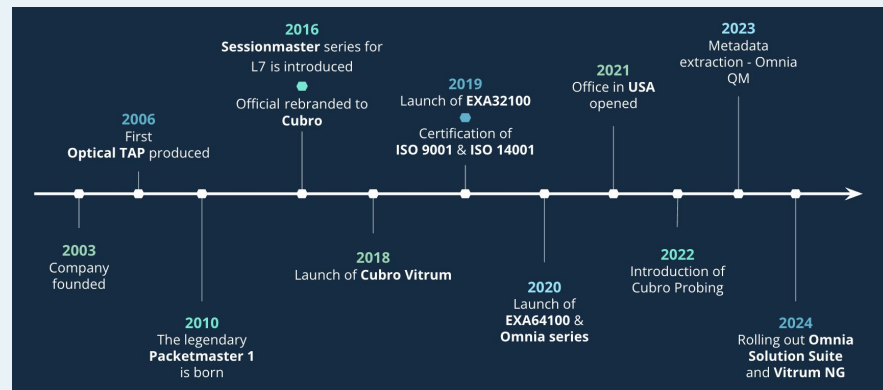
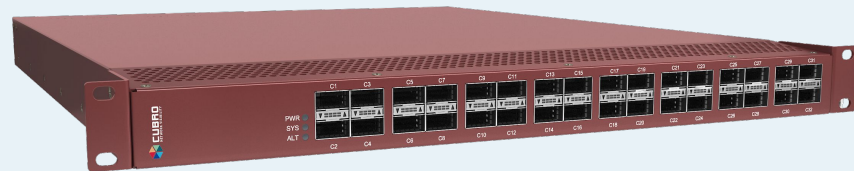


# G5 Plus - Advanced Network Packet Broker - Overview

June 2025

- Cubro Generation 5 (none +) was launched in 2018.
  - EXA48600 - 48 x 1G/10G & 6 x 40G/100G
  - EXA32100 - 32 x 40G/100G
- Based on Cavium Xpliant programmable chipset. Offered superior features compared to Broadcom chipset based products.
  - Tunnel termination and inside tunnel filtering
  - Number of simultaneous rules



# What is Generation 5+ (G5+) of Advanced NPBs?

G5+ family consists of four products that are all based on latest generation of programmable Ethernet-Switch ASIC.

- EXA32100A - 32 x 40G/100G & 2 x 10G/25G
- **EXA64100 - 64 x 40G/100G & 2 x 10G/25G**
- EXA32400 - 32 x 100G/400G
- EX48800 - 48 x 10G/25G & 8x 40G/100G



**EXA64100**

G5+ key points:

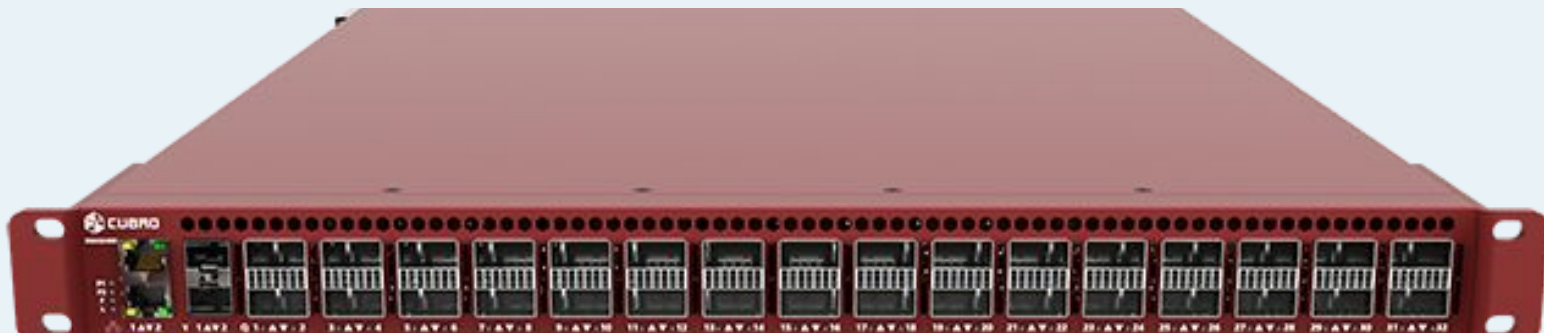
- Tunnel Termination
- State-of-the art VXLAN handling including VNI filtering
- Inner tunnel filtering
- Superior Load-balancing features including inner tunnel hashing
- More than 100k parallel filtering rules

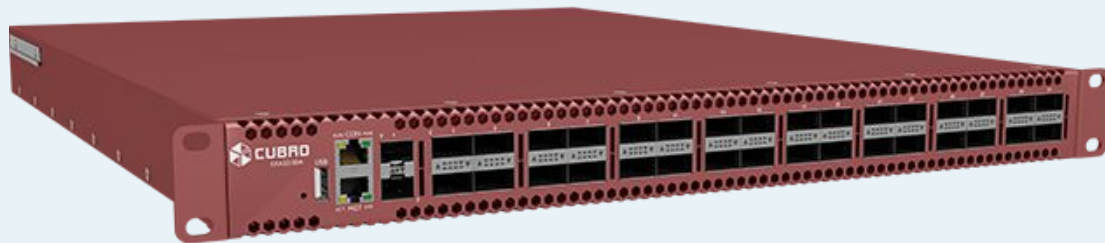
# Technical Details



# G5 Plus Overview

The G5 Plus series is Cubro's market-leading Advanced Network Packet Broker, built on a state-of-the-art multi-core, programmable switch chip. It delivers unmatched performance with full hardware-level traffic filtering. With advanced tunnel decapsulation and inner tunnel filtering, it's ideal for modern overlay networks.





- **32 x 40G/100G** – each of these ports can be used in 4 x 10G or 4 x 25G or 2 x 50G split mode.
- 2 x native SFP+/SFP28 ports for 10G/25G
- Each port can be used simultaneously as input and output and is totally independent of other ports
- Non-blocking architecture
  - 6,5 Tbit/s throughput
  - 2,4B pps packet forwarding
- All ports are included and open to 3rd party transceivers



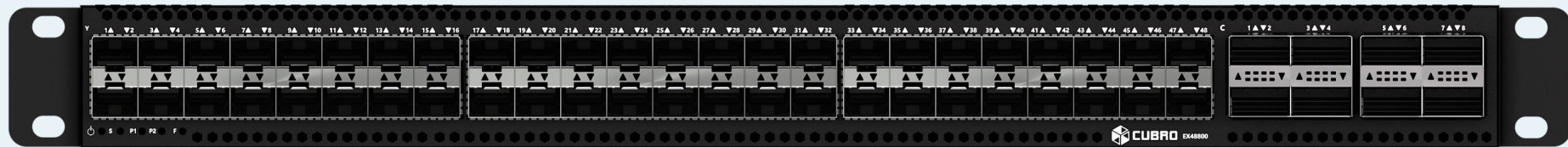
- **64 x 40G/100G** - each of these ports can be used in 4 x 10G or 4 x 25G or 2 x 50G split mode.
- 2 x native SFP+/SFP28 ports for 10G/25G
- Each port can be used simultaneously as input and output and is totally independent from other ports
- Non-blocking architecture
  - 12,9 Tbit/s throughput
  - 4,8B pps packet forwarding
- All ports are included and open to 3rd party transceivers





- **32 x 100G/400G** via QSFP28/QSFP-DD
- 128 x 100G when 100G split mode is activated
- Each port can be used simultaneously as input and output and is totally independent of other ports
- Non-blocking architecture
  - 25,6 Tbit/s throughput
  - 6B pps packet forwarding
- All ports are included and open to 3rd party transceivers





- 4 x 1 Gbps / 10 Gbps / 25 Gbps full duplex ports for any kind of SFP/SFP+/SFP28
- 44 x 10 Gbps / 25 Gbps full duplex ports for any kind of SFP+/SFP28
- 8 x 40 Gbps / 100 Gbps full duplex ports for any kind of QSFP/QSFP28
- No transceiver vendor lock
- Each port can be used simultaneously as input and output and is totally independent from other ports
- Non-blocking architecture (4000 Gbit/s Throughput)
- Port Licensing Model available

# Port licensing model only for EX48800



The EX48800 offers 4 different licensing models defining the number of available ports.

- EX48800-12 = last 12 x 25/10G ports + 8 x 40/100G activated
- EX48800-24 = last 24 x 25/10G ports + 8 x 40/100G activated
- EX48800-36 = last 36 x 25/10G ports + 8 x 40/100G activated
- EX48800-48 = all 48 x 25/10G ports + 8 x 40/100G activated

Unlicensed ports are blocked and cannot be used for any purpose like ingress, egress or loopback.

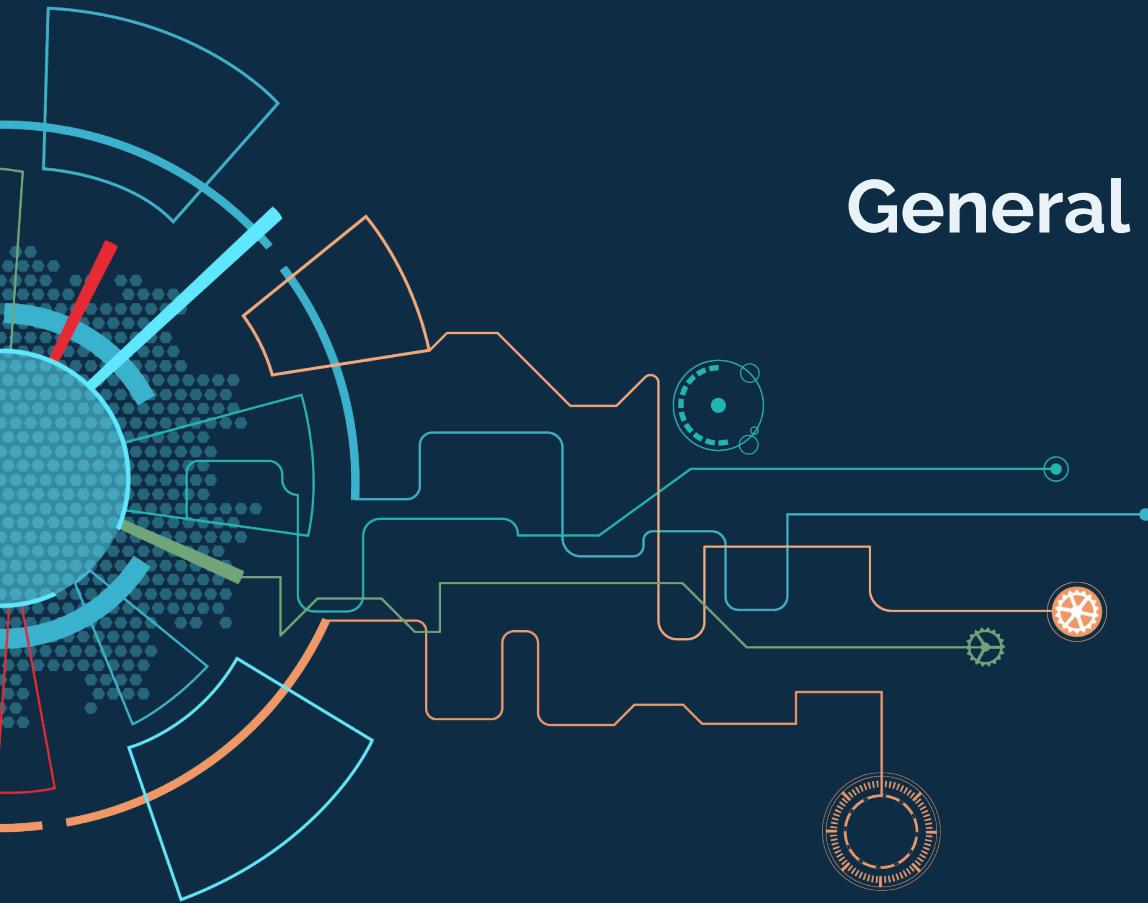
The port licensing model has no effect on the included features.

Pay only for what you need.  
A smart, cost-efficient approach tailored to your requirements.

- 10G/25G/50G/100G break-out mode
- Non-blocking
- Aggregation, Filtering & Load-balancing
- Buffer memory for burst protection
- Open for third party optical modules
- NTP and PTP synchronization
- TACACS+ and RADIUS Authentication
- SNMPv2c, SNMPv3 and RSyslog
- MS Excel filter upload
- Easy to use WebUI, RestAPI and CLI
- Packet Slicing in line rate on all ports for any packet size
- > 100k filtering rule capacity (IPv4 and Ipv6)
- **Tunnel Termination and inside tunnel filtering**
  - GRE, GTP, MPLS, MPLSoGRE, MPLSoUDP, VXLAN, ERSPAN, CFP
- Superior VXLAN traffic handling (VXLAN VNI & inner IP filtering simultaneously)
- Active Tunnel Endpoint / Termination & Encapsulation

Best in-class Advanced NPB

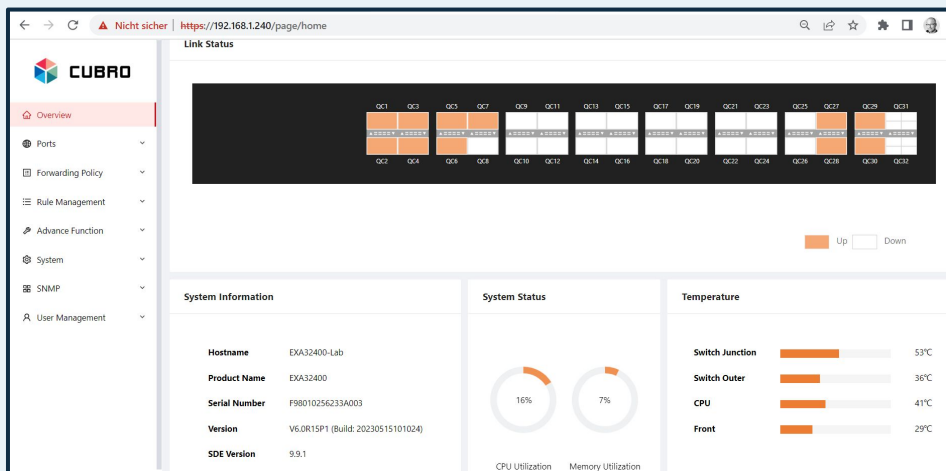
# General Features and Functions



# Straightforward operation via WebUI or CLI



- Straight and easy operation via WebUI or CLI; RestAPI available for easy system integration



```

Welcome to the UNIX shell of this Cubro EXA32400. Please use it with care!!

Access the Cubro CLI Shell to customize your device!

> exmenu

Last login: Mon May 22 05:11:27 2023 from 192.168.0.185
admin@EXA32400-Lab:~$ sudo vtysh
[sudo] password for admin:
EXA32400-Lab# configure terminal
EXA32400-Lab(config)# interface 1
EXA32400-Lab(config-if)# speed 100000
EXA32400-Lab(config-if)# exit
EXA32400-Lab(config)#
```

# Forwarding Policy via drag & drop

## Policy

Choose Interface

QC1

QC3

QC5

QC7

QC9

QC11

QC13

QC15

QC17

QC19

QC21

QC23

QC25

QC27

QC29

QC31

1	3
2	4

QC2

QC4

QC6

QC8

QC10

QC12

QC14

QC16

QC18

QC20

QC22

QC24

QC26

QC28

QC30

QC32

1	3
2	4

+ Forward Policy

Ingress Port

QC8

QC10

QC11

QC12

Ingress Port Group

401

QC13

QC14

Egress Port Group

Add Egress Port Group

# Create filters with MS Excel® & upload to G5plus

- Filtering rules can be easily created and modified via MS Excel® and simply uploaded to the device.

	A	C	E	F	G	K	L	M	N	O	P	Q	R	S	T	
1	ace_id	i	ingress_ports	action	egress_name	ethertype	match_vlan_1	match_vlan_2	match_vlan_3	match_vlan_4	src_ip		dst_ip	protocol	src_port	dst_port
3	105001	C1		forward	C32	double-vlan	100									
4	105002	C1		forward	C32	double-vlan		1000								
5	105003	C1		forward	C32	single-vlan	500				10.0.0.1					
6	107001	C1		forward	C32							10.0.0.2				
7	107002	C1		forward	C32								udp		2152	
8	107003	C1		forward	C32								udp			80
9	107004	C1		forward	C32						10.0.0.3		udp		1000	1100
10	105004	C1		deny		single-vlan	500				10.0.0.4		udp		1000	1100
11	105005	C1		forward	C32	double-vlan	700									
12	105006	C1		forward	C32	double-vlan		1300								
13	105007	C1		forward	C32	single-vlan	1500				123:4567:8910:1112:1314:1516:0:3					
14	107005	C1		forward	C32							123:4567:8910:1112:1314:1516:0:4				
													udp		2153	
													udp			81
											123:4567:8910:1112:1314:1516:0:7		udp		1200	1300
											123:4567:8910:1112:1314:1516:0:4		udp		1200	1300
											10.0.0.4					
						lan	100	200	300	400	10.0.0.4					

## Ingress Rule

### Ingress Rule

 Display/Hide Columns

 Clear Hitcount

 Import All Rules ▾

 Export All Rules ▾

 Delete All Rule

 JSON

 Excel



Rule ID ▾

Filter Key

Action

Ingress Port



# Graphical Throughput per port



Port utilization over time to visualize traffic trends early.

# SNMP management integration and supervision

SNMPv2c and v3 is supported and thus G5 plus can be easily integrated into any SNMP supervision system. MIB file is provided by Cubro.

### SNMP Config

SNMP Server Config

SNMP V3 Users

SNMP Trap Config

* OID	<input type="text" value="32182"/>	1-99999
System Location	<input type="text"/>	Vienna
System Contact	<input type="text"/>	support@cubro.com
User Description	<input type="text"/>	
* Snmp Port	<input type="text" value="161"/>	161
* SNMP Community	<input type="text" value="cubro"/>	
	<div><div>+</div></div>	
<div><div>Confirm</div><div>Cancel</div></div>		

### System Trap Config

Power: ☒

Fan: ☒

Module Optical Power: ☒

Temperature: ☒  °C

Module Temperature: ☒  °C

CPU Utilization: ☒  %

Memory Utilization: ☒  %

Confirm

Cancel

For Fault (SNMP trap) and Performance (SNMP get) Management

17

### Configuration

* Source:	<input type="text" value="All"/>	* Level:	<input type="text" value="ERR"/>
* Server IP:	<input type="text" value="192.168.0.185"/>	* Port:	<input type="text" value="514"/>
* Proto:	<input type="text" value="UDP"/>		

Config

Completely user configurable Syslog

- NTP & PTP time synchronization
- Activity Log
- Automatic Backups
- Security hardened, passed successfully several rounds of in-depth PEN tests at a major European telecom operator

Activity Log

Global Config

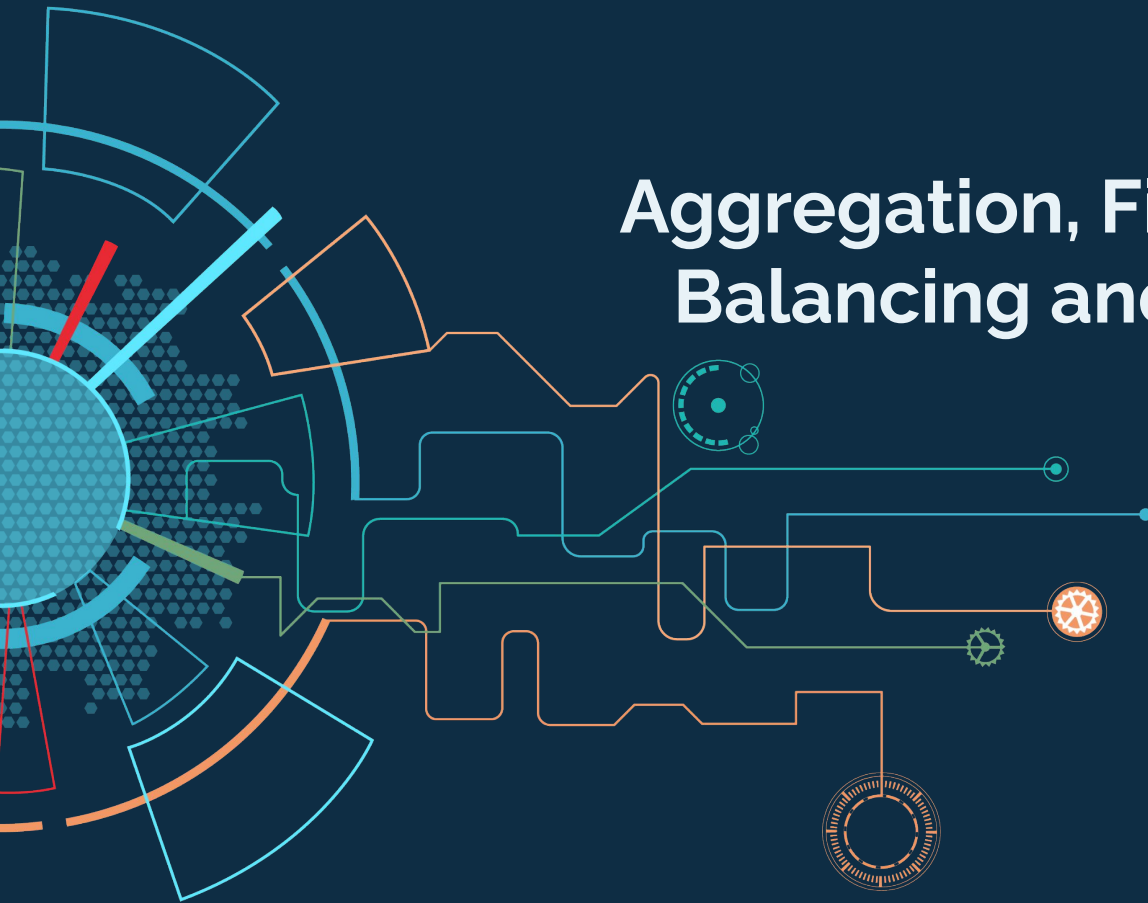
WEB Log

Export Excel

Clear

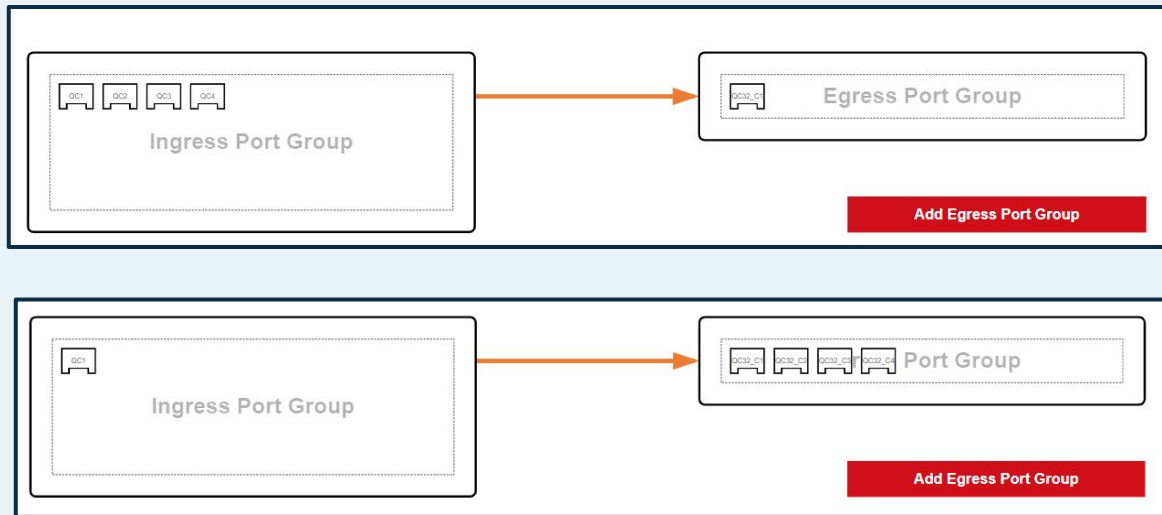
Row ID	Username	Date	IP	Module	Operation	Result	Log
1	admin	2023-05-25 16:37	10.0.8.3	Forwarding Policy	UPDATE	FAILED	Edit a forwarding policy whose ingress port group is QC31_C1.
2	admin	2023-05-25 16:37	10.0.8.3	Forwarding Policy	CREATE	SUCCESS	Create a forwarding policy whose ingress port group is QC9,QC10,QC11,QC12.
3	admin	2023-05-25 16:36	10.0.8.3	Forwarding Policy	CREATE	SUCCESS	Configure the type of the egress port group as copy.The egress port group is: [QC13,QC14].

# Aggregation, Filtering, Load Balancing and much more




All kinds of aggregation supported :

- Many to One
- One to Many
- Many to Many



# Split mode - E.g. 400G into 4 x 100G



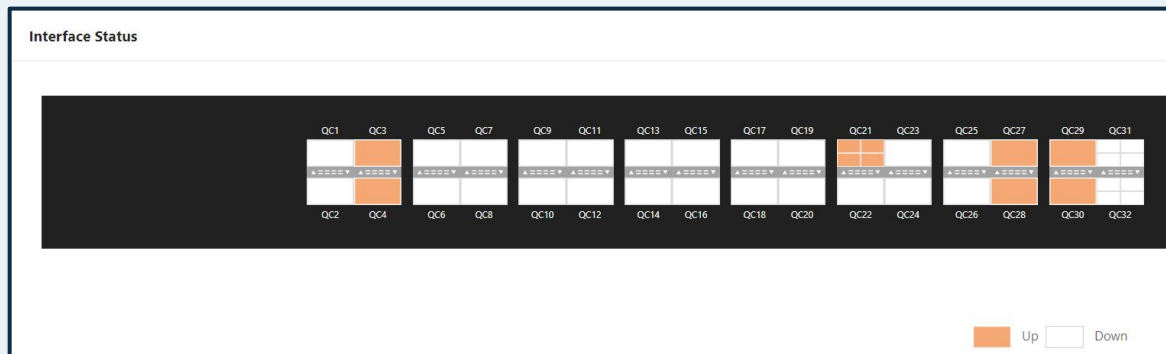
- Overview
- Ports
- Config
- Statistics
- Forwarding Policy
- Rule Management

### Interface Config

☒ Display/Hide Columns

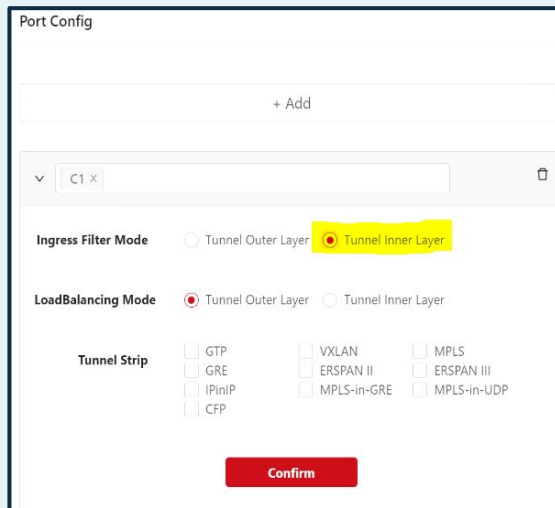
Multi-interfaces Config

Port ID	Enable	Type	Category	Hash Mode	Speed (Mbps)	Split	Split Speed	FEC	Mirror TX	Mirror RX	Hash Seed
QC21	<input checked="" type="checkbox"/>	Ingress Port	mixed	I3-src-dst	400000	4	100000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
QC22	<input checked="" type="checkbox"/>	Ingress Port	mixed	I3-src-dst	100000	0	100000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0
QC23	<input checked="" type="checkbox"/>	Ingress Port	mixed	I3-src-dst	400000	0	-	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0





- Layer 2
  - MAC, VLAN (up to 4 tags)
  - Ether type
  - VXLAN VNI
- Layer 3
  - Protocol
  - DSCP
  - IPv4/IPv6 Address
  - Fragments
- Layer 4
  - Port Number
  - TCP Flag
- Payload
  - ASCII string / Hex pattern



The image shows a 'Port Config' window with a '+ Add' button at the top. Below it is a dropdown menu showing 'C1'. The 'Ingress Filter Mode' section has two radio buttons: 'Tunnel Outer Layer' and 'Tunnel Inner Layer', with the latter selected and highlighted in yellow. The 'LoadBalancing Mode' section has two radio buttons: 'Tunnel Outer Layer' (selected) and 'Tunnel Inner Layer'. Below these is a 'Tunnel Strip' section with a grid of checkboxes for GTP, GRE, IPinIP, CFP, VXLAN, ERSPAN II, MPLS-in-GRE, MPLS, ERSPAN III, and MPLS-in-UDP. A red 'Confirm' button is at the bottom.

- Ingress Filtering
- Egress Filtering
- Middle-stage filtering (via Loopback port function)

**Feed only relevant traffic to the probe/analyzers**

# High number of parallel filtering rules

Number of Rules	Filtering parameter
2048	MAC Addresses
102400	IP Addresses, Protocol type, Port Nr. (five tuple)
2048	Any filtering parameter excluding MAC and String.
8172	Any filtering parameter excluding MAC, VLAN ID and String
1025	ASCII string or Hex Pattern inside payload with defined offset

### Rule Configuration

Wildcard Accurate MAC String

Add Rule Display/Hide Columns Rule ID Query

	Rule ID	Valid	Filter Key									
			Ace Type	IP Version	Source IP	Destination IP	Protocol	DSCP	Frag	Source Port	Destination Port	Tcp
<input type="checkbox"/>	300	<input checked="" type="checkbox"/>	ip	ipv4	10.0.0.1/22			udp	none			

# ASCII string / Hex pattern filtering inside payload

- Filter not only on packet header fields like MAC Address, IP Address or TCP/UDP port numbers but also inside the payload.
- The ASCII string filter functions allows searching for keywords or hex patterns at a defined offset
- E.g. filter out all http “GET” messages from a packet stream.

### Rule Configuration

Wildcard Match      Accurate Match      MAC      **String**

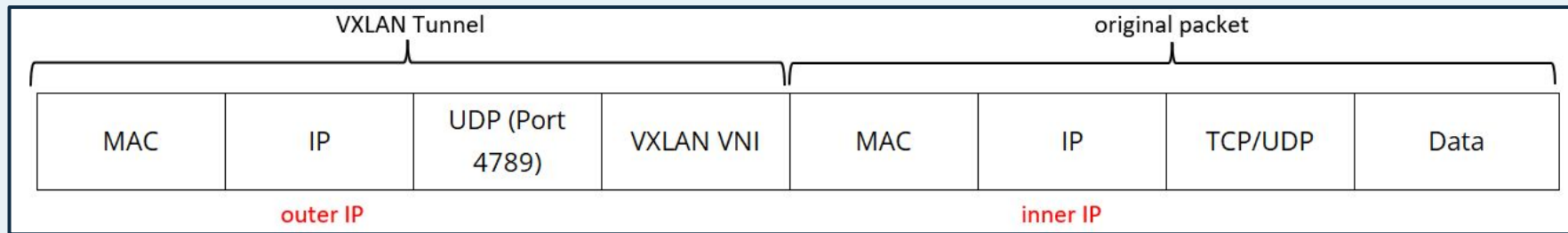
Add Rule Display/Hide Columns

<input type="checkbox"/>	Rule ID	Ace Type	Filter Key	
			Offset	Filter Value
<input type="checkbox"/>	<input type="text" value="114689"/>	string	0	<input type="text" value="GET"/>

Use-case: filter-out 5G user-plane via extended GTP-U header, separate 3G/4G from 5G user-plane

# Encapsulated / Tunneled traffic handling

In modern overlay communication networks, packets are usually encapsulated in tunnels. Typical encapsulations used are VXLAN, GRE or ERSPAN.



## Challenges & Solutions

- Information of interest is hidden inside tunnel. E.g. DNS information inside VXLAN tunnel (outer UDP port 4789, inner UDP port 53). Requires **inner tunnel filtering**
- Analytics/Probes cannot handle tunnel information or gives misleading results when tunnel is present. Requires **tunnel removal**.
- In many (or all) instances, **session-aware load-balancing** using outer IP is ineffective. Typically, sessions rely on inner IP rather than outer IP. It is necessary to **utilize inner tunnel** information for load-balancing purposes.

# Tunnel Removal

Allows to **remove** a wide variety of **tunnel encapsulations** by simply selecting the tunnel type that should be stripped off and that are not required / unwanted by monitoring tools.

### Port Config

+ Add

QC31\_C1 x

**Ingress Filter Mode**

☒ Tunnel Outer Layer ☐ Tunnel Inner Layer

**LoadBalancing Mode**

☒ Tunnel Outer Layer ☐ Tunnel Inner Layer

**Tunnel Strip**

☐ GTP  
☐ GRE  
☐ IPinIP  
☐ CFP

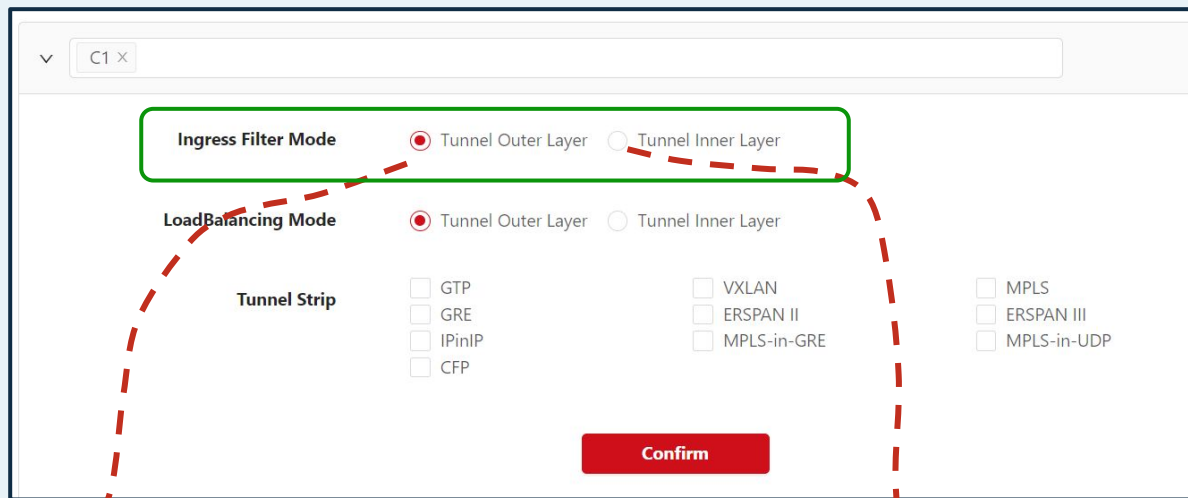
☒ VXLAN  
☐ ERSPAN II  
☐ MPLS-in-GRE  
☐ PPPoE

☐ MPLS  
☐ ERSPAN III  
☐ MPLS-in-UDP

Confirm

# Outer or inner tunnel filtering

G5 plus series provides support for filtering on outer or inner tunnel packet parameters.



The screenshot shows a configuration window for device C1. It features three main sections: 'Ingress Filter Mode', 'LoadBalancing Mode', and 'Tunnel Strip'. Both 'Ingress Filter Mode' and 'LoadBalancing Mode' have radio buttons for 'Tunnel Outer Layer' (selected) and 'Tunnel Inner Layer'. The 'Tunnel Strip' section contains checkboxes for various protocols: GTP, GRE, IPinIP, CFP, VXLAN, ERSPAN II, MPLS-in-GRE, MPLS, ERSPAN III, and MPLS-in-UDP. A red dashed line connects the 'Tunnel Outer Layer' selection in the Ingress Filter Mode section to the 'IP' field in the packet structure diagram below. Another red dashed line connects the 'Tunnel Inner Layer' selection to the 'IP' field in the packet structure diagram below.

**Ingress Filter Mode** ☒ Tunnel Outer Layer ☐ Tunnel Inner Layer

**LoadBalancing Mode** ☒ Tunnel Outer Layer ☐ Tunnel Inner Layer

**Tunnel Strip**

- ☐ GTP
- ☐ GRE
- ☐ IPinIP
- ☐ CFP
- ☐ VXLAN
- ☐ ERSPAN II
- ☐ MPLS-in-GRE
- ☐ MPLS
- ☐ ERSPAN III
- ☐ MPLS-in-UDP

**Confirm**

MAC	IP	UDP (Port 4789)	VXLAN VNI	MAC	IP	TCP/UDP	Data
-----	----	-----------------	-----------	-----	----	---------	------

# Load-balancing

Load-balancing is a vital function to distribute traffic across different monitoring tools evenly and correctly. The Cubro G5+ series supports **session-aware load balancing**. With this feature of the G5+, every packet that belongs to the same conversation/flow is sent to the same physical output port within a load-balancing group.



Add Egress Port Group

## Egress Configuration

Egress Port

QC20 x QC22 x QC23 x

Port Config

Egress Type

☐ Copy ☒ Load Balance ☐ Super Group

Type

☐ Dynamic ☒ Flexible ☐ Static

Port Weight

QC20 QC22

1

1

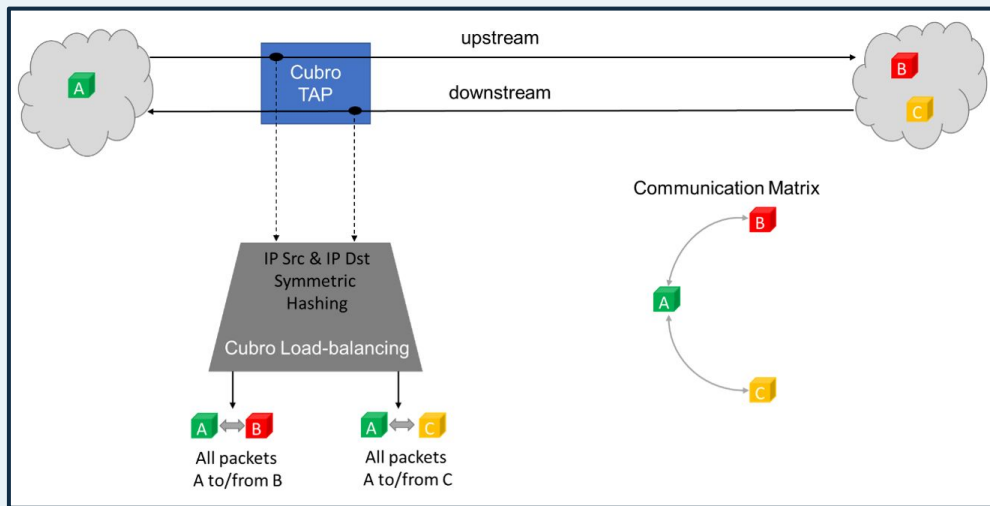
QC23

1



# Session-aware load balancing & Hash-key calculation

Hash-keys are used to define the load-balancing behaviour among the various members (=ports) in the load-balancing group. For example, if hash-key is configured as IP Source and IP Destination Address, then for the hashing calculation only IP Source and IP Destination values are used. Therefore, all packets (=up and downstream) will be available at the same physical output port.



Source	Destination	Hash-key Result	Physical Output Port
A	B	X	1
B	A	X	1
A	C	Y	2
C	A	Y	2

# Hash-key calculation settings

**Interface Config**

[Display/Hide Columns](#) [Multi-interfaces Config](#)

Port ID ▾	Enable	Type	Category	Hash Mode
C1	<input checked="" type="checkbox"/>	Ingress Port ▾	mixed ▾	3-src-dst ▴
C2	<input checked="" type="checkbox"/>	Ingress Port ▾	mixed ▾	l2-src-dst
C3	<input checked="" type="checkbox"/>	Ingress Port ▾	mixed ▾	l3-src
C4	<input checked="" type="checkbox"/>	Ingress Port ▾	mixed ▾	l3-dst
				l3-src-dst
				four-tuple
				five-tuple

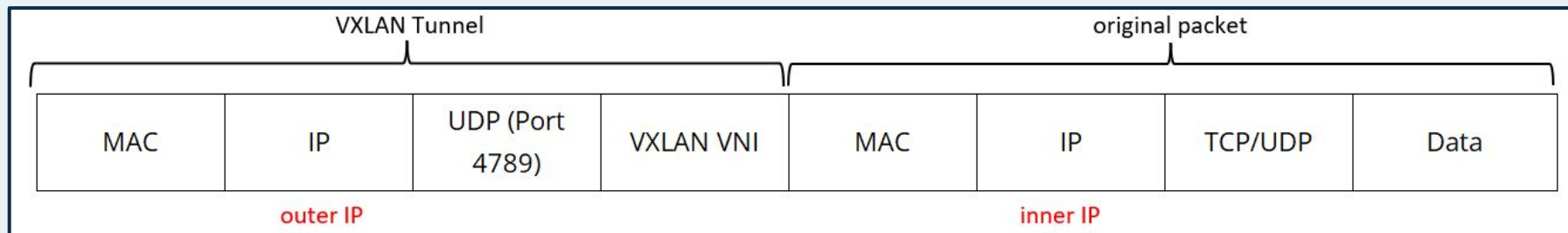
- Full flexibility to cope with all needs
- Individual setting per port

# Hash-key calculation methods

Hash-key calculation method	Hash-key calculation based on	Remark
I2-scr-dst	full MAC Src & MAC Dst Addr	
I3-src	full IP Src Addr	
I3-dst	full IP Dst Addr	
I3-src-dst	full IP Src & IP Dst Addr	Upstream & downstream direction give SAME hash results -> upstream & downstream stay together-> session aware E.g. 10.0.0.1 talks to 10.0.0.2: Hash result = x 10.0.0.2 talks back to 10.0.0.1: Hash result = x
four-tuple	full IP Src & Dst Addr & Layer 4 Src & Dst Port	
five-tuple	full IP Src & Dst Addr & Protocol & Layer 4 Src & Dst Port	

# Encapsulated / Tunneled traffic

In modern overlay communication networks, packets are usually encapsulated in tunnels. Typical encapsulations used are VXLAN, GRE or ERSPAN. Problem is that **several levels of IP** are used.



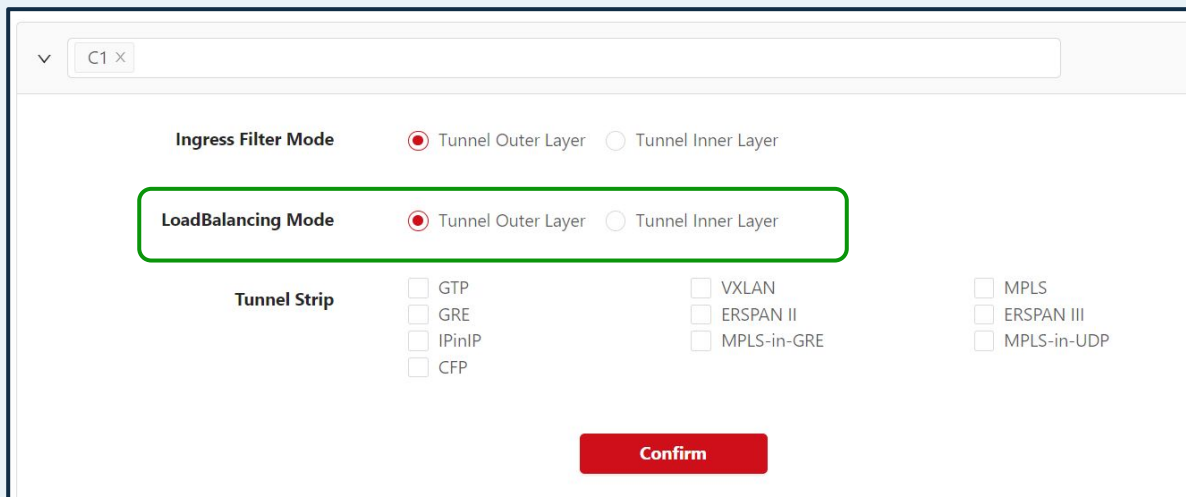
## Avoid Using Outer IPs as hash-criteria

Outer IPs belong to the overlay network and offer limited variation. Relying on them for load balancing can lead to asymmetry and poor traffic distribution. Optimal load balancing needs diverse IP combinations.

A session is usually based on the inner IP (user IP) but not on outer.

# Cubro offers the choice

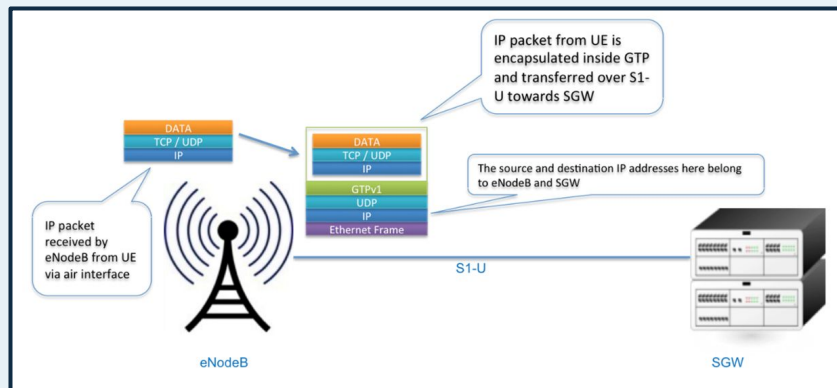
Cubro Advanced NPBs offer the choice to use outer tunnel or inner tunnel information for load-balancing.



The screenshot shows a configuration window for a Cubro Advanced NPB. At the top, there is a tab labeled 'C1'. Below the tab, there are two sections: 'Ingress Filter Mode' and 'LoadBalancing Mode'. Both sections have two radio button options: 'Tunnel Outer Layer' (selected) and 'Tunnel Inner Layer'. The 'LoadBalancing Mode' section is highlighted with a green border. Below these sections, there is a 'Tunnel Strip' section with a list of checkboxes for various tunneling protocols: GTP, GRE, IPinIP, CFP, VXLAN, ERSPAN II, MPLS-in-GRE, MPLS, ERSPAN III, and MPLS-in-UDP. At the bottom of the window, there is a red 'Confirm' button.

MAC	IP	UDP (Port 4789)	VXLAN VNI	MAC	IP	TCP/UDP	Data
-----	----	-----------------	-----------	-----	----	---------	------

# Load-balancing mobile GTP-U traffic



GTP is used in mobile networks to transport packets from the NodeB to the internet via an IP tunnel. Load-balancing could be based on outer IP Addresses which are the IP Addresses of the eNodeBs and SGWs. The problems using the outer tunnel for the hash-key calculation are:

- Limited IPs → **Asymmetric Load** - Few IP addresses can cause uneven traffic distribution.
- When the outer tunnel IP changes (e.g. due to eNodeB change), the hash key result changes, and the session shifts to a different output port. This breaks session continuity from the user perspective, making **load-balancing non-session-aware** and increasing the processing effort needed for correlation and call analysis.

**Solution:** Use inner IP Address = user IP

# Load-balancing to multiple groups

Egress Configuration

Egress Port: C11 x C12 x C13 x C14 x C15 x C16 x C17 x

Port Config

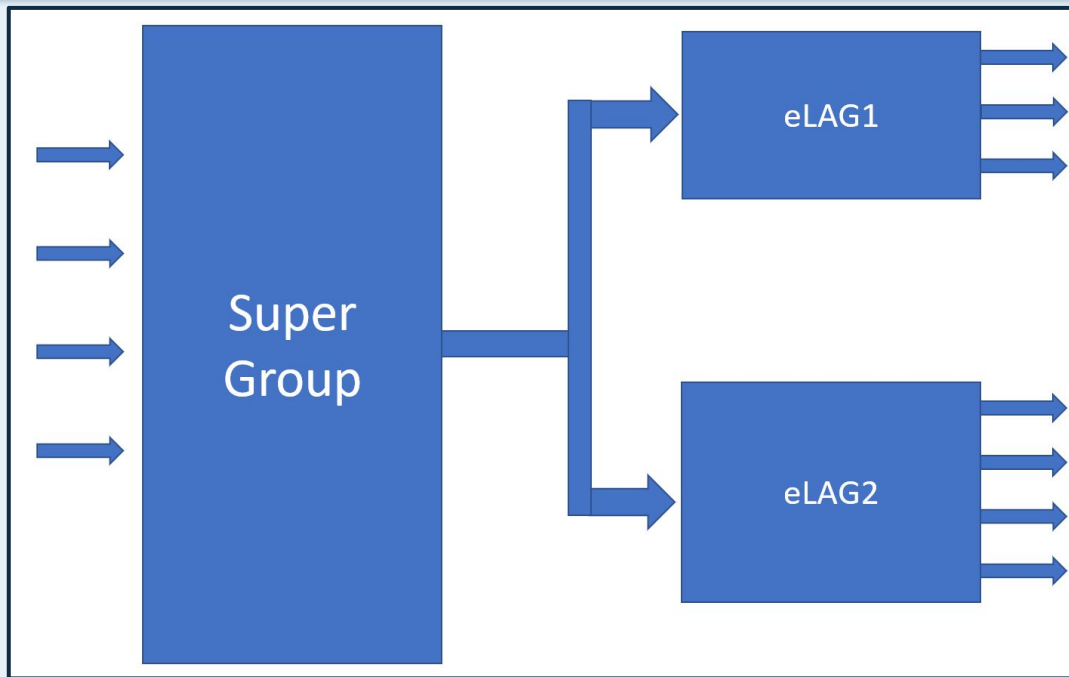
Egress Type: ☐ Copy ☐ Load Balance ☒ Super Group

The traffic that matches the rule will be copied to multiple interface groups for load balancing processing

Load Balance Group: C11 x C12 x C13 x - +

Load Balance Group: C14 x C15 x C16 x C17 x - +

Cancel Confirm



Distribute traffic to parallel analytic tools



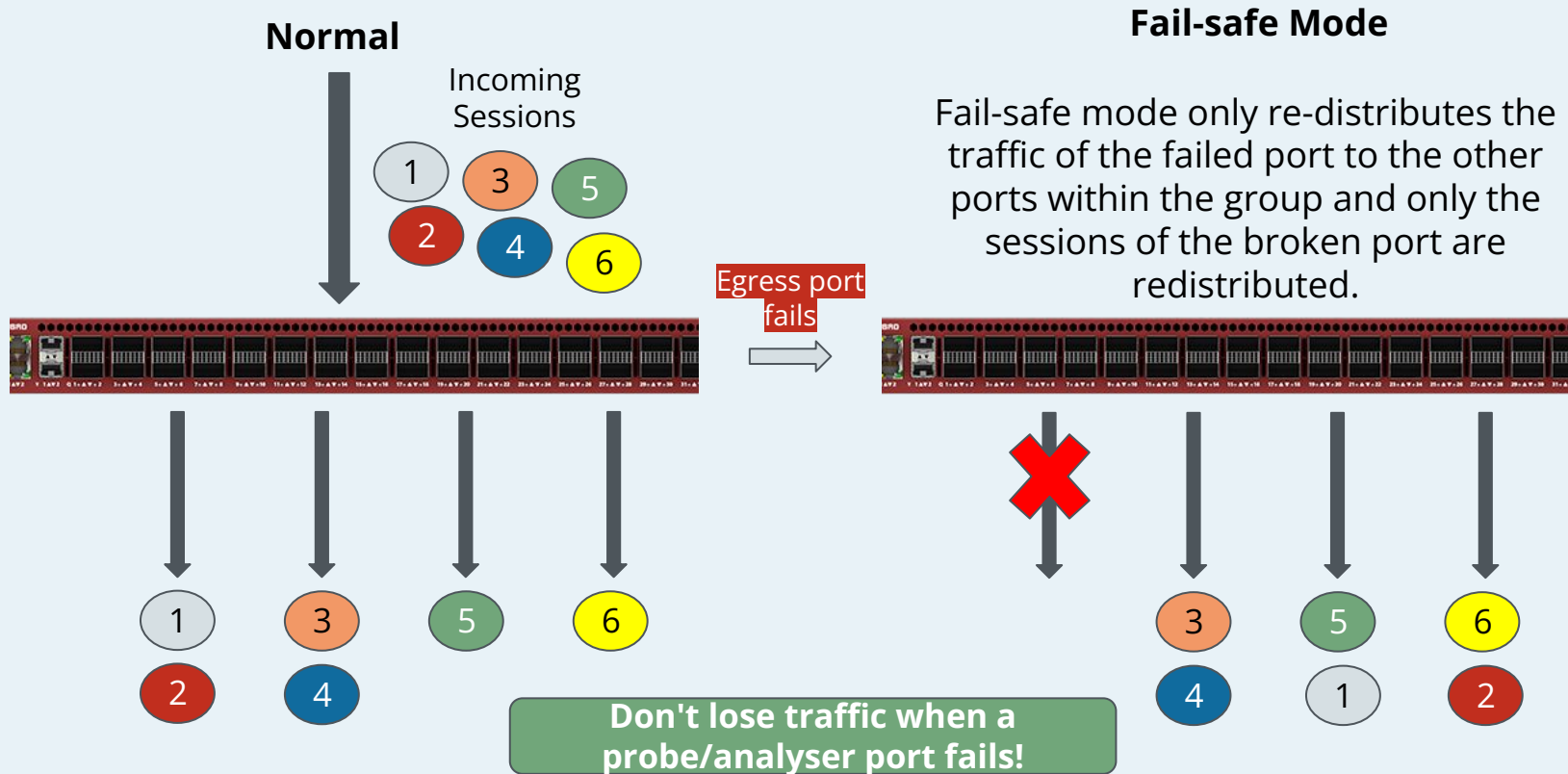
# Traffic handling when an output ports fails

Cubro's Advanced NPBs support different types of load-balancing modes to protect against port failures.



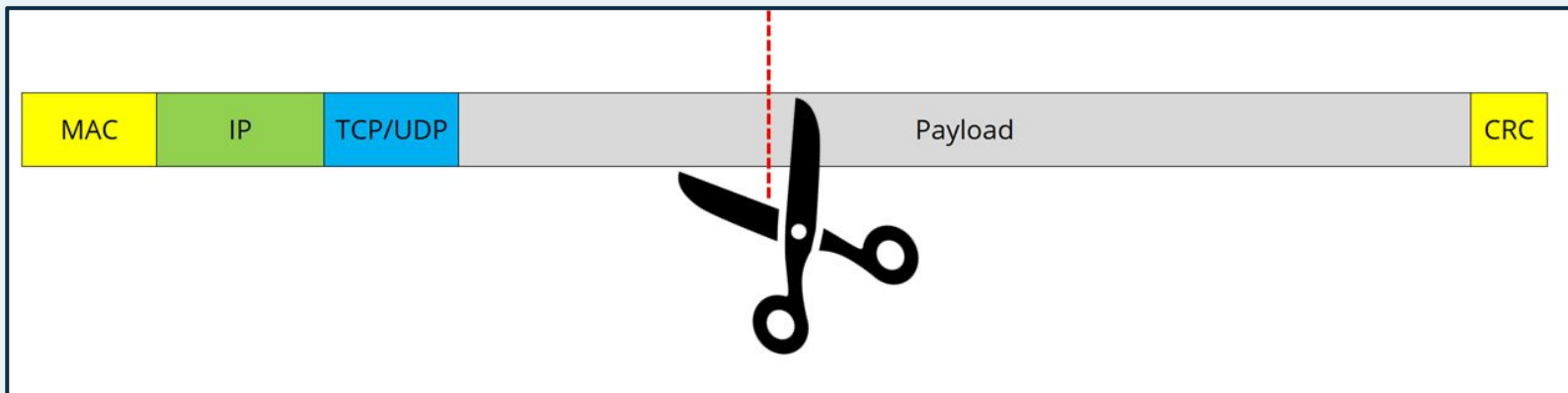
**Don't lose traffic when a probe/analyser port fails**

# Fail-safe Load Balancing



# Slicing for any packet size to reduce output bandwidth

- Cubro G5+ Advanced NPBs allow to set the slicing size to **any value between 64B and 9192 Byte**.
- FCS is automatically corrected; all other fields inside the packet stay unchanged.



Reduces the output bandwidth sent to analytics and probing by removing parts of a packet that are not needed.

# Active Tunnel Endpoint and encapsulated GRE output

- Replies to incoming ARPs and Pings
- Every port with its **own** IP Address & MAC Address

GRE Tunnel Port

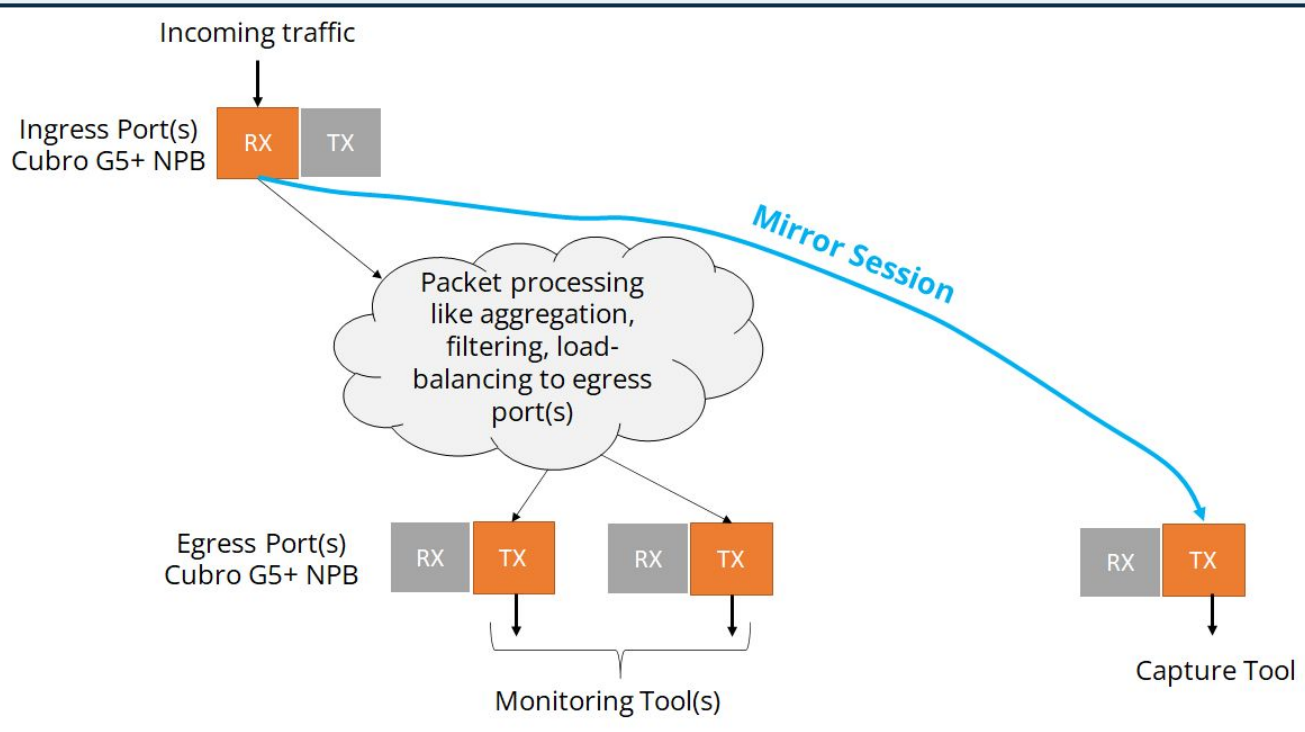
GRE Tunnel Port

☒ Display/Hide Columns   ID

<input type="checkbox"/>	ID ↕	Port ID	Local MAC	Remote MAC	Local IP	Remote IP
<input type="checkbox"/>	2	C2 ▼	00:00:00:00:00:02	00:00:00:00:00:11	10.0.0.2	100.100.100.100
<input type="checkbox"/>	1	C1 ▼	00:00:00:00:00:01	00:00:00:00:00:11	10.0.0.1	100.100.100.100



# Mirror function to easily add an output port for troubleshooting purposes



- Mirror RX port to output
- Mirror TX port to output
- Reduce mirrored output via filtering to reduce traffic load

# Easy output port redundancy

Allows to define spare port for any output port. When main output port fails, traffic is moved to backup port within **milliseconds**.

Also possible for complete load-balancing groups.

Port

Display/Hide Columns

Add

Delete

Spare▼

Cancel Choose All

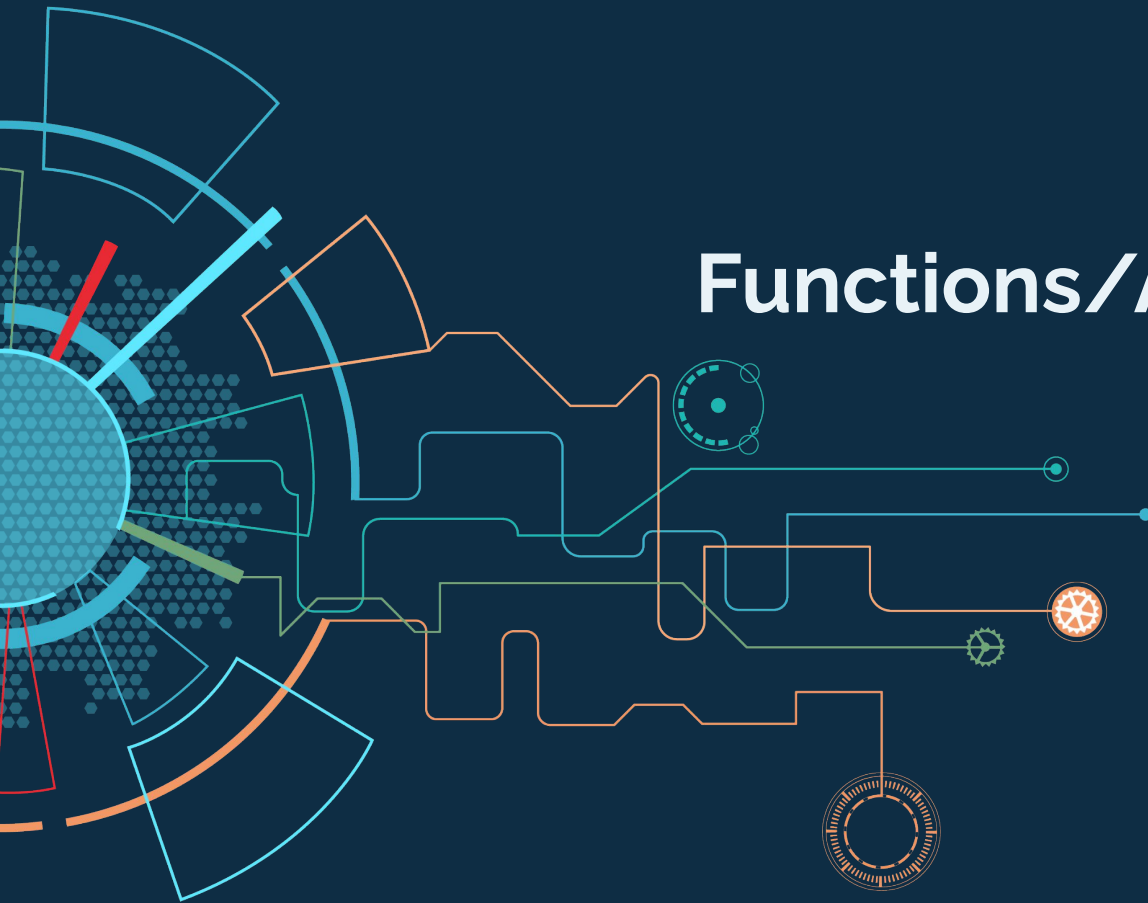
<input checked="" type="checkbox"/>	Port	Spare	Spare Work Type	Linkages
<input checked="" type="checkbox"/>	<div>QC10</div>	<div>Port: QC11</div>	<div></div>	<div></div>

<

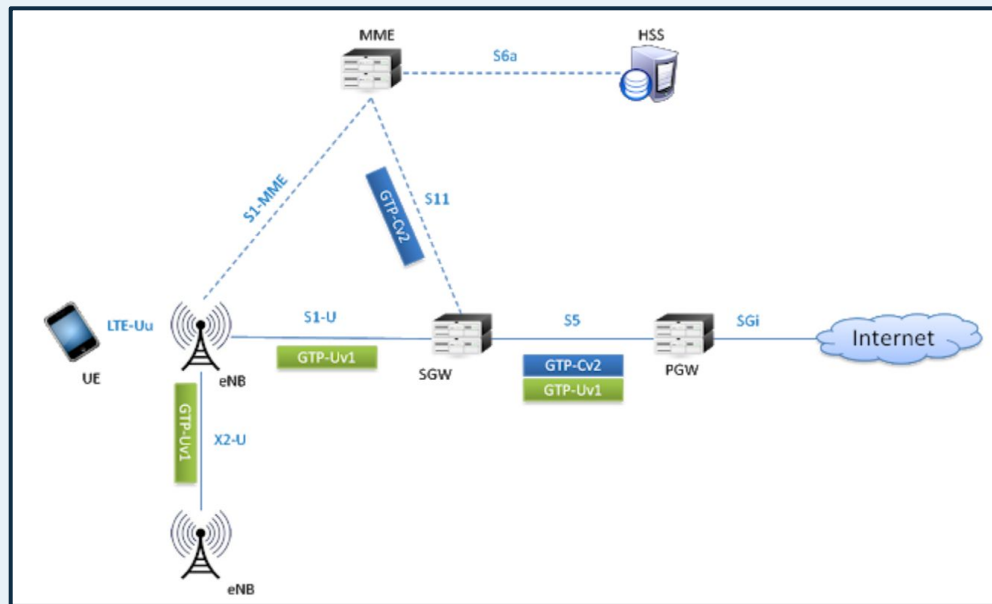
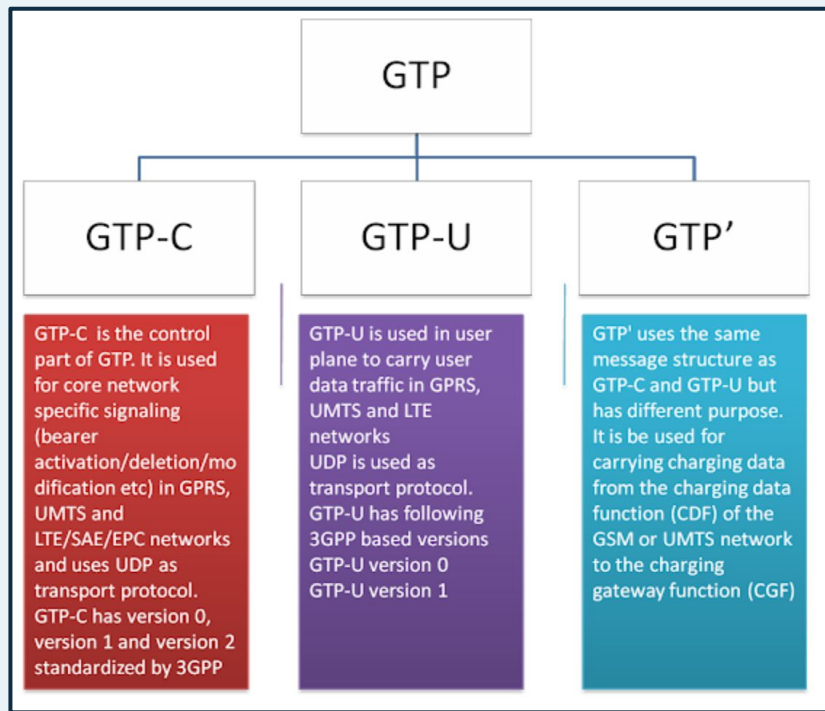
1

>

# G5+ GTP Functions/Applications



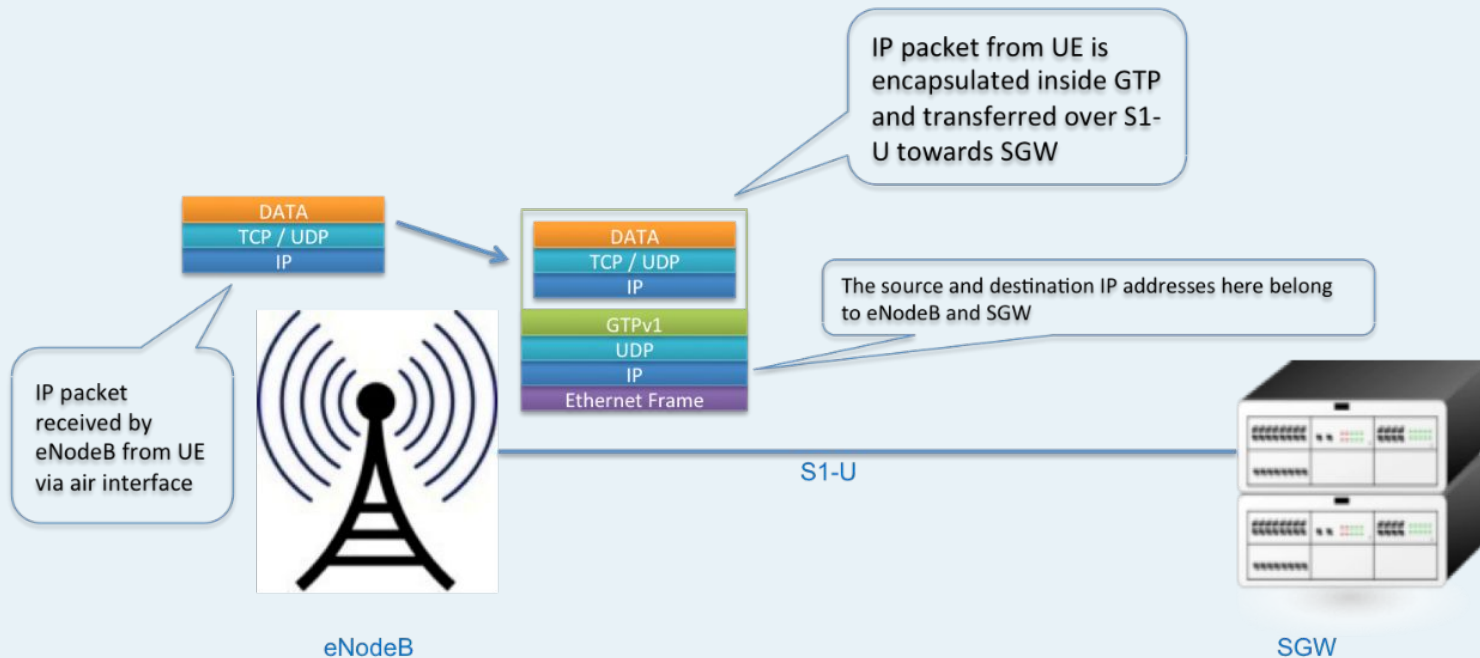
## GTP = GPRS Tunneling Protocol





# GTP-U Overview

*GTP is used to transport packet data from the eNodeB to the SGW via an IP tunnel.*



# Difference between Control Plane and User Plane

GTP-U = is the user-plane (where the user traffic is transported)

```

> Frame 3: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits)
> Ethernet II, Src: Azurewav_ce:5d:f9 (00:25:d3:ce:5d:f9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 212.129.65.23, Dst: 212.129.65.81
> User Datagram Protocol, Src Port: gtp-user (2152), Dst Port: gtp-user (2152)
> GPRS Tunneling Protocol
> Internet Protocol Version 4, Src: 192.168.111.20, Dst: 192.168.111.255 GTP inner IP
> User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137) GTP inner TCP/UDP
> NetBIOS Name Service

```

GTP-C = is the control plane of the protocol; Note that GTP-C does not have an inner IP

```

> Frame 1: 201 bytes on wire (1608 bits), 201 bytes captured (1608 bits)
> Ethernet II, Src: Azurewav_ce:5d:f9 (00:25:d3:ce:5d:f9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 212.129.65.13, Dst: 212.129.65.65
> User Datagram Protocol, Src Port: gtp-control (2123), Dst Port: gtp-control (2123)
> GPRS Tunneling Protocol

```

# Separate 4G / 5G User plane

Usually filtering of user-plane is done via UDP Port 2152 which defines user-plane traffic but UDP Port 2152 is used for 3G, 4G as well as 5G. So filtering on UDP Port 2152 is not the right solution to get only 5G user-plane.

But 5G user-plane traffic is usually using a GTP extension header:

```
> Frame 6: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface \Device\NPF_{9437CF42-1780-49FF-BA95-450F459B1FA5}, id 0
> Ethernet II, Src: HuaweiTe_b8:81:35 (24:44:27:b8:81:35), Dst: zte_21:14:20 (d4:c1:c8:21:14:20)
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 2
> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 2
> Internet Protocol Version 4, Src: 172.30.81.159, Dst: 172.20.174.58
> User Datagram Protocol, Src Port: 2152, Dst Port: 2152
> GPRS Tunneling Protocol
  > Flags: 0x36
    Message Type: T-PDU (0xff)
    Length: 60
    TEID: 0x004abcd7 (4898007)
    Sequence number: 0x66d9 (26329)
    Next extension header type: PDU Session container (0x85)
  > Extension header (PDU Session container)
    > PDU Session Container
      > Next extension header type: Internet Protocol Version 4
    > Internet Protocol Version 4
      > Transmission Control Protocol
```

With the string filter function the G5+ series is perfectly suited to separate 5G from 3G/4G user-plane.

```
> Internet Protocol Version 4, Src: 172.30.81.159, Dst: 172.20.174.58
> User Datagram Protocol, Src Port: 2152, Dst Port: 2152
> GPRS Tunneling Protocol
  > Flags: 0x36
    Message Type: T-PDU (0xff)
    Length: 60
    TEID: 0x004abcd7 (4898007)
    Sequence number: 0x66d9 (26329)
    Next extension header type: PDU Session container (0x85)
  > Extension header (PDU Session container)
    > Internet Protocol Version 4, Src: 10.165.242.222, Dst: 213.94.75.78
```

```
0000 d4 c1 c8 21 14 20 24 44 27 b8 81 35 81 00 00 02 ...I..$0^..5...
0010 81 00 00 02 08 00 45 58 00 60 03 c5 00 00 1f 11 ...-EX^.....
0020 3f 64 ac 1a 51 9f ac 14 ae 3a 08 68 08 68 00 4c ?d-Q...:h-h-L
0030 00 00 36 ff 00 3c 00 4a bc d7 66 d9 00 35 01 10 ..6-<J...f..
0040 01 00 45 00 00 34 aa 34 40 00 40 06 72 5f 0a a5 ..E..4.4@@.P..
0050 f2 de d5 5e 4b 4e b7 4e 01 bb 3d b8 28 bb 7a 13 ..AKN-N...(:z
0060 40 27 80 10 3f d6 d0 03 00 00 01 01 08 0a 14 af @'..?.....
0070 da 16 69 10 17 25 ..1..%
```

Rule Configuration

WildcardAccurateMACString

Add RuleDisplay/Hide Columns

	Rule ID	Ace Type	Filter Key		Handle		Tunnel Encapsulation		Modify Source M
			Offset	Filter Value	Action	VLAN ID	Tunnel Type	Tunnel ID	
	60001	string	11	0x85	none				

> Extension head

> Internet Protoc

0000d4 c1 c8 21 14 20 24 44 27

001081 00 00 02 08 05 45 58 00

00203f 64 ac 1e 51 3f ac 14 ae

003000 00 56 ff 09 3c 00 4a 7c

004001 00 45 08 00 34 33 34 00

0050f2 de d5 5e 4b 4e b7 4e 01

006040 27 80 10 3f de 00 03 00

0070da 16 69 10 17 25

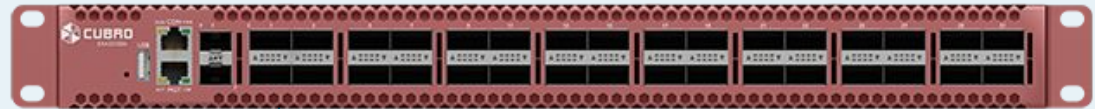
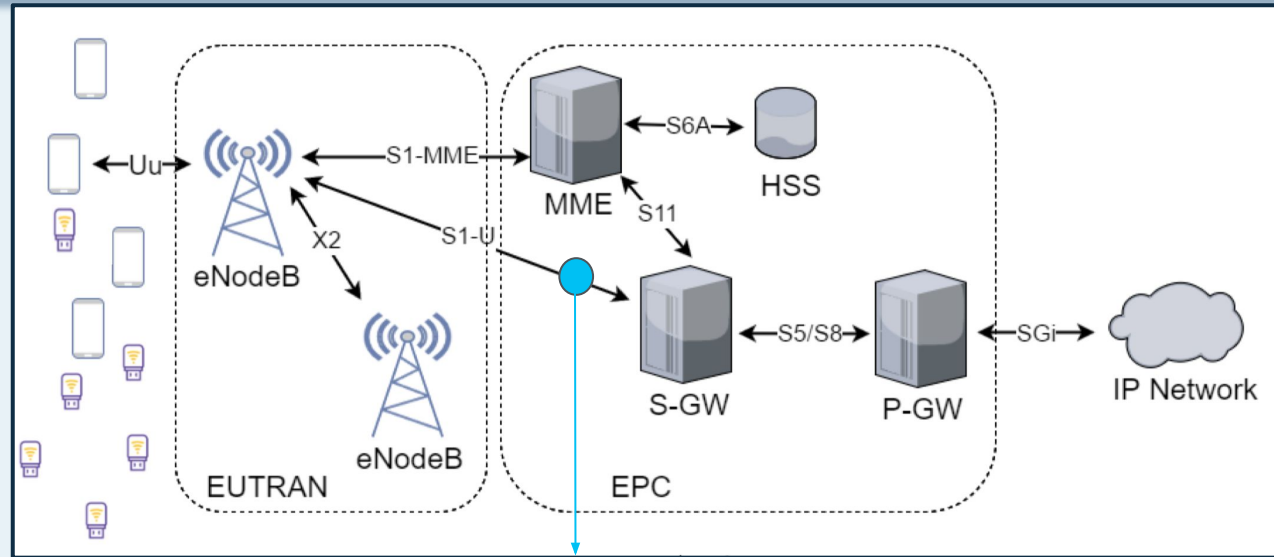
- GTP-U tunnel termination
  - Remove GTP-U tunnel header
- GTP-U Inner IP filtering including IP range filtering
  - Drop traffic by simple inner IP filtering to avoid overload on monitoring probes
- GTP-U Inner Layer 4 (application) filtering
  - Filter directly on S1-U interface and feed the traffic to the right monitoring system
- GTP-U load-balancing
  - Balance output traffic to probes by means of inner IP address

**All in full line-speed without throughput restrictions**

# GTP Inner IP Range Filtering

Reduce the load to the monitoring probes by dropping non required traffic.

Filter on GTP inner IP Address range to drop traffic from/to LTE modes.

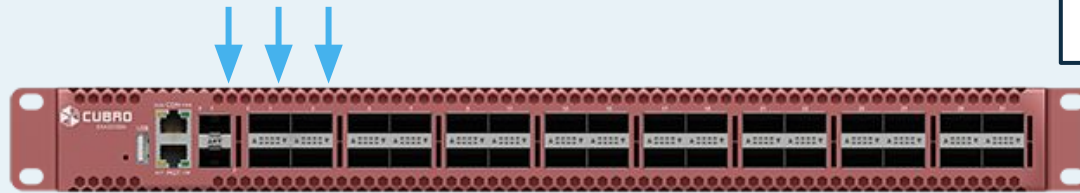


# Application filtering inside GTP Tunnel

Cubro G5+ allows direct access to application information inside GTP by using GTP inner UDP filtering. - e.g. DNS.

This is a simple and scalable solution to offload irrelevant traffic from the probes and thus saves costs.

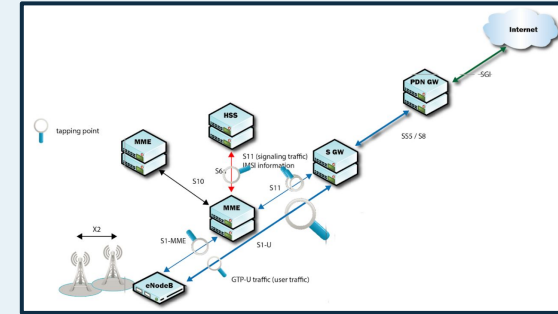
n x 100G (S1-U and S11)



Load-balanced  
User traffic (incl. DNS)

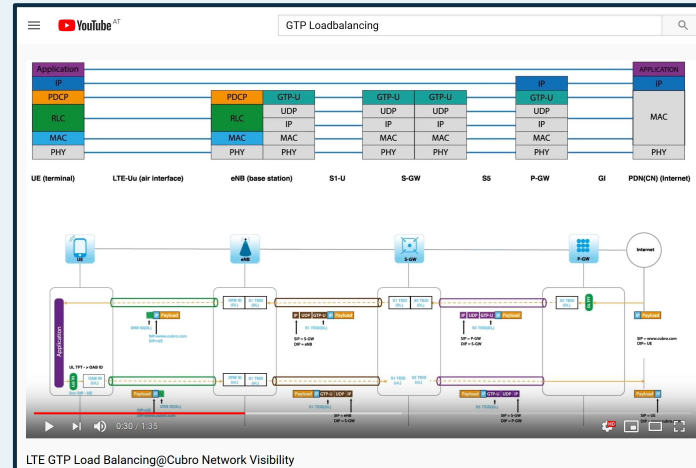
All signalling traffic

All DNS Traffic





- Usually the S1-U interface is the most loaded on a mobile network.
- To distribute user-plane S1-U traffic to various probes is of key importance.
- Session-aware load-balancing from UE point of view is critical. Check our YouTube on GTP Load-balancing <https://youtu.be/4UXhaxi1OMw>
- Cubro G5 series handles GTP load-balancing in hardware to support Tbit/s processing power.



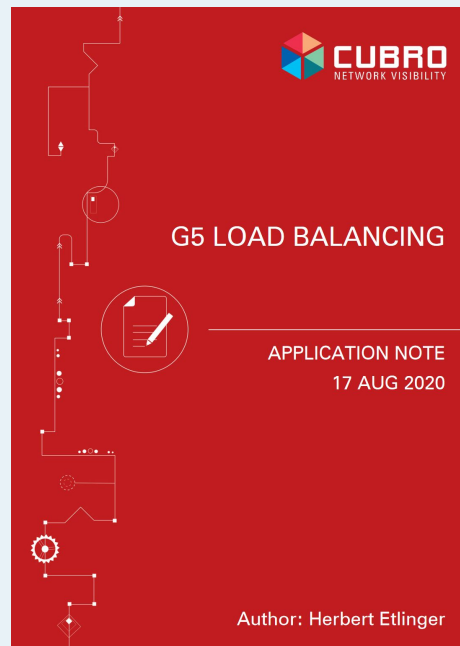
# Load-balancing - some more details

## Hash-key calculation

To cope with a wide range of requirements, the EXA48600 & EXA32100 allow various methods to calculate the hash-key. Hash-keys are used to define the load-balancing behaviour among the various members in the load-balancing group. For example, if the hash-key is configured as “IP Source Address”, the hashing would be performed based on the source IP address of the packet only. Therefore, all packets with the same source IP address will be available at the same physical output port. The EXA48600 and EXA32100 support following hash-key calculation methods:

Hash-key calculation method	Hash-key calculation based on	Remark
I3-src	full IP Src Addr	
I3-dst	full IP Dst Addr	
I3-src-dst	full IP Src & IP Dst Addr	
I4-src-dst	full Layer 4 Src & Dst Port	
four-tuple	full IP Src & Dst Addr & Layer 4 Src & Dst Port	Upstream & downstream direction give DIFFERENT hash results -> upstream & downstream is split apart -> not session aware E.g. 10.0.0.1 talks to 10.0.0.2: Hash result = x 10.0.0.2 talks back to 10.0.0.1: Hash result = y
four-tuple-m8	middle 8 Byte of IP Src & Dst Addr & full Layer 4 Src & Dst Port	
five-tuple	full IP Src & Dst Addr & Layer Protocol & Layer 4 Src & Dst Port	
I3-src-dst-m8	middle 8 Byte IP Src & IP Dst Addr	
five-tuple-m8	middle 8 Byte IP Src & Dst Addr & Layer Protocol & Layer 4 Src & Dst Port	
I3-src-dst-symmetric	full IP Src & IP Dst Addr	Upstream & downstream direction give SAME hash results -> upstream & downstream stay together-> session aware E.g. 10.0.0.1 talks to 10.0.0.2: Hash result = x 10.0.0.2 talks back to 10.0.0.1: Hash result = x
I4-src-dst-symmetric	full Layer 4 Src & Dst Port	
four-tuple-symmetric	full IP Src & Dst Addr & Layer 4 Src & Dst Port	
four-tuple-m8-symmetric	middle 8 Byte of IP Src & Dst Addr & full Layer 4 Src & Dst Port	
five-tuple-symmetric	full IP Src & Dst Addr & Layer Protocol & Layer 4 Src & Dst Port	
I3-src-dst-m8-symmetric	middle 8 Byte IP Src & IP Dst Addr	
five-tuple-m8-symmetric	middle 8 Byte IP Src & Dst Addr & Layer Protocol & Layer 4 Src & Dst Port	

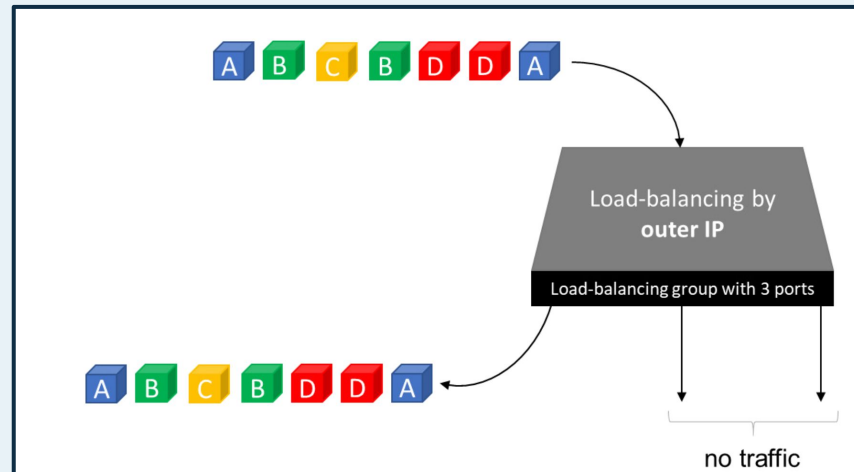
Check the application note below to find detailed information on how load-balancing works.



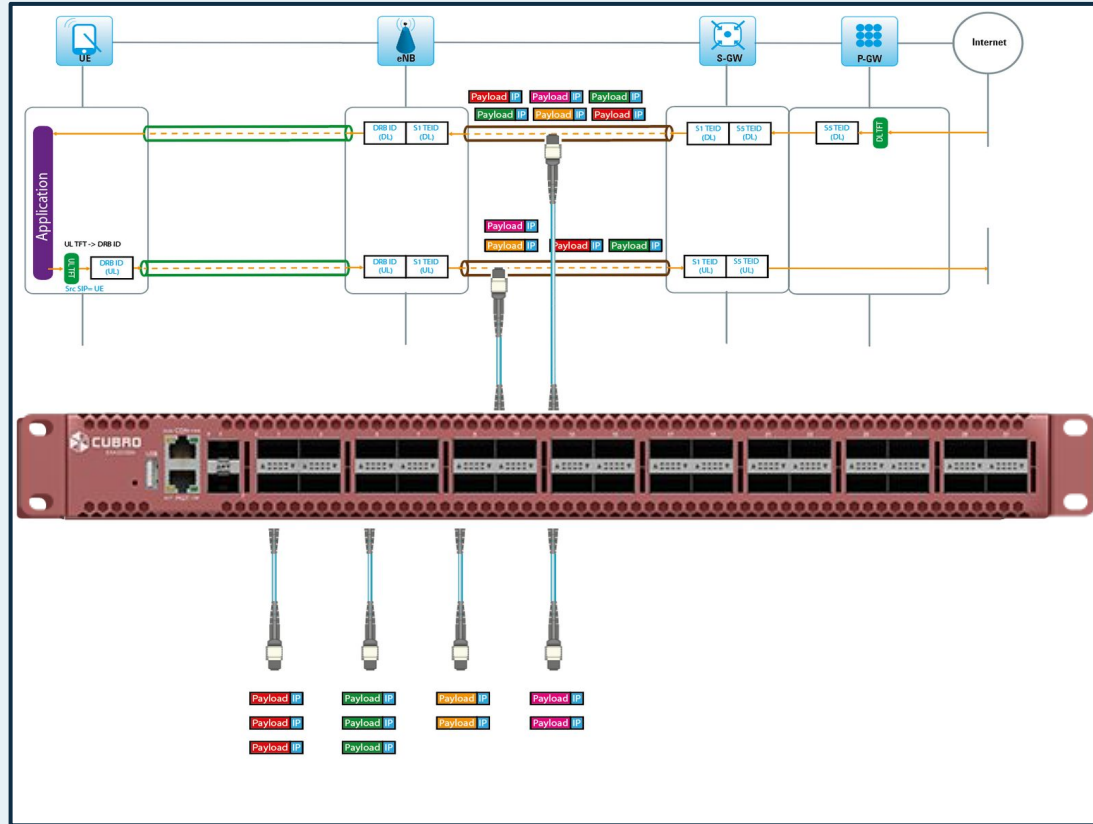


# Problems caused by using outer IP for load-balancing

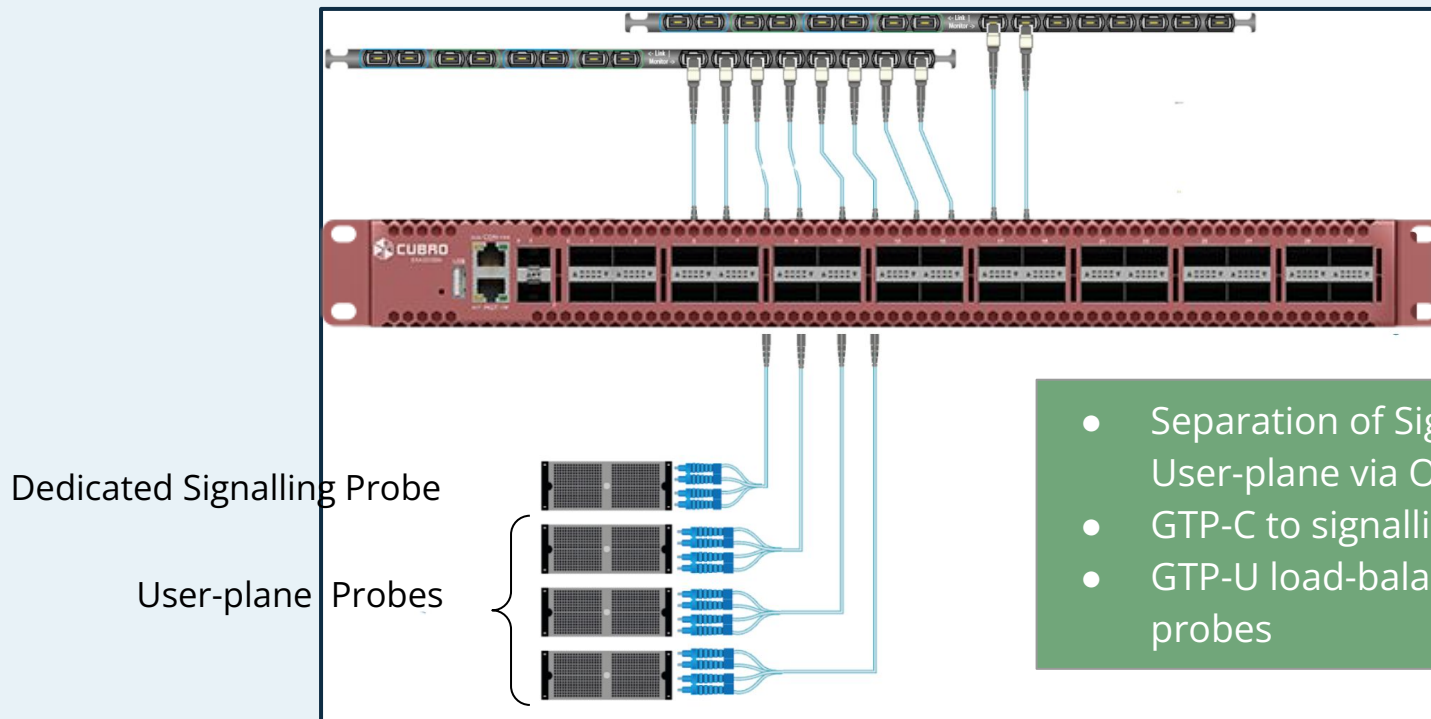
- The monitoring session for a user will be interrupted when the customer is moving to another location.
- Due to the small amount of outer IPs, the load-balancing could be asymmetric. This means the output ports can be overloaded which causes packet drop and thus bad monitoring quality.



# Solution - Load-balancing by means of GTP inner IP



# Mobile traffic monitoring - full picture



- Separation of Signalling and User-plane via Outer UDP filtering.
- GTP-C to signalling probe only
- GTP-U load-balanced to user-plane probes

Cubro G5 plus is by far the most complete Advanced NPB for 400G applications available. It offers state-the-art functionality to cope with the widest range of applications.

- High port-density and high throughput applications like seen in mobile telecom environments
- Overlay network traffic handling such as tunnel removal, inner tunnel filtering and load-balancing
- Traffic aggregation and filtering
- Break-out to existing 10G, 25G, 50G and 100G equipment

Customer  
Support



Technical  
Capabilities



Value  
Money



Reliability



Flexibility



Customer  
First



Collaboration





We have operations in all time zones.  
Reach us at: [support@cubro.com](mailto:support@cubro.com)