

# **INAV USER DOCUMENTATION v 0.1**

**Nathan Robinson**  
**Jeff Scaparra**

## **TOC**

### **Server and Server Tools**

- **Hardware Requirements**
- **Installation**
  - **Software Requirements**
  - **Downloading the tar.gz from inav.scaparra.com**
  - **via the subversion server**
- **Deployment scenarios**
  - **Raw Data Capture - monitor port on switch**
  - **Raw Data Capture - via network tap**
  - **Local Computer**
  - **Pcap Files**
  - **Comma Delimited Files**
  - **SFlow (Future Feature)**
  - **Netflow (Future Feature)**
- **Running on a non-standard port**
- **Testing with tnav**
  - **Changing the server configuration on the fly**
  - **Viewing all edges in the graph**
  - **Viewing nodes as they die**
- **Trouble Shooting**

### **Client**

- **Hardware Requirements**
- **Installation**
  - **Software Requirements**
  - **Downloading the jar file from inav.scaparra.com**
  - **via the subversion server**
- **UI**
  - **Changing the server and port that the client connects to**
  - **Changing the bandwidth amounts and colors**
  - **Edgelife and graph refresh**
  - **Navigating the graph**
  - **Node Data**
  - **Information and limitation of the physics engine**
- **Trouble Shooting**

# Interactive Network Active-Traffic Visualization (INAV)

## ***Preface***

INAV began as a class project in the spring of 2007 and has been continued to be developed and will continue to be developed. Originally INAV was developed for visualization of traffic in real time as a response to the need to see connection information and understand the results quickly. Other tools that can be used to analyze traffic in real time are etherApe, wireshark, and tcpdump (to name a few of the more popular). The goal behind creating a new tool was to develop something that would be able to process massive amount of data and allow the user to visually make conclusions much faster than sorting through a text file like with wireshark or tcpdump. EtherApe also has a number of limitations especially when there is port scanning and the network being monitored is large.

## **INAV SERVER**

### ***Hardware Requirements***

As with any software, the better the hardware that it runs on the better the application will run. That said the INAV server can be used even in a production environment with relatively cheap hardware.

The server currently running in our testbed that is processing the data for the entire computer science department at a large university is (3000+ connections):

- 1.4 Ghz PIII
- 512 KB Cache
- 1 Gbps Fiber card
- 512 MB RAM
- Gentoo Linux 2.6

I would suggest this to be the minimum recommended hardware for any enterprise application of INAV.

## **Installation**

### **PREREQUISITES**

libpcap0.8 - can be installed on ubuntu via `$ sudo aptitude install libpcap0.8-dev`

g++ - can be installed on ubuntu via `$ sudo aptitude install g++`

### **Tar files from [inav.scaparra.com](http://inav.scaparra.com)**

INAV can be downloaded from <http://inav.scaparra.com/download/server>.

Download the tarball to the directory where you would like inav to reside:

```
scap@venus:~/tmp$ wget http://inav.scaparra.com/files/server/INAV-Server-0.3.2.tar.gz
--14:51:30-- http://inav.scaparra.com/files/server/INAV-Server-0.3.2.tar.gz
=> `INAV-Server-0.3.2.tar.gz'
Resolving inav.scaparra.com... 66.33.221.126
Connecting to inav.scaparra.com|66.33.221.126|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 191,248 (187K) [application/x-tar]

100%[=====>] 191,248      5.15K/s  ETA
00:00

14:52:05 (5.35 KB/s) - `INAV-Server-0.3.2.tar.gz' saved [191248/191248]
```

Unpacking the tarball:

```
scap@venus:~/tmp$ tar zxvf INAV-Server-0.3.2.tar.gz
server/
server/packet.h
server/sniffer.h
server/makefile
...
```

Compiling the server:

```
scap@venus:~/tmp$ cd server/
scap@venus:~/tmp/server$ make
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o clientComm.o clientComm.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o clientCommData.o clientCommData.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o baseData.o baseData.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o snifferData.o snifferData.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o sniffer.o sniffer.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o ethernet.o ethernet.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o ip.o ip.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o tcp.o tcp.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o packet.o packet.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o filterData.o filterData.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o graphData.o graphData.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o bandwidthMonitor.o bandwidthMonitor.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o semaphore.o semaphore.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o icmp.o icmp.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o traceroute/tracerouteData.o
traceroute/tracerouteData.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o traceroute/tracerouteThread.o
traceroute/tracerouteThread.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o commandLineParser.o
commandLineParser.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o xmlParser.o xmlParser.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o helper.o helper.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o udp.o udp.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o parseCommas.o parseCommas.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o debugThread.o debugThread.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o inavServer.o inavServer.cpp
g++ -lpthread -lpcap -o inavd clientComm.o clientCommData.o baseData.o snifferData.o
sniffer.o ethernet.o ip.o tcp.o packet.o filterData.o graphData.o bandwidthMonitor.o semaphore.o
icmp.o traceroute/tracerouteData.o traceroute/tracerouteThread.o commandLineParser.o
xmlParser.o helper.o udp.o parseCommas.o debugThread.o inavServer.o
```

```
scap@venus:~/tmp/server$
```

Congratulations the server have been installed and can be run by calling `./inavd` in that folder.

## Installing from subversion

**Warning:** Checking out `inav` from subversion will ensure that you have the most up to date code however there is no guarantee that it has undergone ANY testing for bugs etc... It may not compile or may not work right. If you have problems if the this method please revert to the normal installation methods.

Checking out the code:

```
scap@venus:~/tmp$ svn co http://inav.scaparra.com/INAV/server
A  server/packetData.h
A  server/tester.cpp
A  server/commandLineParser.h
A  server/baseData.h
...
```

Compiling the code:

```
scap@venus:~/tmp$ cd server/
scap@venus:~/tmp/server$ make
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o clientComm.o clientComm.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o clientCommData.o clientCommData.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o baseData.o baseData.cpp
g++ -ggdb -g3 -D INAV_VERSION=\"0.3.2\" -c -o snifferData.o snifferData.cpp
...
```

Congratulations the server have been installed and can be run by calling `./inavd` in that folder.

## Deployment scenarios

### Raw Packet Capture

There are two forms of raw packet capture sniffing from a network tap and sniffing from a monitor port on a switch. Each has its own pros and cons and it will depend on your network as to which is best to suit your needs.

#### Raw Packet Capture via monitor port

Pros:

- Can see all the traffic that the switch can see
- Most managed switches can provide this data
- Can use all of the packet data for filtering

Cons:

- Not all packets can be ensured that they will be captured. If the switch is processing more data than can traverse the monitor port the excess data is dropped.

This is the original capture deployment mode for INAV. In this mode all data about the packets are captured as long as the data traversing the switch is not greater than the amount that can be sent out of the monitor port. For this reason if the switch has different speed ports, the monitor port should be on the fastest interface on the switch. This mode is preferred over other methods when the user would like to visualize local traffic as well as traffic traversing the Internet.

### **Raw Packet Capture via network tap**

Pros:

- Can see all traffic on a particular link that is being "tapped"
- Easy to install
- As long as the mechanism used to read the packets is as fast as the link it will be able to capture all packets. (Limited by the pcap library)

Cons:

- Requires extra network gear
- Can't see any traffic that isn't traversing the link.

This uses the same interface capture method as raw packet capture via a monitor port. The difference is the device and interface that the capture port is connected to. In this set up a tap is placed between the internal LAN and the external Internet. The downside to this method is that local traffic that doesn't leave the LAN can not be seen and is therefore not processed by the visualization.

### **PCAP Files**

Pros:

- Can be replayed and reanalyzed over and over
- easy to produce elsewhere for playback at a different location(s) at a later date.

Cons:

- Not real time (not always a bad thing)

### **CSV Files ( Comma delimited files )**

Pros:

- Can take any data that could be outputted in this form.
- Easy to produce

Cons:

- Not real time (not always a bad thing)

### **Netflow**

Pros:

- Less overhead than a monitor port
- Can see all the traffic on the device
- Multiple devices can send netflow data to the server

Cons:

- Vendor Specific (not all hardware is capable)
- Doesn't send all the packet information (some forms of filtering will be impossible)

## **SFlow**

Pros:

- Less overhead than a monitor port
- Can see all the traffic on the device
- Multiple devices can send netflow data to the server

Cons:

- Vendor Specific (not all hardware is capable)
- Doesn't send all the packet information (some forms of filtering will be impossible)

# INAV CLIENT

## Hardware Requirements

As with any software, the better the hardware that it runs on the better the application will run. That said the INAV client can be used even in a production environment with relatively cheap hardware.

The client used to visualize the entire computer science department at a large university is:

- 3200+ X2 AMD
- 512 KB Cache
- 10 Mb/s link
- 2 GB RAM
- Java 1.5 (5.0)
- Windows XP SP2, Ubuntu Linux 7.04, Gentoo Linux 2.6, OS 10.4

The client is designed on a framework that allows it to run on any system. I would suggest this to be the minimum recommended hardware and software for any enterprise application of INAV.

## Installation

### PREREQUISITES

The INAV client is designed to run on any architecture and any OS. We have experienced problems, and as such have developed solutions or workarounds so that our goal of complete system compatibility can be maintained.

At a minimum, you will need the JAVA runtime environment installed. On OS X, this is already installed (Java 1.5) and for other operating systems, you will need to install manually. Java can be installed from [http://java.sun.com/javase/downloads/index\\_jdk5.jsp](http://java.sun.com/javase/downloads/index_jdk5.jsp). You will want to install the Java Runtime Environment (JRE) 5.0 Update XX (where XX is the largest number on the page).

### JAR file from the server

Once you have Java installed, you will be able to double click on the INAV.jar file to run it, or in some cases - right click and select "Open with Java Platform..."

### Installing from subversion

**Warning:** Checking out inav from subversion will ensure that you have the most up to date code however there is no guarantee that it has undergone ANY testing for bugs etc... It may not compile or may not work right. If you have problems if the this method please revert to the normal installation methods.

Checking out the code:

```
scap@venus:~/tmp$ svn co http://inav.scaparra.com/INAV/display  
...
```

Compiling the code:

```
scap@venus:~/tmp$ cd dispaly/  
scap@venus:~/tmp/display$ do stuff
```

...

Congratulations the display have been installed and can be run by calling java stuff THING DOODAD in that folder.

Alternatively, you can save the client (precompiled) at <http://inav.scaparra.com/INAV/display/INAV.jar>.