

# **μProbes for Monitoring and Troubleshooting**



# Motivation

Create a low-cost sensor for remote troubleshooting and remote monitoring:

- Many companies/people have remote sites with limited traffic.
- Need to supervise activities from central location and remote-connect and troubleshoot when necessary.
- (Optional) Stream monitoring data to the central location
- Low-cost both in terms of
  - Price: monitoring cannot be too expensive otherwise people won't use it.
  - Usage: small sites often don't have local IT people.



# Solution Constraints

- Remote, live monitoring (ntopng/wireshark...) is required.
- Packet storage is likely but not compulsory.
- Mandatory: CE/FCC Certified (no toys or hacking tools) as we cannot break a network when using non-certified tools.
- The box must be transparent to the traffic:
  - If it breaks and people are unable to fix it immediately it can be disconnected (until a fix is found) and the network will continue to operate without any change in IP address assignment.
  - It must be cheap so keeping a spare unit should not be a problem.

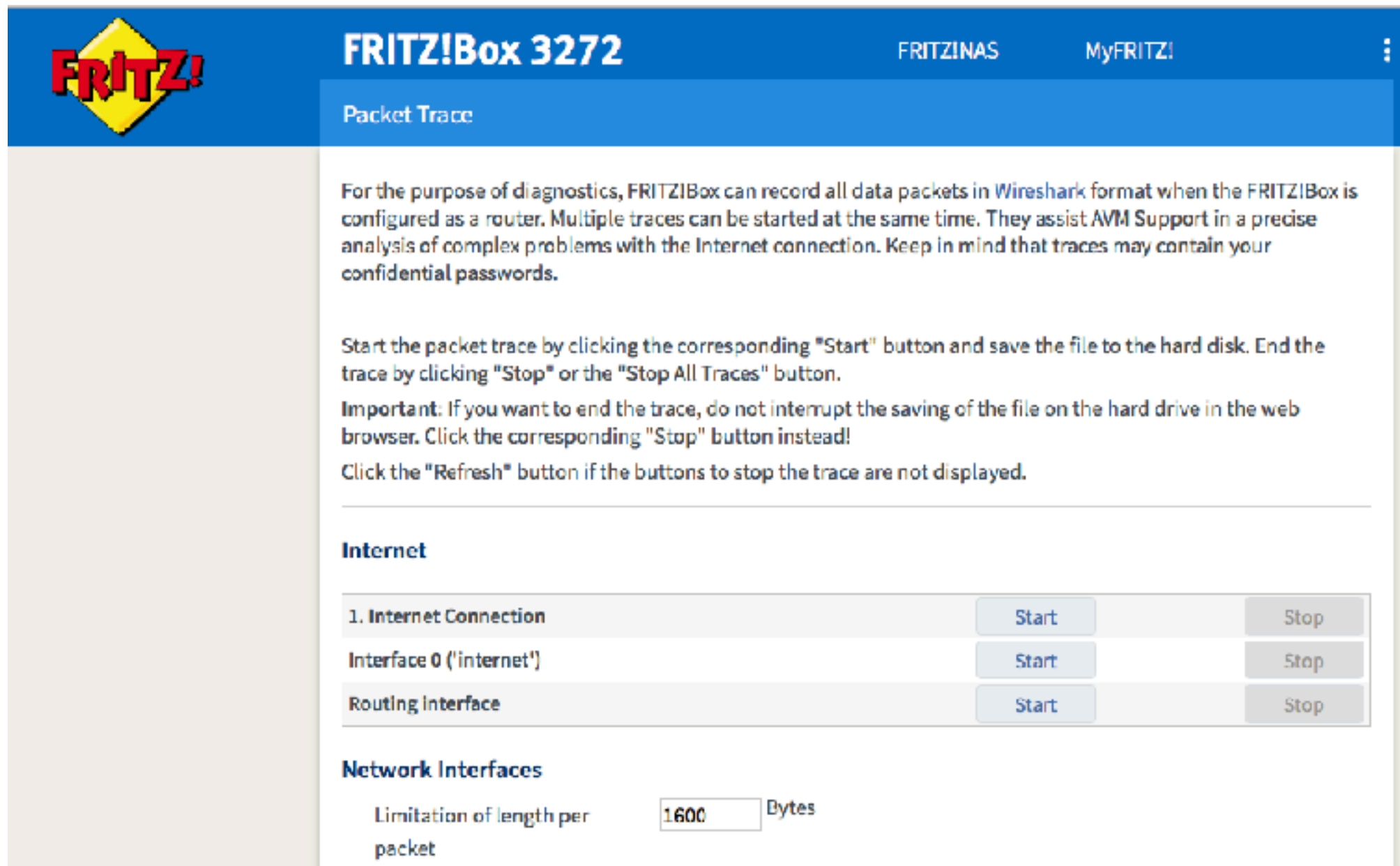


# Part 1: General Purpose Hardware



# FrizBox! [1/3]

- Some xDSL routers have the (hidden) ability to forward traffic for analysis.



The screenshot shows the FRITZ!Box 3272 web interface. The top navigation bar includes the FRITZ! logo, the model name 'FRITZ!Box 3272', and links for 'FRITZINAS' and 'MyFRITZ!'. The main section is titled 'Packet Trace' and contains a detailed explanation of the feature, instructions on how to start and stop traces, and a table of available interfaces for tracing.

**FRITZ!Box 3272** FRITZINAS MyFRITZ!

### Packet Trace

For the purpose of diagnostics, FRITZ!Box can record all data packets in [Wireshark](#) format when the FRITZ!Box is configured as a router. Multiple traces can be started at the same time. They assist AVM Support in a precise analysis of complex problems with the Internet connection. Keep in mind that traces may contain your confidential passwords.

Start the packet trace by clicking the corresponding "Start" button and save the file to the hard disk. End the trace by clicking "Stop" or the "Stop All Traces" button.

**Important:** If you want to end the trace, do not interrupt the saving of the file on the hard drive in the web browser. Click the corresponding "Stop" button instead!

Click the "Refresh" button if the buttons to stop the trace are not displayed.

#### Internet

1. Internet Connection	Start	Stop
Interface 0 ('internet')	Start	Stop
Routing Interface	Start	Stop

#### Network Interfaces

Limitation of length per packet  Bytes



X



# FrizBox! [2/3]

```
#!/bin/bash

FRITZIP=192.168.2.1
FRITZPWD=$1
#IFACE="2-0"
IFACE="1-lan"

FRITZUSER=""
SIDFILE="/tmp/fritz.sid"

if [ ! -f $SIDFILE ]; then
    touch $SIDFILE
fi

SID=$(cat $SIDFILE)

CHALLENGE=$(curl -s http://$FRITZIP/login_sid.lua | grep -o "<Challenge>[a-z0-9]\{8\}" | cut -d'>' -f 2)
HASH=$(perl -MPOSIX -e '
    use Digest::MD5 "md5_hex";
    my $ch_Pw = "$ARGV[0]-$ARGV[1]";
    $ch_Pw =~ s/(.)/$1 . chr(0)/eg;
    my $md5 = lc(md5_hex($ch_Pw));
    print $md5;
' -- "$CHALLENGE" "$FRITZPWD")
curl -s "http://$FRITZIP/login_sid.lua" -d "response=$CHALLENGE-$HASH" -d 'username='${FRITZUSER} | grep -o
"<SID>[a-z0-9]\{16\}" | cut -d'>' -f 2 > $SIDFILE

SID=$(cat $SIDFILE)

echo "Capturing traffic.." 1>&2

wget -q0- http://$FRITZIP/cgi-bin/capture_notimeout?ifaceorminor=$IFACE\&snaplen=\&capture=Start\&sid=$SID | /usr/
local/bin/tshark -r - -w /tmp/fritz.pcap

#cd /Users/deri/network/ntopng
#wget -q0- http://$FRITZIP/cgi-bin/capture_notimeout?ifaceorminor=$IFACE\&snaplen=\&capture=Start\&sid=$SID | ./ntopng
-i -
```



## Some comments

1. You can monitor traffic both on wired and wireless links
2. Traffic is streamed locally (from remote this is forbidden by the internal firewall) from the router over HTTP.
3. You still need a local PC able to start the HTTP streaming and receive packets. In essence this solution is an “alternative/embedded” network tap but not a monitoring tools as it lacks the monitoring capabilities.
4. Using this solution you can create a cheap yet passive solution for remote monitoring without changing the network configuration or buy extra devices.



# Ubiquiti EdgeRouter-X [1/7]

- Ubiquiti EdgeRouter-X (price < 60 Euro/< 75 \$)
- MIPS (32 bit) based architecture with ~128MB free RAM
- Running Vyatta OS (Debian).



X





# Ubiquiti EdgeRouter-X [2/7]

```
root@ubnt:/home# cat /proc/cpuinfo
system type      : MT7621
machine          : Unknown
processor        : 0
cpu model       : MIPS 1004Kc V2.15
BogoMIPS        : 583.68
wait instruction : yes
microsecond timers: yes
tlb_entries     : 32
extra interrupt vector : yes
hardware watchpoint : yes, count: 4, address/irw mask: [0x0ffc, 0x0ffc, 0x0ffb, 0x0ffb]
isa             : mips1 mips2 mips32r1 mips32r2
ASEs implemented : mips16 dsp mt
shadow register sets : 1
kscratch registers: 0
core            : 0
VPE             : 0
VCED exceptions  : not available
VCEI exceptions  : not available
```



X



# Ubiquiti EdgeRouter-X [3/7]

EdgeMAX<sup>®</sup> EdgeRouter X v1.8.0

Welcome ubnt to edgex

Dashboard Routing Firewall/NAT Services VPN QoS Users Config Tree Wizards

Services

**Routes**

connected	static	rip	ospf	bgp	total
2	1	0	0	0	3

**OSPF** is disabled  
areas: n/a

**NAT** is disabled  
rules: 0

**Firewall** is disabled  
rulesets: 0  
rules: 0

**DHCP** is disabled  
active servers: 0  
inactive servers: 0

**Interfaces**

br0  
eth0  
eth1  
eth2  
eth3  
eth4  
switch0

Interface Configuration for br0

Basic Advanced Bridge Interfaces

☐ eth0  
☒ eth1  
☒ eth2  
☐ eth3  
☐ eth4  
☐ switch0

Save Cancel

Description	Interface	Type	MTU	Tx Rate (Mbps)	Rx Rate (Kbps)	Status	Actions
br0	br0	bridge	1500	0 bps	0 bps	Connected	Actions
eth0	eth0	ethernet	1500	53.66 Mbps	686.12 Kbps	Connected	Actions
eth1	eth1	ethernet	1500	0 bps	0 bps	Connected	Actions
eth2	eth2	ethernet	1500	0 bps	0 bps	Connected	Actions
eth3	eth3	ethernet	1500	0 bps	0 bps	Disconnected	Actions
eth4	eth4	ethernet	1500	0 bps	0 bps	Disconnected	Actions

Showing 1 to 7 of 7 entries

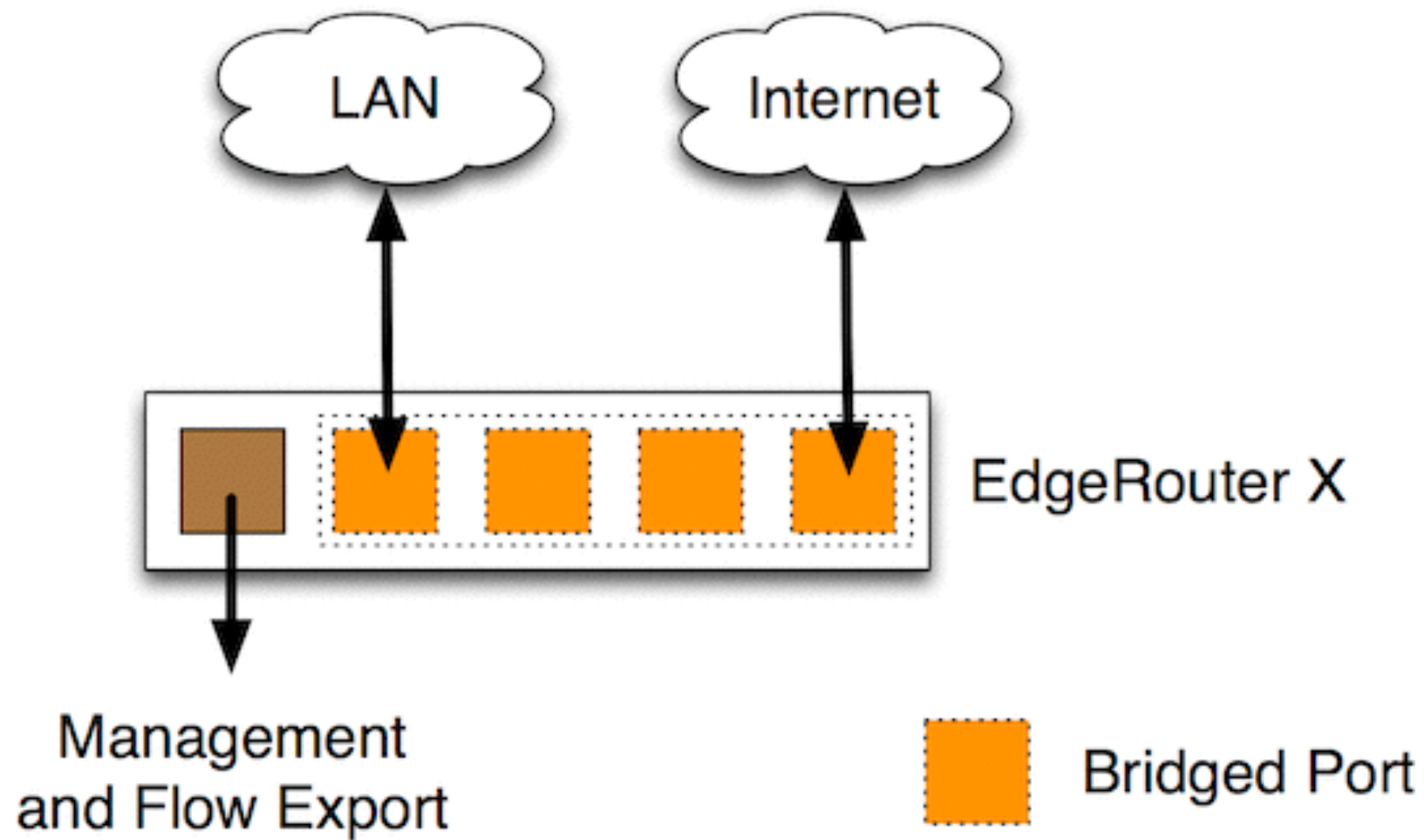
© Copyright 2012-2015 Ubiquiti Networks, Inc.



X



# Ubiquiti EdgeRouter-X [4/7]



# Ubiquiti EdgeRouter-X [5/7]

Wireshark · Interface Options: SSH remote capture: ssh

Remote SSH server address	192.168.2.113
Remote SSH server port	22
Remote SSH server username	ubnt
Remote SSH server password	....
Path to SSH private key	<input type="text"/> ...
SSH key passphrase	<input type="text"/>
Remote interface	<input type="text"/>
Remote capture command	/usr/sbin/tcpdump -i eth0 -U -w -
	<input checked="" type="checkbox"/> Use sudo on the remote machine
Remote capture filter	<input type="text"/>
Packets to capture	5

☒ Save parameter on capture start

Help Restore Defaults Close Start



# Ubiquiti EdgeRouter-X [6/7]

```
deri@Lucas-iMac 212> ssh ubnt@192.168.2.113 "sudo /usr/sbin/tcpdump -i eth0 -U -w -" | ./ntopng -i -  
Welcome to EdgeOS
```

By logging in, accessing, or using the Ubiquiti product, you acknowledge that you have read and understood the Ubiquiti License Agreement (available in the Web UI at, by default, <http://192.168.1.1>) and agree to be bound by its terms.

```
ubnt@192.168.2.113's password: 30/Oct/2017 08:19:04 [Ntop.cpp:1437] Setting local networks to 127.0.0.0/8  
30/Oct/2017 08:19:04 [Redis.cpp:111] Successfully connected to redis 127.0.0.1:6379@0  
30/Oct/2017 08:19:04 [Redis.cpp:111] Successfully connected to redis 127.0.0.1:6379@0  
...  
30/Oct/2017 08:19:07 [HTTPserver.cpp:915] HTTP server listening on port(s) 3000,4000  
30/Oct/2017 08:19:07 [main.cpp:394] Working directory: /var/tmp/ntopng  
30/Oct/2017 08:19:07 [main.cpp:396] Scripts/HTML pages directory: /Users/deri/network/ntopng  
30/Oct/2017 08:19:07 [Ntop.cpp:350] Welcome to ntopng x86_64 v.3.1.171025 - (C) 1998-17  
ntop.org  
30/Oct/2017 08:19:07 [Ntop.cpp:360] Built on MacOSX 10.13  
...
```



# Ubiquiti EdgeRouter-X [7/7]



🏠 - **Flows** - Hosts - Devices - Interfaces - ⚙️ - 🔌 - 🔍 Search Host

## Flows

	Application	L4 Proto	Client	Server	Duration	Breakdown	Actual Thpt	Total Bytes	Info
<a href="#">Info</a>	SSH 🔓	TCP	192.168.2.113:ssh	LUCAS-IMAC:51623	6 min, 35 sec	Client	0 bit/s ➡	1.94 GB	
<a href="#">Info</a>	SSL 🔒	TCP	192.168.2.113:https	LUCAS-IMAC:51434	6 min, 33 sec	Client	0 bit/s ➡	1.09 MB	
<a href="#">Info</a>	SSL_No_Cert 🔓	TCP	LUCAS-IMAC:51627	192.168.2.113:https	7 sec	Client Server	0 bps	2.91 KB	
<a href="#">Info</a>	SSL_No_Cert 🔓	TCP	LUCAS-IMAC:51655	192.168.2.113:https	7 sec	Client Server	471.62 bit/s ⬆	2.91 KB	
<a href="#">Info</a>	SSL_No_Cert 🔓	TCP	LUCAS-IMAC:51626	192.168.2.113:https	0 sec	Client Server	0 bps	1.56 KB	
<a href="#">Info</a>	SSL_No_Cert 🔓	TCP	LUCAS-IMAC:51664	192.168.2.113:https	0 sec	Client Server	0 bps	1.56 KB	
<a href="#">Info</a>	? Unknown	⚠️ TCP	192.168.2.113:https	LUCAS-IMAC:51622	1 sec	Client Server	0 bps	427 Bytes	
<a href="#">Info</a>	NTP ⌚	UDP	192.168.2.113:ntp	188.213.165.209 🇮🇹:ntp	0 sec	Client Server	0 bit/s ➡	180 Bytes	
<a href="#">Info</a>	Spotify 🎧	UDP	LUCAS-IMAC:57621	192.168.2.255:57621	6 min, 1 sec	Client	0 bit/s ➡	1.01 KB	

Showing 1 to 9 of 9 rows

ntopng Enterprise Edition v.3.1.171025  
User [admin](#) Interface [admin](#)

🕒 06:50:04 40100 | Uptime: 11 min, 1 sec  
10 L 3 Devices 51 Flows



X



# Some Remarks

- You can use these boxes as “tap alternatives” for remote monitoring.
- On the EdgeRouter X there is 128 MB of free memory and it is possible to run nprobe on it that can stream flows to a remote collector (e.g. ntopng).
- Storage (no USB for adding an external disk) and memory constraints prevent running on the box more sophisticated tools (e.g. ntopng) or dump packet for troubleshooting.
- A local computer is still necessary so it is not possible to implement a single-box solution.



X



# Part 2: Low Cost General Purpose Hardware





# Problem Statement

- Create a low-cost solution featuring
  - 1 Gbit ports: even if the speed might be limited the port must be Gbit (and not Fast Ethernet) as Internet speed is increasing and it is not uncommon to have fast Internet connections (In Italy 200 Mbit Internet connection costs 29.95 Euro).
  - Dual ports: traffic must flow inside the box so that it acts as a bump-in-the-wire, and be disconnected from the network in case of fault.
  - It can be used to collect packets/flows streamed from devices (as previously discussed).



# Low Cost Boxes: \*PI Solutions

- There are many solutions on the market for makers. The most popular is the Raspberry PI and variations (BananaPI, OrangePI, Pine64 ...)

μSD: not good for writing  
(BananaPI... have eMMC)

Case  
Unless you choose a  
PI-compatible board,  
the case is a problem

USB 2.0 (no USB 3)

Single Fast Ethernet  
(BananaPI... have Gigabit)

Beside the PI most boards  
are not CE/FCC Certified.  
Even the PI is sold as kit  
so you leave the problem  
up to your users



x



# Low Cost Boxes: Dual Ethernet [1/2]

- Hardware bypass allows the network not to be disrupted in case of hardware fault or monitoring application crash.
- Compulsory for traffic enforcement applications, desirable for passive. monitoring application
- Boxes with internal ethernet bypass costs 500\$ and up and require the software to be modified in order to talk with the watchdog for keep-alive updates. So these solutions will be relatively low-cost.



X



# Low Cost Boxes: Dual Ethernet [2/2]

- It is possible to add dual ethernet (no bypass) to single port boxes by means of an USB external ethernet port.
- On the market there are also dual port gigabit ethernet adapters (40-60 Euro price range) so that you can use the on-board ethernet for management and the extra ports for monitoring.



# Low Cost Boxes: Ethernet Speed

Even if you use a gigabit ethernet adapter over USB remember that:

- USB ports are 2.0 (no 3.0) so speed is limited to 480 Mbit (in theory)
- Even if you use a gigabit adapter you won't usually be able to go above 200 Mbit.

Connection	Pi	Download	LAN – from Pi	LAN – to Pi	iperf
<a href="#">Edimax USB 802.11n</a>	B+	2.94 MB/s	2.59 MB/s	2.87 MB/s	44.5 Mbps
Onboard WiFi <sup>1</sup>	3 B	3.86 MB/s	0.48 MB/s	0.75 MB/s	38.5 Mbps
Onboard LAN	B+	2.69 MB/s	2.66 MB/s	3.43 MB/s	94.4 Mbps
Onboard LAN	2 B	8.54 MB/s	7.77 MB/s	9.12 MB/s	94.8 Mbps
Onboard LAN	3 B	3.45 MB/s	9.12 MB/s	8.39 MB/s	94.8 Mbps
<a href="#">USB3 GigE</a>	B+	6.32 MB/s	2.95 MB/s	3.44 MB/s	222 Mbps



X





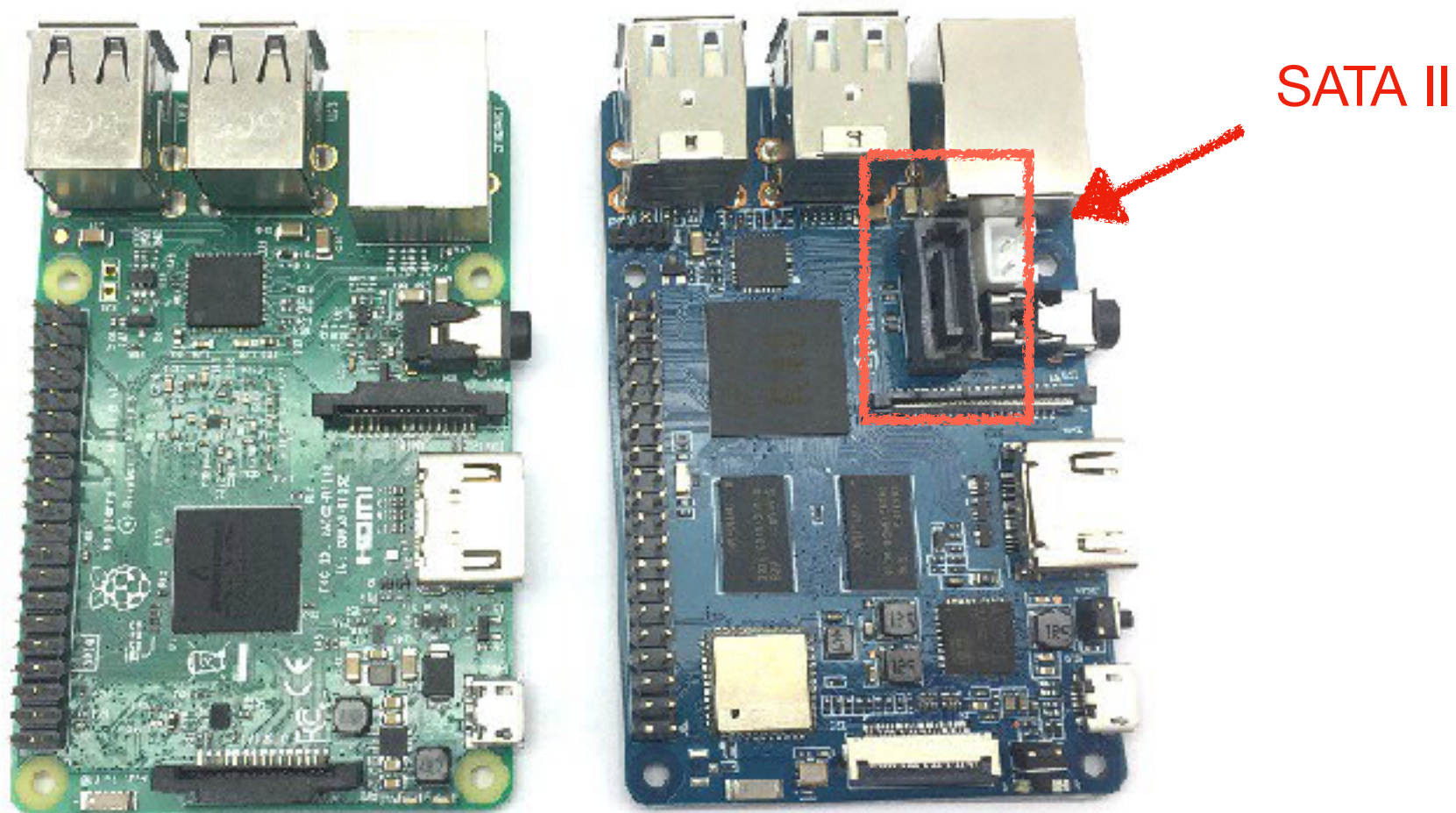
# Low Cost Boxes: Traffic Storage [1/3]

- Storage is important for dumping traffic to disk or writing flows
- Most boxes are limited to  $\mu$ SD for storage that is good for boot but not for writing it. Problems: speed and reliability.
- USB can be used to connect an external drive or you can mount a network drive or NAS for storing data (not a good idea as will be limited by the ethernet speed that is used to capture traffic)



# Low Cost Boxes: Traffic Storage [2/3]

- There are some PI boards (e.g. Banana PI M2 Berry/Ultra) that feature an internal/external SATA port.



- Warning: adding storage will prevent you from using PI cases so you need to build your own,

# Low Cost Boxes: Traffic Storage [3/3]

- Total price ~30 Euro (excluding HDD) but no CE/FCC.
- 1 GB RAM, Quad Core, Ubuntu 16.04 LTS.
- Suitable for ntop/nProbe/n2disk.
- It comes with an optional aluminium case.



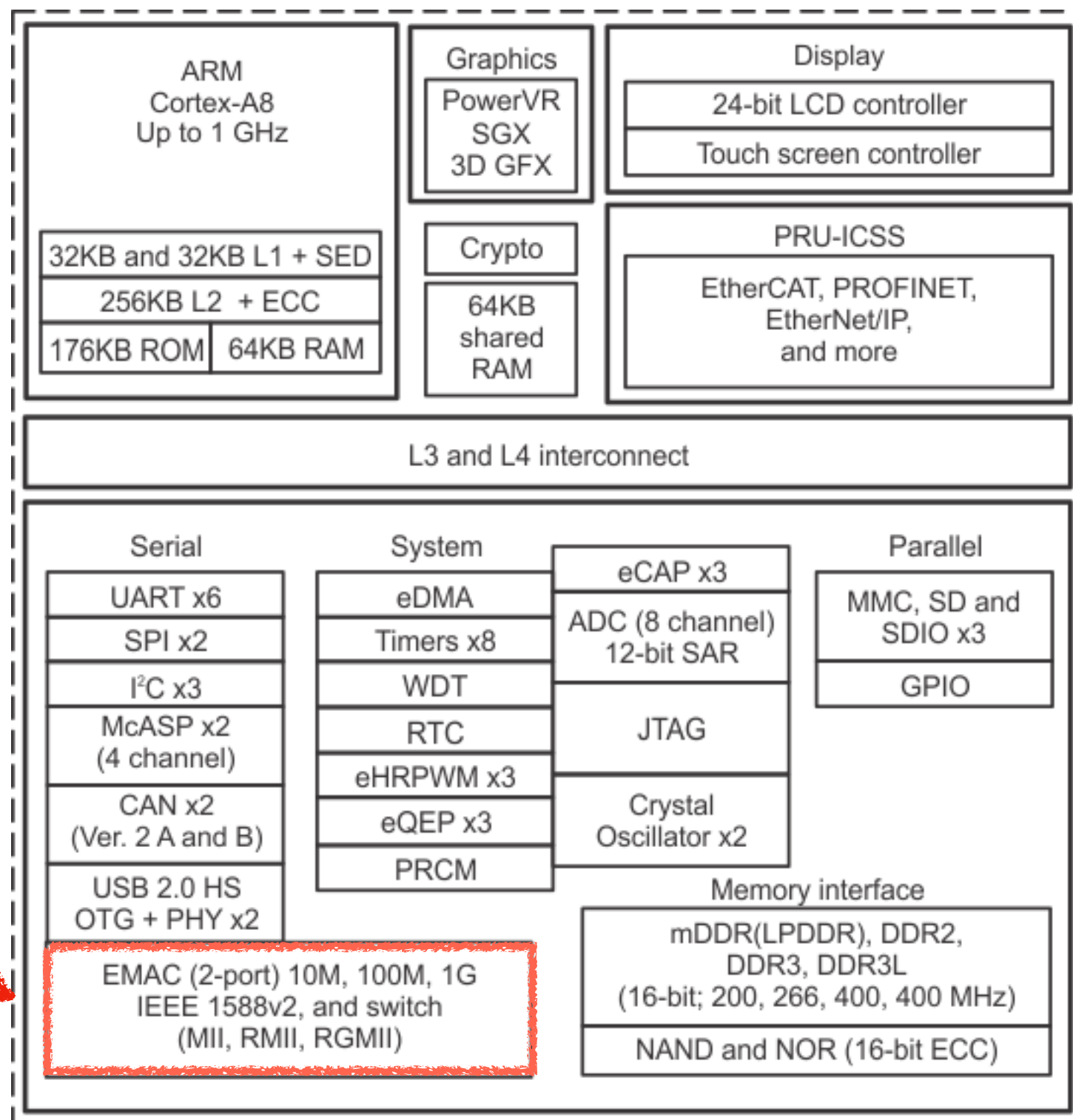
<http://www.friendlyarm.com>

x





# Low Cost Boxes: SOC [1/3]



Dual Ethernet  
Embedded Switch



X



# Low Cost Boxes: SOC [2/3]

- Ethernet ports can either be physical or logical over an ethernet switch (very common in particular on low-cost xDSL routers).
- Programming the switch (usually by writing on a register) it is possible to see from the OS:
  - Two ethernet ports (e.g. eth0 and eth1) and the application has to bridge traffic in software (e.g. br0).
  - A single ethernet port (e.g. eth0) where the embedded switch is bridging traffic in hardware and thus theoretically at line rate.
- This solution is used on the EdgeRouter-X and on other low-cost boxes.



# Low Cost Boxes: SOC [3/3]

The embedded switch is an interesting solution but in many cases it has some limitations:

- As port bridging is physically implemented with VLANs (see [http://processors.wiki.ti.com/index.php/Sitara\\_Linux\\_Dual\\_Emac\\_Mode#Dual\\_EMAC\\_Mode](http://processors.wiki.ti.com/index.php/Sitara_Linux_Dual_Emac_Mode#Dual_EMAC_Mode)), bridging VLAN tagged packets might be a problem.
- Software bridging in dual MAC mode is not supported ([https://e2e.ti.com/support/arm/sitara\\_arm/f/791/p/560213/2051035](https://e2e.ti.com/support/arm/sitara_arm/f/791/p/560213/2051035))

```
[root@alarm alarm]# brctl addif br0 eth0
[ 606.819494] br0: port 1(eth0) entered blocking state
[ 606.824613] br0: port 1(eth0) entered disabled state
[ 606.839543] cpsw 4a100000.ethernet eth0: failed to initialize vlan filtering on this port
[ 606.856116] br0: port 1(eth0) entered blocking state
[ 606.861167] br0: port 1(eth0) entered disabled state
[ 606.874195] cpsw 4a100000.ethernet eth0: failed to initialize vlan filtering on this port
can't add eth0 to bridge br0: Invalid argument
```



X



# ADI Engineering uFW

## Pro:

- Low cost device (< 100\$) with dual ethernet.
- It comes with a 4GB EMMC where you can dump some traffic (or not be concerned about  $\mu$ SD reliability).



## Cons:

- Single-core CPU (< 1 GHz).
- No USB for external storage.
- In-kernel ethernet bridging does not work (SOC).
- Limited Ethernet speed (no line rate).

<http://www.adiengineering.com>

X



# CatchWire

- Similar (ARM) to uFW with the following differences
  - USB port to be used for connecting storage.
  - PoE support.
  - Shipped with monitoring tools including Wireshark and nProbe.

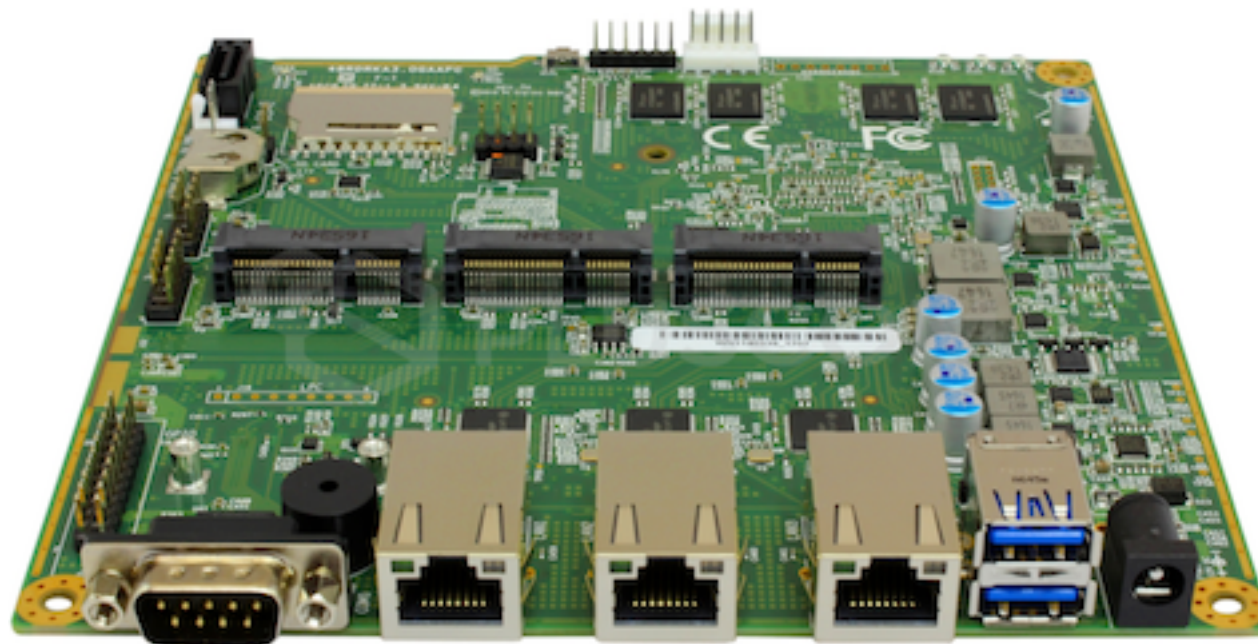


<https://catchwire.net>  
X



# APU2

- Quad core CPU (AMD x64), 2/4 GB or RAM (85 Euro+).
- Two or three ethernet ports: management + bridging.
- Pizza-box like enclosure, based on Linux.
- Bridging speed > 400 Mbit.
- Supports all ntop tools (ntopng, n2disk, PF\_RING).



<http://www.pcengines.ch>  
X





# Operating System [1/2]

- One of the assumptions we have made while designing a low-cost probe for our community was that the box had to run Linux.
- If you are used to install your distro and do apt-get to update your tools, on a non-x86 platform life is much harder because:
  - There is no BIOS and thus the boot image must be provided by the vendor on a case-by-case basis.
  - Usually manufacturers pay little attention to updates and thus you get a kernel and you are stuck with it.
  - Most manufacturers support a specific kernel version (often 3.x series) and you have to stick with it forever.



# Operating System [2/2]

- On ARM-land you need to make sure, before buying a board, that
  - The board is natively supported by one of the main distri suppliers (e.g. armbian).
  - You can install a vibrant community-based distribution like Raspbian (🍷).
  - Do not look just at specs (e.g. memory and CPU speed), test the board. For cost reasons, not all boards have peripherals properly connected and thus the network speed is not alike across similar boards.
  - Often you have no video, sometimes a serial port. Make sure you do not brick the board !





# Solution Comparison

	Ubiquity	Raspberry*	uFW	Catchwire	PC Engines
Architecture	MIPS	ARM	ARM	ARM	x86
Wireshark	Via extcap	Yes	tshark	tshark	Yes
PF_RING	No	Yes	Yes	Yes	Yes
ntopng	No	Yes	Yes	Yes	Yes
nProbe	No	Yes	Yes	Yes	Yes
n2disk	No	Yes	Yes	Yes	Yes
Ethernet Ports	5	1+WiFi	2	2	3+WiFi
Bridging Performance	7	6	4	4	9
Price	\$	\$	\$	\$\$\$	\$\$



X



# What's Next ?

- Ntop distributes packages tools for ARM, MIPS and x64 at <http://packages.ntop.org>
- Thanks to PF\_RING, at a lower degree of performance, you can run on an embedded box the same applications you run on a powerful x86 box.
- Starting at ~60 Euro you can create your remote Wireshark/ntop/nProbe monitoring box.
- Why low cost? Because many networks are remote, not all traffic passes through a central point, monitoring should be lightweight and cheap, troubleshooting is not an option.

