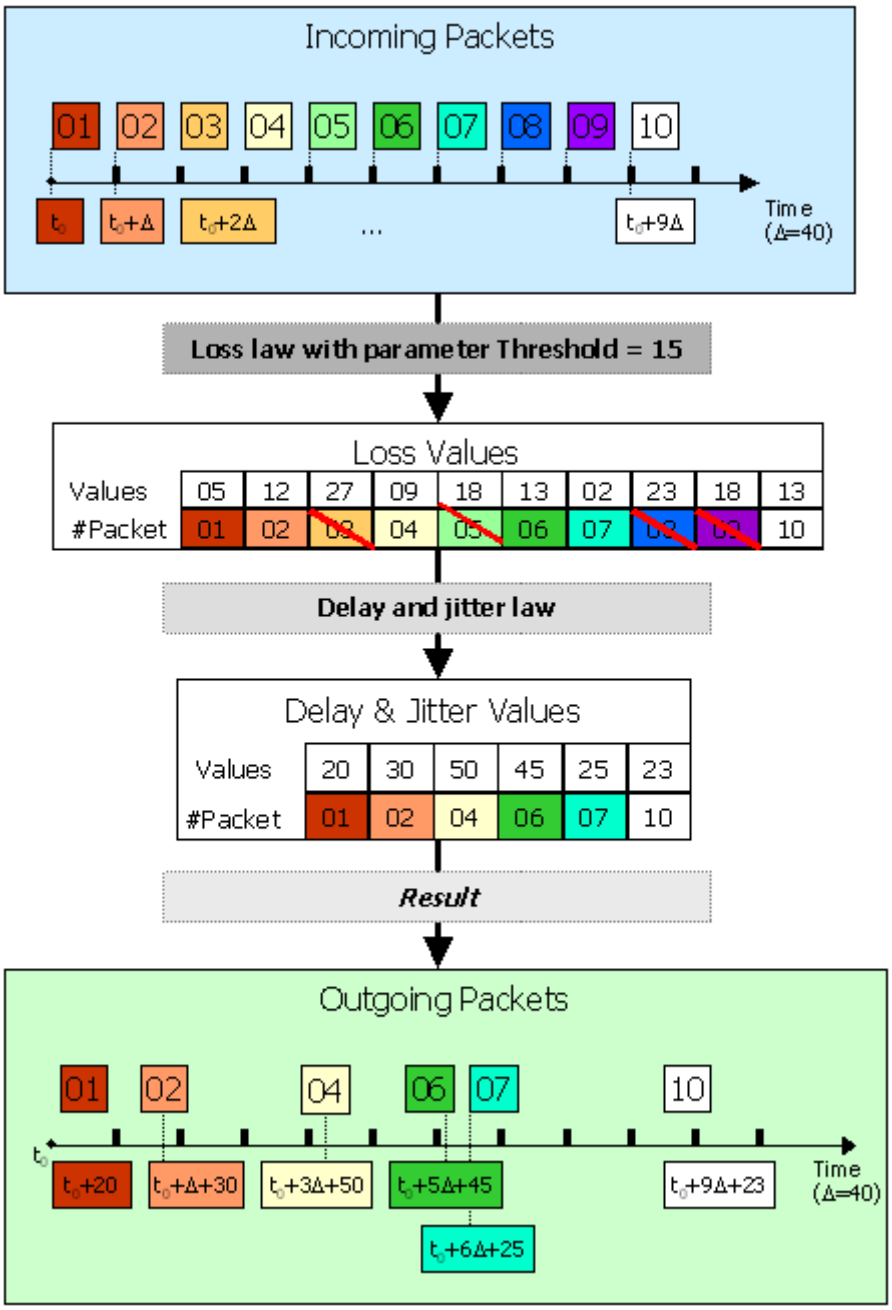
The **NetDisturb** driver extracts a couple of values from the table to get the throughput to apply and its duration. When the duration expires, the next couple of values is extracted from the table, and so on.

A constant delay can be added to each packet, to simulate the network delay, for example the satellite upload or download frame delay.


NetDisturb Client - Delay & Jitter laws
Law Identifier
Current List:
Mask3
Rename the Law Identifier
Delete the Law Identifier
New Identifier:
Add the Identifier
Parameters of the Law
Constant Delay & User File with Throughput and Duration values
File...
Constant Delay (ms):
20
unused
unused
unused
unused
unused
Range of Values:
No range for User File with Throughput and Duration
The file name is empty
Save Parameter Changes
User File with Throughput and Duration Values. The throughput is defined for an amount of time (duration).
This is the case for example in a satellite connection, where the throughput is frequently reallocated.
Incoming packets are queued until they can be sent. When the queue is full, incoming packets are lost.
OK
Cancel

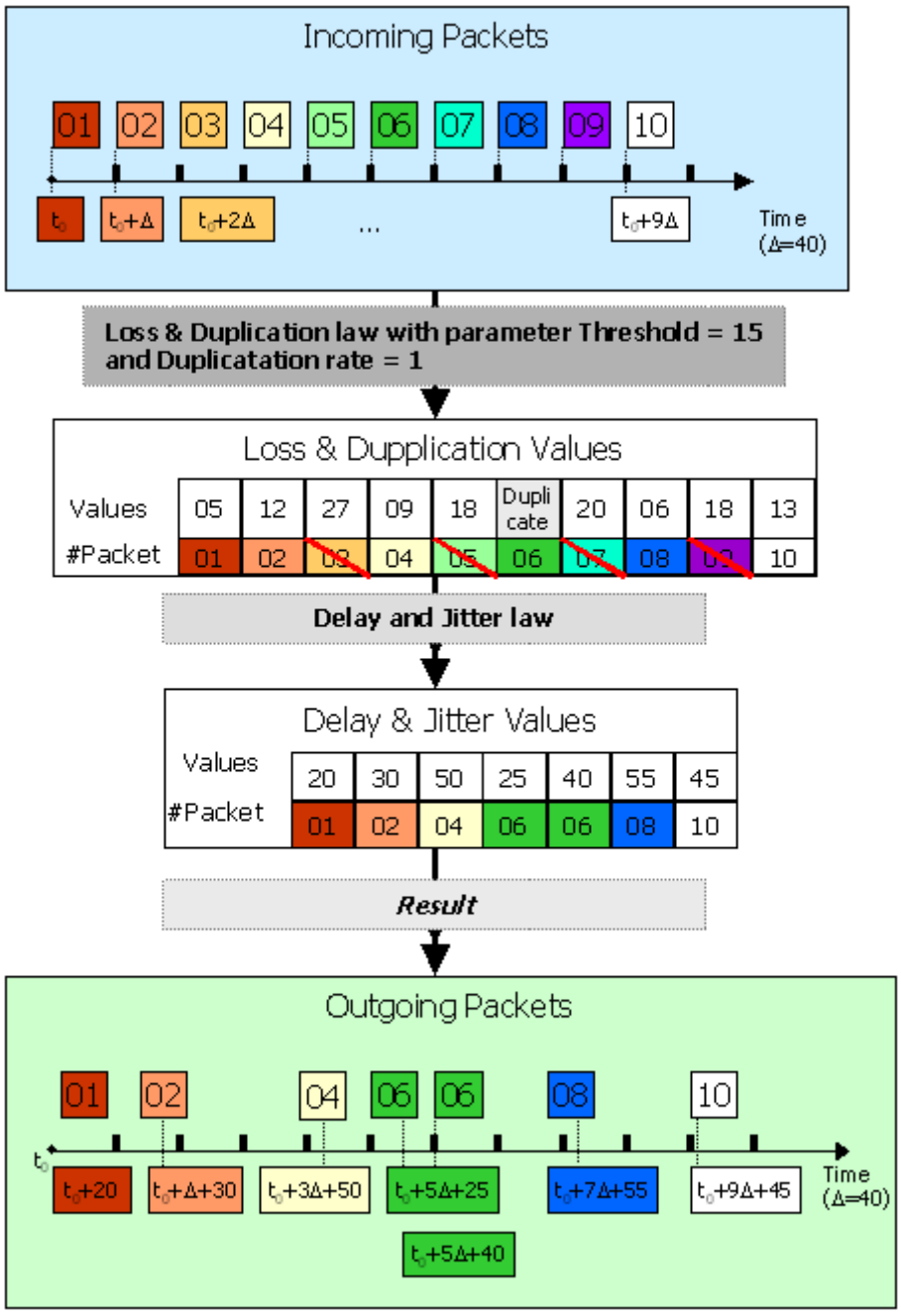
7.4.5 Loss/Duplication and Delay/Jitter Dynamic

The next figure shows the impact of a Loss & Duplication law and a Delay & Jitter law on a set of packets.



7.4.6 Loss with Duplication and Delay/Jitter Dynamic

The next figure shows the impact of a Loss & Duplication law with a Delay & Jitter law on a set of packets.



7.5 Use of the Aggregates

7.5.1 What is an aggregate?

An aggregate is an association of several IP Flows (at least 2) sharing the same Delay & Jitter Laws.
To be defined, the aggregate has to have a Delay & Jitter Law for at least one direction (A➔B and/or B➔A).
The IP Flow order in the aggregate defines the priority of packets to delay. While the top IP Flow packets gets the highest priority, the other IP Flows packets are queuing until there are no higher priority packets.

All the IP Flows related to the aggregate must have their own Mask and possibly a Loss & Duplication Law, but they lose their own Delay & Jitter Law for the benefit of the law defined in the aggregate.
For a given IP Flow belonging to an aggregate, the non-lost packets are subjected to the Delay & Jitter Law of the aggregate.

A priority level applies to packets according to the IP Flow they belong to. The priority is decreasing according to the Flow number, i.e. the packets of the Flow # X get a higher priority than the packets of the Flow # X+1, etc.
All the packets of the Flow # X will be handled before the packets of the Flow # X+1 are taken into account. By waiting to be handled, the packets of the Flow # X+1 are put into a queue. When this queue of a Flow is full, the new packets of this Flow are lost.

All the IP Flows of an aggregate start and stop simultaneously. To start an aggregate, all the IP Flows defined for this aggregate must have a defined mask.

7.5.2 When do we need to use an aggregate?

We use an aggregate when we wish to have different priorities for the various IP Flows to be impaired and when we wish to apply the same Delay & Jitter Law to these IP Flows.

Example of the simulation of a satellite access (IPv4 and IPv6) with a varying time bandwidth and a priority rule for the IP packets

In this example, we define an aggregate with three IP Flows with the following properties, that defines the order of treatment for the received IP packets:

1) The first IP Flow is related to HTTP packets and we associate a Loss Law,
2) The second IP Flow is related to the TCP packets and we associate a Duplication Law,
3) The third IP Flow is related to the UDP packets without applying a Loss & Duplication Law.

So the HTTP packets are first processed with the IP Flow # 01, then the TCP packets are handled with the IP Flow # 02 and the UDP packets are finally processed with the IP Flow # 03.

To implement this example, take the following steps:

- **Step 1:** you must define the Mask and the Loss & Duplication Law for the three IP Flows:
  - IP Flow #01:
    - Mask: Destination Port List = 80 and Source Port List = 80
    - Loss & Duplication Law: select 1 predefined loss law
  - IP Flow #02:
    - Mask: Protocol = TCP
    - Loss & Duplication Law: select 1 predefined duplication law
  - IP Flow #03:
    - Mask: Protocol = UDP
    - No Loss & Duplication Law

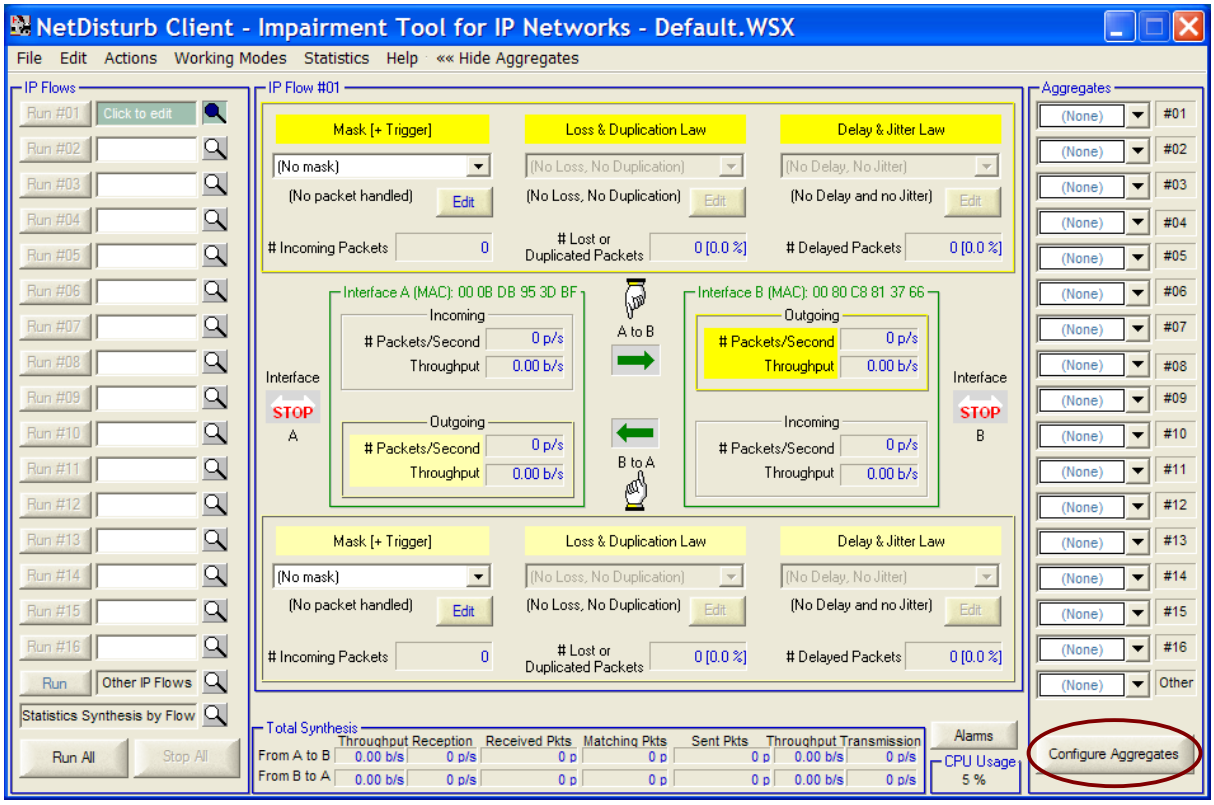
- **Step 2:** you create now an aggregate (blue color for example) with the following Delay & Jitter Law: the 'Constant Delay & User File with Throughput and Duration values' law allowing to simulate the bandwidth variation according to the time (a file containing a couple of integer and positive values <Throughput (in Kbps) | Duration (in ms)> must already exist).

- **Step 3:** you can now apply the blue aggregate to the three IP Flows.

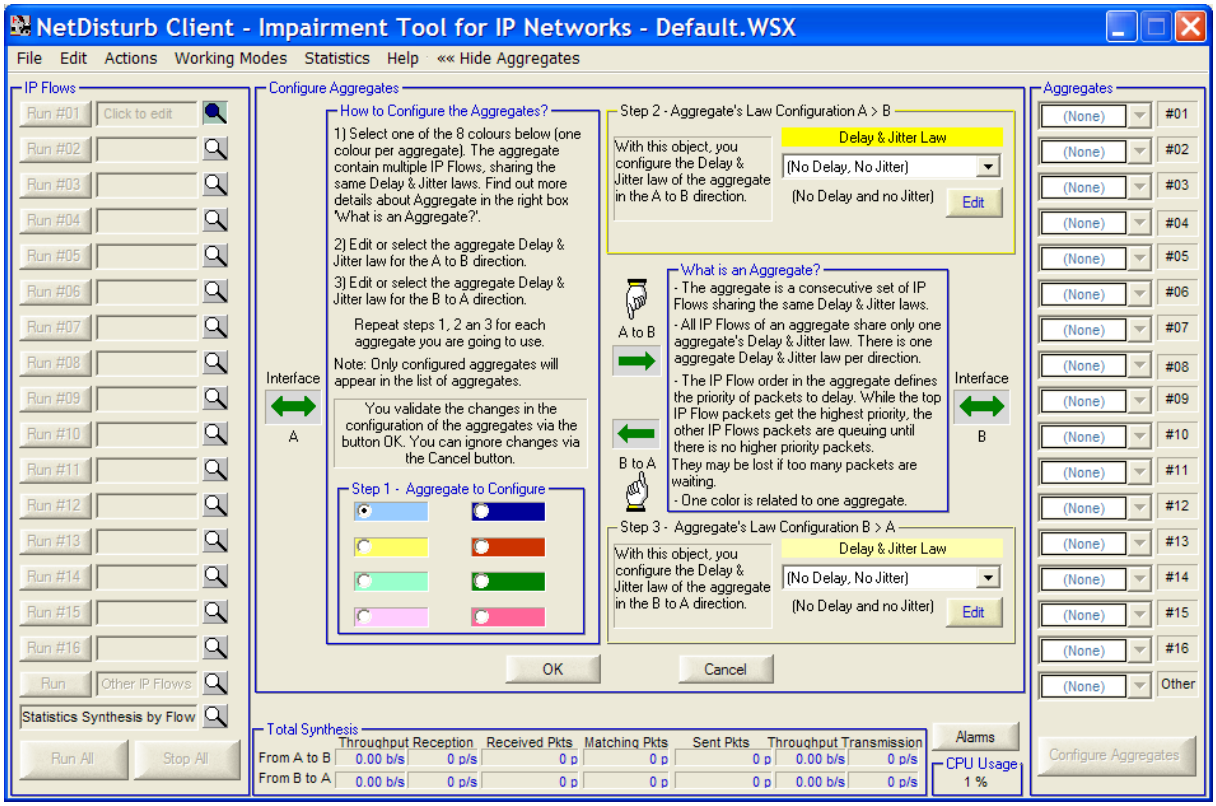
- **Step 4:** Run "IP Flow # 01" to start. When an IP packet is received, **NetDisturb** checks if this packet can be associated to one of the IP Flows of the aggregate, if yes it will apply the Loss & Duplication Law before the Delay & Jitter Law of the aggregate.

7.5.3 How to configure the aggregates

Click the "Show Aggregates >>" menu to display the aggregates section on the right of the window.



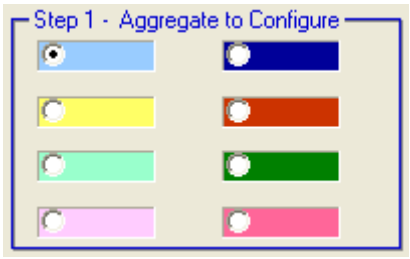
Then press the "Configure Aggregates" button, and the center part of the main window now displays the section to configure the aggregates as shown below:



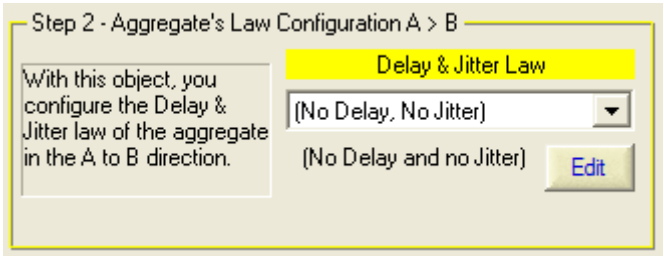
You can define up to 8 aggregates and one aggregate is associated with one color.

Four steps are necessary to parameter an aggregate.
The step 2 and the step 3 are optional, but at least one Delay & Jitter Law should be defined.

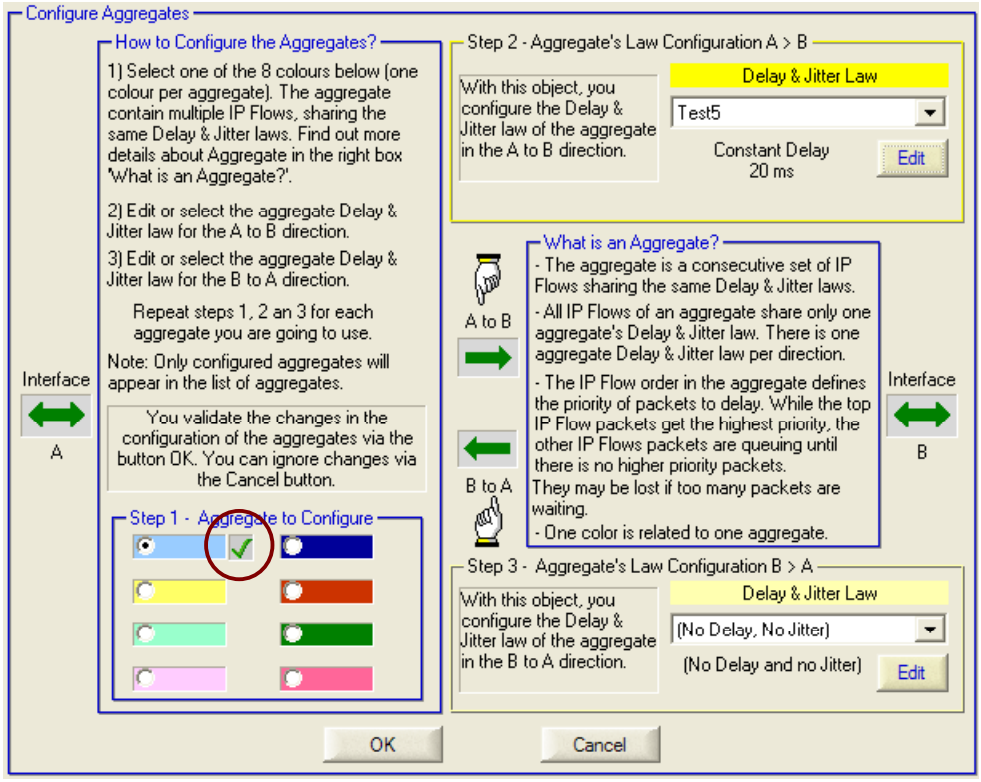
- Step 1: Select first a color among 8 for the aggregate



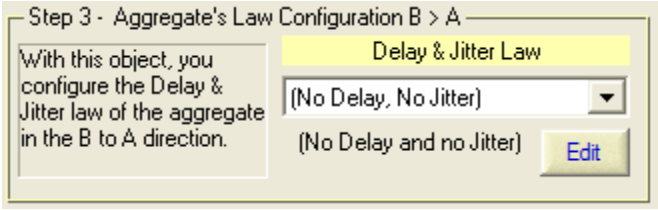
- Step 2 (optional): Select or define a Delay & Jitter Law in the A ➔ B direction



Once a law has been selected or defined, a tick mark is displayed on the right of the color box, as shown below:



- Step 3 (optional): Select or define a Delay & Jitter Law in the B ➔ A direction

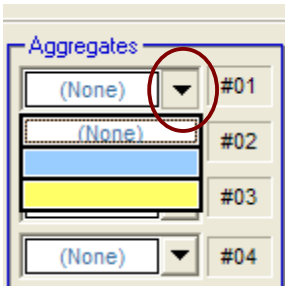


Once the law has been selected or defined, if the tick mark was not already present, it will be displayed on the right of the color box, as shown above.

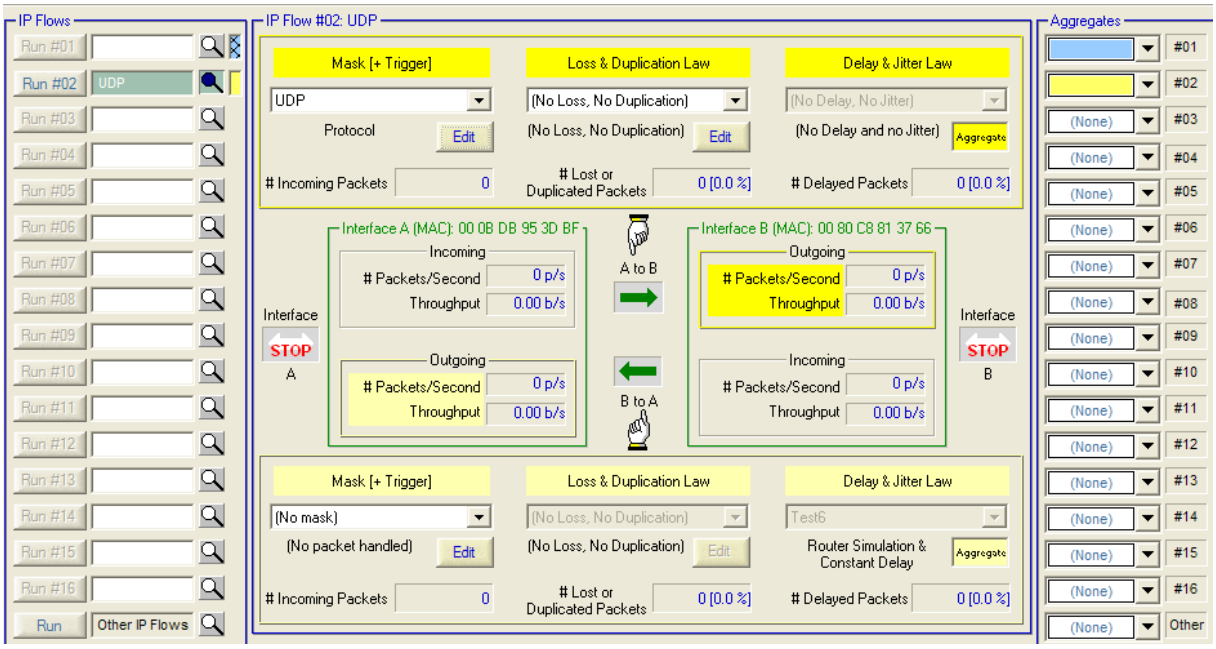
- Step 4: Click OK to save the aggregate

7.5.4 How to associate a colored aggregate to an IP Flow

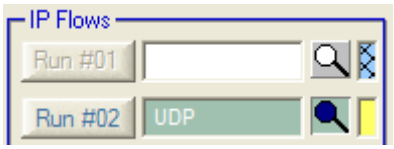
Click the combo-box as shown below - in this example two aggregates have been defined: light Bleu and Yellow. Then select the colored aggregate:



Once the aggregate is selected, a colored mark is displayed on the right of the IP Flow.
For the following example, the light Blue aggregate is associated to the IP Flow # 01, and the yellow aggregate is associated to the IP Flow # 02.



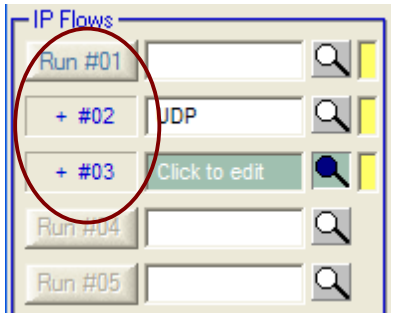
For the following example, the light Blue aggregate is associated to the IP Flow # 01, and the yellow aggregate is related to the IP Flow # 02.



The colored mark located on the right may have two states:

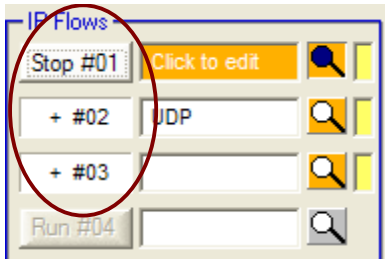
- full color, for example  meaning that a mask is defined for this IP Flow
- or hatched color, for example  meaning that a mask is not defined for this IP Flow and can't be started.

You can associate the same colored aggregate to several IP Flows as in the example below where 3 IP Flows are associated to the yellow aggregate:



Note that the label of the buttons change when an aggregate is associated to several IP Flows (except for the first one): the label of the "Run #02" and "Run #03" buttons change to "+ #02" and "+ #03"

To start the aggregate, press the "Run #xx" button.

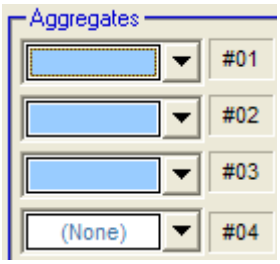


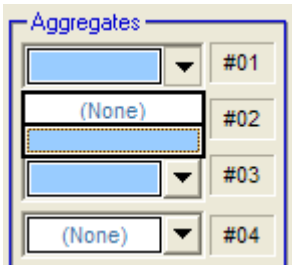
To stop the aggregate, press the "Stop #xx" button.

7.5.5 How to disassociate an IP Flow belonging to a colored aggregate

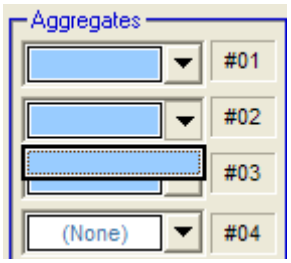
Example:
A light blue light aggregate is associated with three IP Flows (#01, #02 et #03).

You can only dissociate an IP Flow belonging to an aggregate if this IP Flow is the first or the last of the aggregate

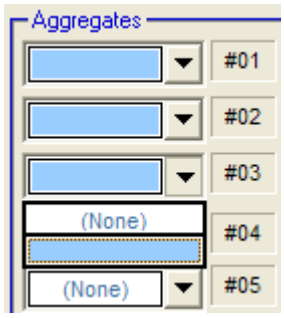




With this configuration you can disassociate the IP Flow #01.



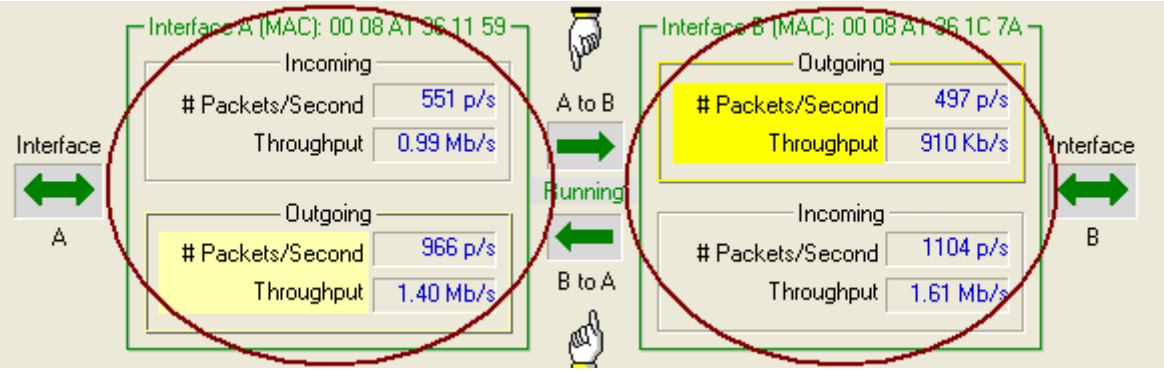
The IP Flow #02 can't be disassociated because the previous IP Flow (#01) and the next IP Flow (#03) are associated to the aggregate.



With this configuration, you can disassociate the IP Flow #03.

7.6 The NetDisturb Client Statistics

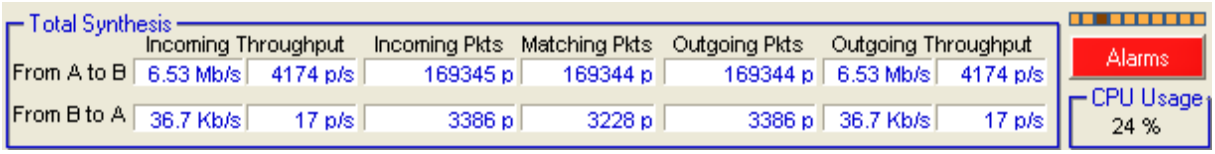
The traffic on the two interfaces is displayed in the central part of the window when an IP Flow is selected, with a section for each interface A and B.
Each section includes one receiving area (incoming) and one sending area (outgoing). The GUI displays the following statistics:



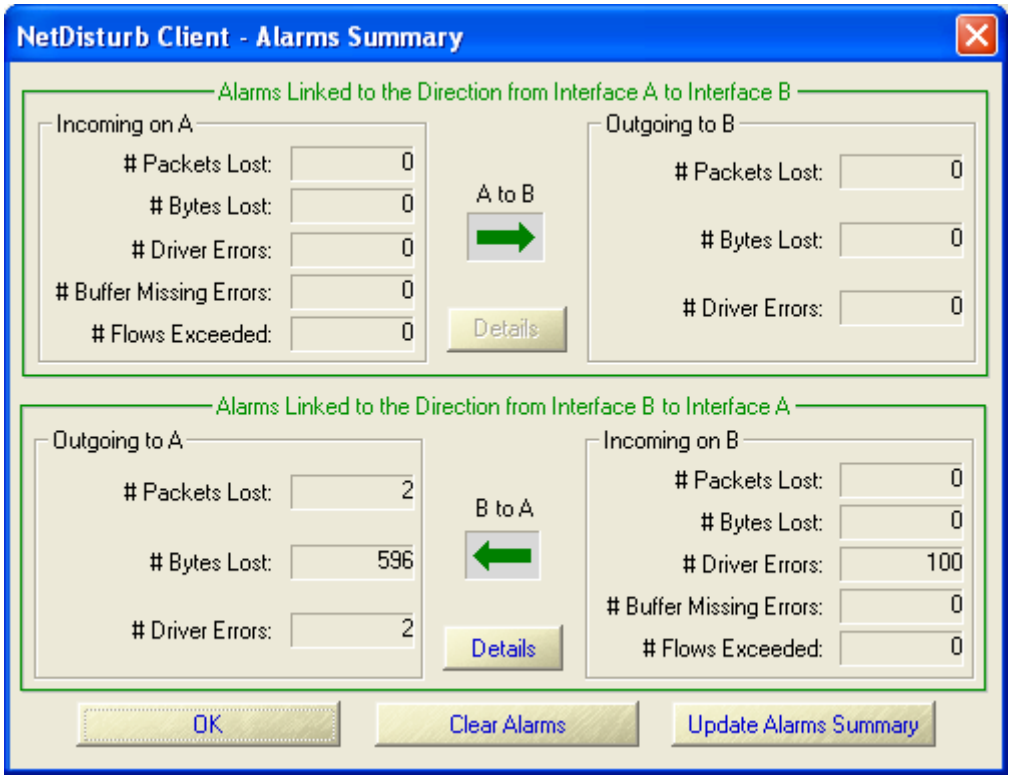
# Packets/Second	This field presents the instant number of packets per second for the IP Flow.
Throughput	This field displays the instant throughput in bits/Kbits/Mbits per second, according to the sampling period defined in the **NetDisturb** Client configuration.

7.7 The Errors Detected by the NetDisturb Driver

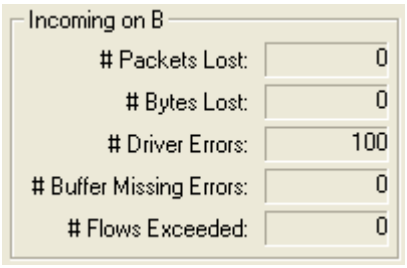
If errors occur at the **NetDisturb** driver level, the 'Alarm' button located in the right bottom of the client area is red colored.



Click on the "Alarms" button to get details about the errors and the following window is displayed:

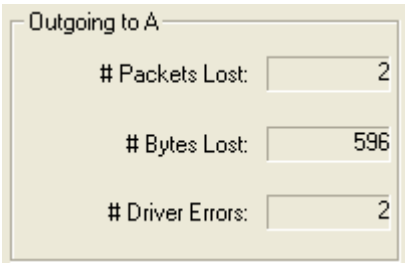


The alarms are classified per direction: **A ➔ B** and **B ➔ A**.
The Information displayed is different depending of the direction (incoming or outgoing).



For the incoming direction:

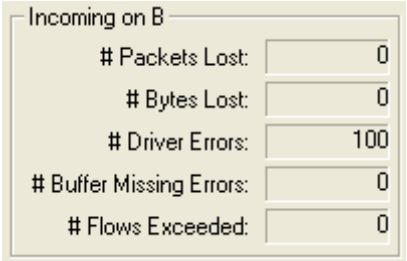
- Number of packets lost
- Number of bytes lost
- Number of errors returned by the driver of the Interface
- Number of buffers that were missing to keep packets
- Number of ignored connections



For the outgoing direction:

- Number of lost packets
- Number of lost bytes
- Number of errors returned by the driver of the Interface

7.7.1 Details for the Incoming Errors



- **# Packets Lost**
  Number of packets lost due to memory allocation errors or interface access errors.

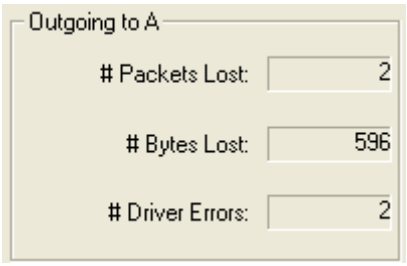
- **# Bytes Lost**
  Number of bytes lost (total packet size including the MAC header) due to memory allocation errors or interface access errors.

- **# Driver Errors**
  This error counter is the number of alarms returned by the NIC driver indicating that some errors have occurred from the started time of the NIC. Errors can be due to one of the following reasons:
    - CRC error
    - NIC or Driver Buffer overrun error

- **# Buffer Missing Errors**
  When a packet is received and memory allocation done by the **NetDisturb** driver failed, this counter is increased. You can increase the number of buffers allocated by the **NetDisturb** driver by changing registry parameters (see paragraph 9.2 to increase the number of buffers)

- **# Flows Exceeded**
  This counter is handled only when the working mode "Laws apply to each IP Flow" is selected. When a packet is received for a new connection but that new connection cannot be added because the maximum number of connections configured has been reached or due to a memory allocation error, this counter is increased for each packet received (see paragraph 9.2 to increase the number of connections).

7.7.2 Details for the Outgoing Errors



- **# Packets Lost**
  Number of packets lost due to memory allocation errors or interface access errors.

- **# Bytes Lost**
  Number of bytes lost (total packet size including the MAC header) due to memory allocation errors or interface access errors.

- **# Driver Errors**
  This error counter is the number of alarms returned by the NIC driver indicating that some errors have occurred from the started time of the NIC. Errors can be due to one of the following reasons:
    - CRC error

- NIC or Driver Buffer overrun error

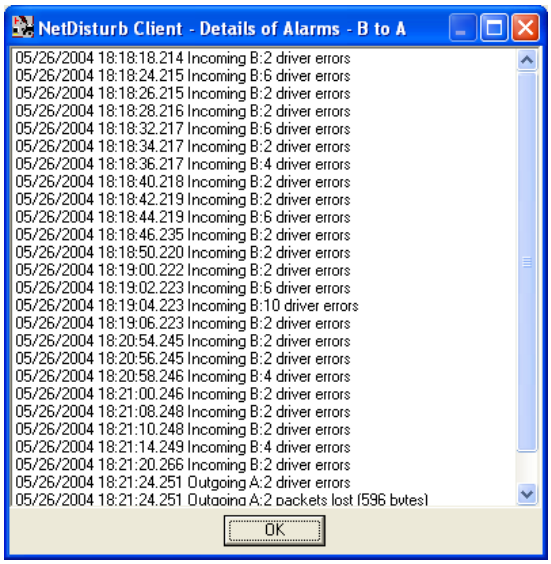
7.7.3 Alarm Management

Four buttons are used to manage these alarms.

▸ The Details button

This button opens a window with details for the alarms:

- Timestamp
- Number of errors
- Error type



▸ The Clear Alarms button

The ‘Clear Alarms’ button resets the alarms list and number for all direction and interfaces.

▸ The Update Alarms Summary button

The ‘Update Alarms Summary’ button interrogates the **NetDisturb** driver to refresh the error list.

▸ The OK button

The OK button closes the Alarms List window and reset the status of the "Alarms" button in the Client Window.

The Alarm Button changes from red 

to gray 

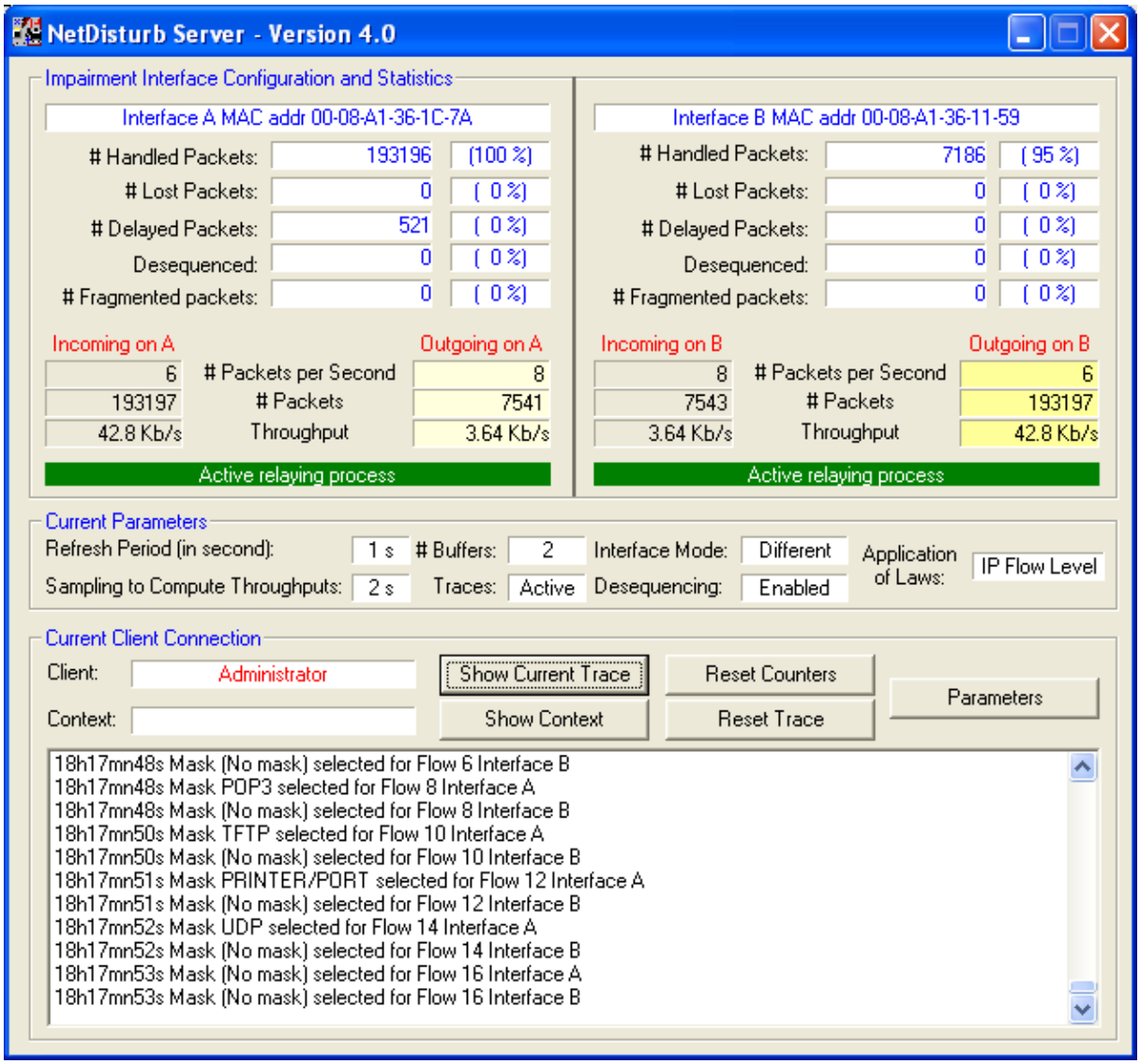
until new errors occur.

PART 8 Using the NetDisturb Server

The **NetDisturb** Server links the **NetDisturb** driver and the **NetDisturb** Client. In addition, it performs the following tasks:

- ⇨ To get a thorough view of the traffic on the two interfaces and on the perturbations made.
- ⇨ To follow the command entered by the connected client, to see the driver configuration, and the applied mask and laws.
- ⇨ To configure the password for Administrator connections.

The **NetDisturb** Server window is composed of three sections:



⇨ Impairment Interface Configuration and Statistics

This section displays the used NICs. Statistics (percentages or absolute values) are associated to each impairment parameter: number of handled, lost, delayed, desequenced and fragmented packets.

The # Fragmented Packets statistics shows the number of packets rejected by the **NetDisturb** driver because it can't handle IP packet with the fragment flag set.

This section displays also the numbers of incoming and outgoing packets, the number of packets per second and the throughput.

The indication on the relaying process is presented as follows:

No packets handled (red color)	The **NetDisturb** driver doesn't handle any packet (physical cut off of the Ethernet link).
Active relaying process (green color)	The **NetDisturb** driver is running, relayed packets are processed following the selected masks and the defined impairment laws.

⇨ Current Parameters

This section reminds the current configuration and includes:

1. The refresh period to display statistics for the **NetDisturb** Server.
2. The sampling period used to calculate the throughput displayed by the **NetDisturb** Server.
3. The number of buffers for laws values related to TCP/UDP connections.
4. The Trace mode: active or inactive
5. The Interface Mode: always different in that version.
6. The desequencing mode: Enabled or Disabled
7. The method to apply the laws:
   - 'IP Flow level' means Laws apply to the IP Flow
   - or 'TCP/UDP connections' means Laws apply to each TCP/UDP connection of the IP Flow

⇨ Current Client Connection

This section shows the currently connected client, the opened context and the trace list.
In this section the following buttons can be pressed:

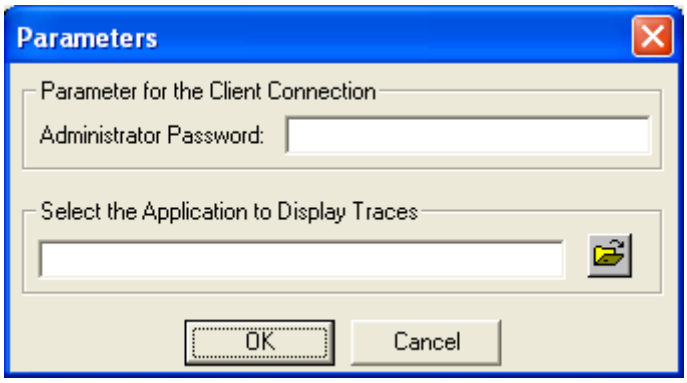
- **Show Current Trace:** allows opening the trace file in order to examine the commands entered by the Client. The program path used to display the traces must be configured in the parameters of **NetDisturb** Server.

- **Reset Counters:** allows the reset of the **NetDisturb** Server Interface counters. This action has no incident for the **NetDisturb** Client. This button is available only when the driver is running.

- **Show Context:** displays the content of the current context.

- **Reset Trace:** allows clearing the traces displayed in the window bottom part. It does not affect the trace file.

- **Parameters:** allows opening of the Parameters window for the **NetDisturb** Server. The available parameters are the administrator password and the viewer for the traces as shown below:



PART 9 Annexes

9.1 The Default Context Values

• Refreshing time for statistics display	1s
• Sampling period for throughput computing	2s
• Relaying process	Relaying packets without operations on both interfaces
• Mode	Internet
• Interface mode	Identical for both interfaces
• Traces	Active
• Driver relaying status	Running
• Buffer number	2
• Flow mode	Mono-flow
• For the 16 definable masks	
Mask	All packets
Loss & Duplication law	*Not defined*
Delay & Jitter law	*Not defined*
• Other IP Flows	
Loss & Duplication law	*Not defined*
Delay & Jitter law	*Not defined*

9.2 The NetDisturb Registry Values

IMPORTANT: this paragraph describes the Registry parameters for the NetDisturb Client, NetDisturb Server and NetDisturb driver.

You should be careful when changing in one of these values because inappropriate value may render NetDisturb unusable. We recommend to backup the Registry before or at least to save the key's values before any change.

You need administrator rights access to change the Registry database. The system 'regedit.exe' program can be used to check and modify the Registry.
Each parameter in the Registry is identified by a name, a type and a value.
The parameters are located into a hierarchical key tree.
This paragraph gives the key location, the parameter name with its type and possible set of value, and default value when applicable.

9.2.1 The Registry parameters related to the NetDisturb Client

This part is related to the **NetDisturb** Client parameters located in the Registry. Some parameters refer to the dialog with the **NetDisturb** Server and should be changed accordingly.

9.2.1.1 Parameters Configuration

Key = HKEY_LOCAL_MACHINE\SOFTWARE\ZTI\NetDisturbClient

Name	Type	Value
AcroReadInfo	REG_SZ	Date of the help file (**the user should not change it)**
AcroReadTimer	REG_DWORD	Internal timeout related to the Adobe Reader®
ExchangeTimeout	REG_DWORD	Internal timeout related to the **NetDisturb** Client to **NetDisturb** Server dialog (default is 5000 ms)
Help_Menu	REG_DWORD	Index in the help file **(the user should not change it)**
IPAddress	REG_SZ	**NetDisturb** Server IP Address (default: 127.0.0.1)
ServerPath	REG_SZ	Full server path to the script sub directory (There is no default value but a typical value is: C:\Program Files**NetDisturb**\Server\Script\)
TCPPort	REG_SZ	RPC port number used to dialog with the **NetDisturb** Server part (default: 2020)
TraceLevel	REG_DWORD	Trace level generated by the Client (see note) (default: 0)
TraceFileName	REG_SZ	File name where traces are saved in when the TraceLevel flag is saved. (default: empty)
UserName	REG_SZ	Latest user name

Note:

- The level of trace is a set of flags. When the flag is set (1) the level is active. When the flag is reset (0) the level is inactive. Ex: TraceLevel=5, the flag 4 and 1 are active, other are inactive.
- Traces are displayed to the standard debug port.
- Flag values are shown in **hexadecimal**:
  - `0001` Errors level
  - `0002` Information level
  - `0008` Verbose level
  - `0010` Time: add timestamp information
  - `0100` File: trace are saved into a file too (the TraceFileName entry is used)
  - `1000` RPC: add the RPC trace information
  - Example:
    If TraceLevel = 113 means Error and Information level of traces are saved also into a file and including the timestamp for each trace.

9.2.1.2 The Most Recent File list

This list is for information only.

It is handled by the system and you must not change it.

Key = HKEY_CURRENT_USER\Software\ZTI\NetDisturbClient\Recent File List

Name	Type	Value
File1	REG_SZ	The most recent path context file used
File2	REG_SZ	A more recent path context file used
File3	REG_SZ	A more recent path context file used
File4	REG_SZ	The oldest path context file used

9.2.2 The Registry parameters related to the NetDisturb Server

Key = HKEY_LOCAL_MACHINE\SOFTWARE\ZTI\NetDisturbServer

Name	Type	Value
ApplicationName	REG_SZ	Trace viewer

IHMRefresh	REG_DWORD	Period of refresh, in second. (default is 1)
Interface A	REG_SZ	MAC address of the latest selected Interface A
Interface B	REG_SZ	MAC address of the latest selected Interface B
Password	REG_SZ	Password required for the 'Administrator' user (default: empty)
Sampling	REG_DWORD	Sampling period to compute throughput (default: 2)
TCPPort	REG_SZ	RPC port number used to dialog with the Client part (default: 2020)
TraceLevel	REG_DWORD	Trace level generated by **NetDisturb** Server (see note) (default: 0)

Note:

- The level of trace is a set of flags. When the flag is set (1) the level is active. When the flag is reset (0) the level is inactive. Ex: TraceLevel=5, the flag 4 and 1 are active, other are inactive.
- Traces are displayed via the standard debug mechanism (for trace display program such as dbmon or DebugMon).
- Flag values are shown in **hexadecimal**:
  - `0001` Error level
  - `0002` Important level
  - `0008` Information level
  - `0100` Verbose level (1)
  - `0200` Verbose level (2)
  - `1000` Put trace generated into the **NetDisturb** Server trace window
  - Example:
    If TraceLevel = 1001 means Error level of traces shown into the window trace.

9.2.3 The Registry parameters related to the NetDisturb driver

Key = HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetDisturb

Key (Windows NT only) **=** HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Disturb

Name	Type	Value
DisplayName	REG_SZ	Name of the service (Default is '"NetDisturb Impairment")
ErrorControl	REG_DWORD	1
ImagePath	REG_SZ	system32\drivers\disturb.sys
Start	REG_DWORD	3
Type	REG_DWORD	1

9.2.4 The NetDisturb Driver Traces

There is another key related to the level of traces generated by the **NetDisturb** driver. These traces can be captured via a tool such as DebugMon of OSR Inc. (www.osronline.com selection Download).

IMPORTANT: Changing the level of the traces may block your PC until you reboot. The level of the traces provided by the NetDisturb driver should be modified only with the help of the technical ZTI support (support@zti-telecom.com).

Key = HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetDisturb\Parameters

TraceLevel	REG_DWORD	Trace level generated by the **NetDisturb** Driver (see note) (default: 0)
Note: the level of trace is a set of flags. Values aren't provided here to avoid mishandling of the **NetDisturb** driver. Please contact ZTI technical support if you need more details.		

9.2.5 The Windows Registry (Windows XP)

The goal of this modification of the Windows system parameters is to enable the RPC service that is required by the **NetDisturb** Server and Client to dialog.

Key = HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows NT\RPC

Name	Type	Value
RestrictRemoteClients	REG_DWORD	0x00000000

9.3 The Mathematical Laws used by NetDisturb

9.3.1 The Uniform Law

⇨ *Presentation*

The Uniform law has two parameters: α and β. It generates a random number included uniformly between α and β. If α is equal to β, the generated number is always α = β.

⇨ *Mathematical function*

Uniform law on (α, β) range

$f(x) = 1/(\beta - \alpha)$ if $\alpha < x < \beta$

$f(x) = 0$ else

⇨ *Uniform law - example of generated values for1000000 draws for this law with* $\alpha = 0$ *and* $\beta = 100$

The factor 1000000 is because the figure intents to show the actual behavior of the random generator. To do that, we draw 1000000 times a random value and count the actual frequencies. The theoretical (= calculated values) curve and actual (= generated values) curve are displayed below.

9.3.2 The Uniform Correlated Law

The Uniform Correlated law is the same law as Uniform law. Only the process differs: the difference is related to the two thresholds used by the **NetDisturb** driver (see the “Loss laws configuration” paragraph for more details).

9.3.3 The Exponential Law

⇨ *Presentation*

The Exponential law has only one parameter: λ.
The more λ is small, the more the power of 10 for the generated number is high.

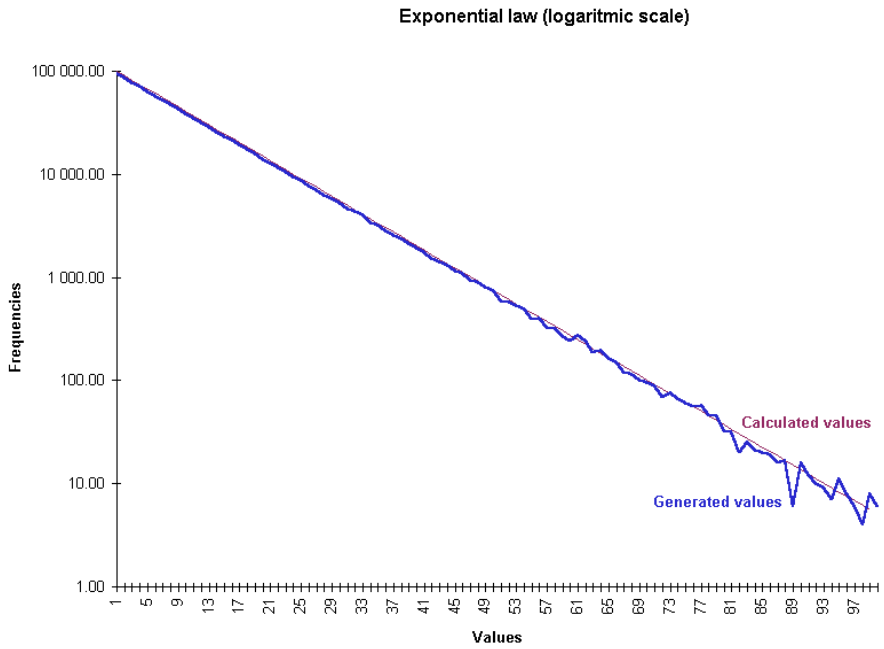
⇨ *Mathematical function*

Exponential law $(\lambda > 0)$

$$f(x) = \lambda e^{-\lambda x} \quad \text{if} \quad x \geq 0$$
$$f(x) = 0 \quad \text{if} \quad x < 0$$

⇨ *Exponential law - example of generated values for 1000000 draws with* $\lambda = 0{,}1$

The factor 1000000 is because the figure intents to show the actual behavior of the random generator (not to show the theory of the exponential law). To do that, we draw 1000000 times a random value and count the actual frequencies. The theoretical (=calculated values) and actual (=generated values) curves match perfectly for bigger values.



⇨ *Exponential law: Table of generated values*

Value	Result of the law
λ = 1	10 ms

λ = 0,1	100 ms
λ = 0,01	1s
λ = 0,001	10s
λ = 0,0001	1mn43
λ = 0,00001	17mn19
λ = 0,000001	2h53
-- Precision limit of λ --	