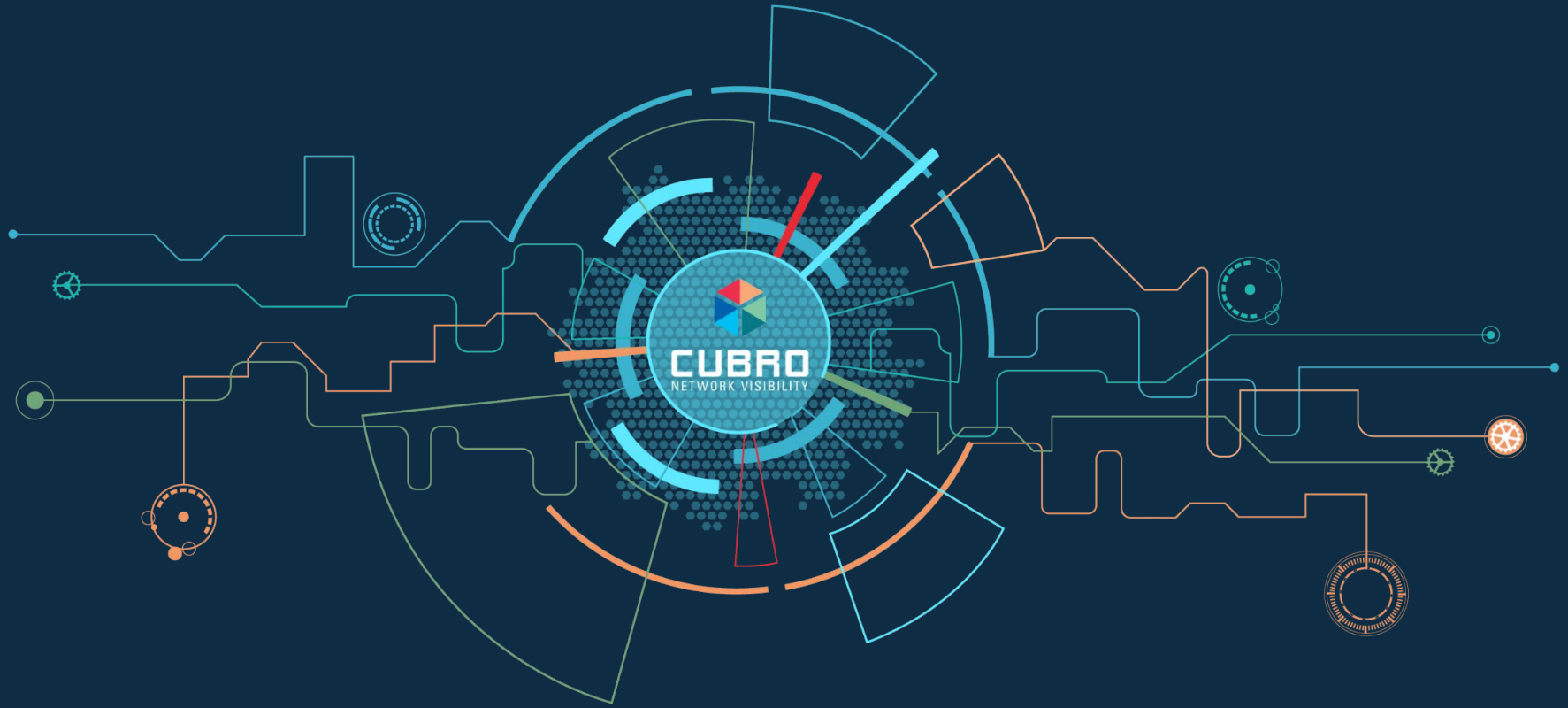
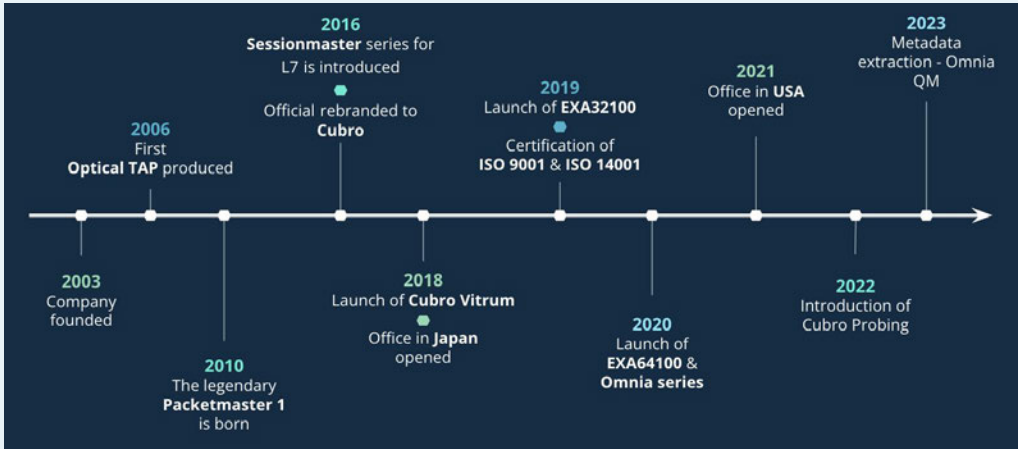


G5 Plus - Advanced Network Packet Broker - Overview

September 2023

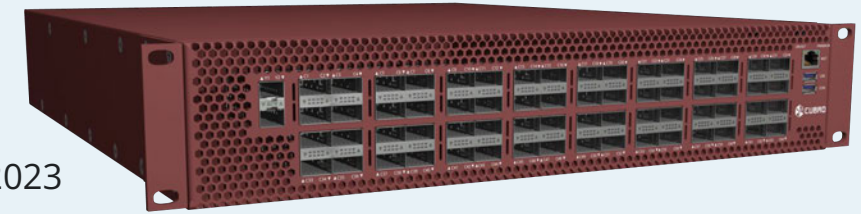
Background

- 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 three 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**; launched in May 2023

G5+ key points:

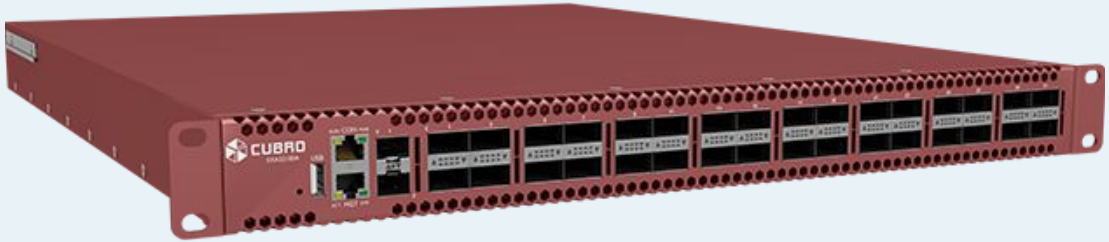
- Tunnel Termination
- State-of-the art VXLAN handling including VNI filtering
- Inner tunnel filtering
- Superior Load-balancing features including inner tunnel hashing
- More than 100k parallel filtering rules

Technical Details

G5 Plus Overview

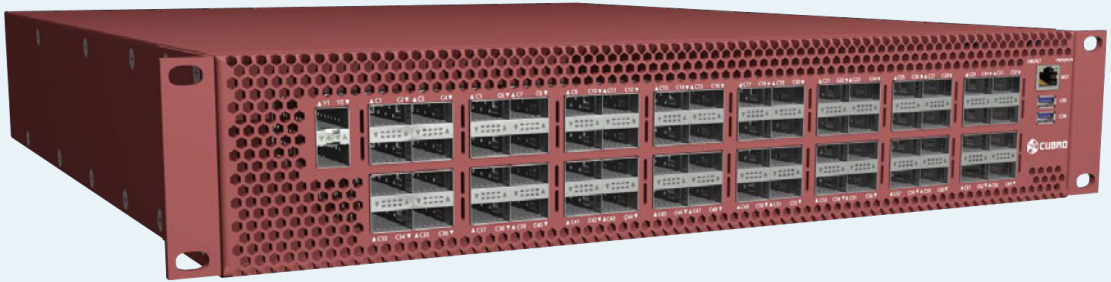
G5 plus is the market leading Advanced Network Packet Broker series. It is based on a state-of-art multi-core, industry-leading programmable switch chip architecture, and it allows all traffic filtering features to be implemented at the hardware level for **unmatched throughput and performance**. The various possibilities to remove tunnel encapsulations from packets and the possibility of inner tunnel filtering make the G5 plus series **ideal for any modern overlay network**.

EXA32100A



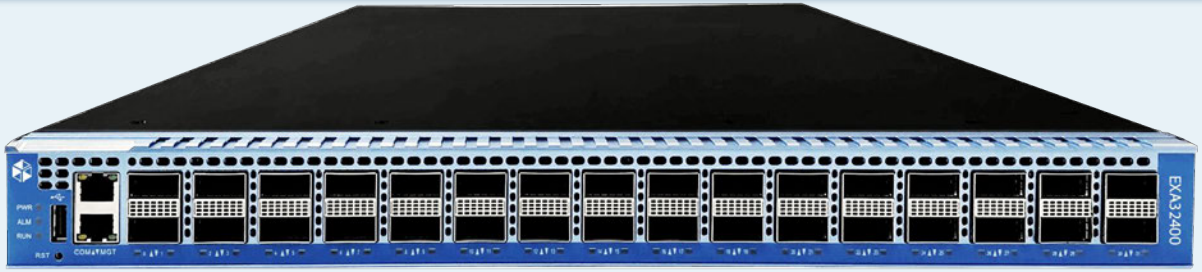
- **32 x 40G/100G** – each of these ports can be used in 4 x 10G or 4 x 25G or 2 x 50G split mode.
- 2 x native SFP+/SFP28 ports for 10G/25G
- Each port can be used simultaneously as input and output and is totally independent of other ports
- Non-blocking architecture
  - 6,5 Tbit/s throughput
  - 2,4B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

EXA64100



- **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

EXA32400



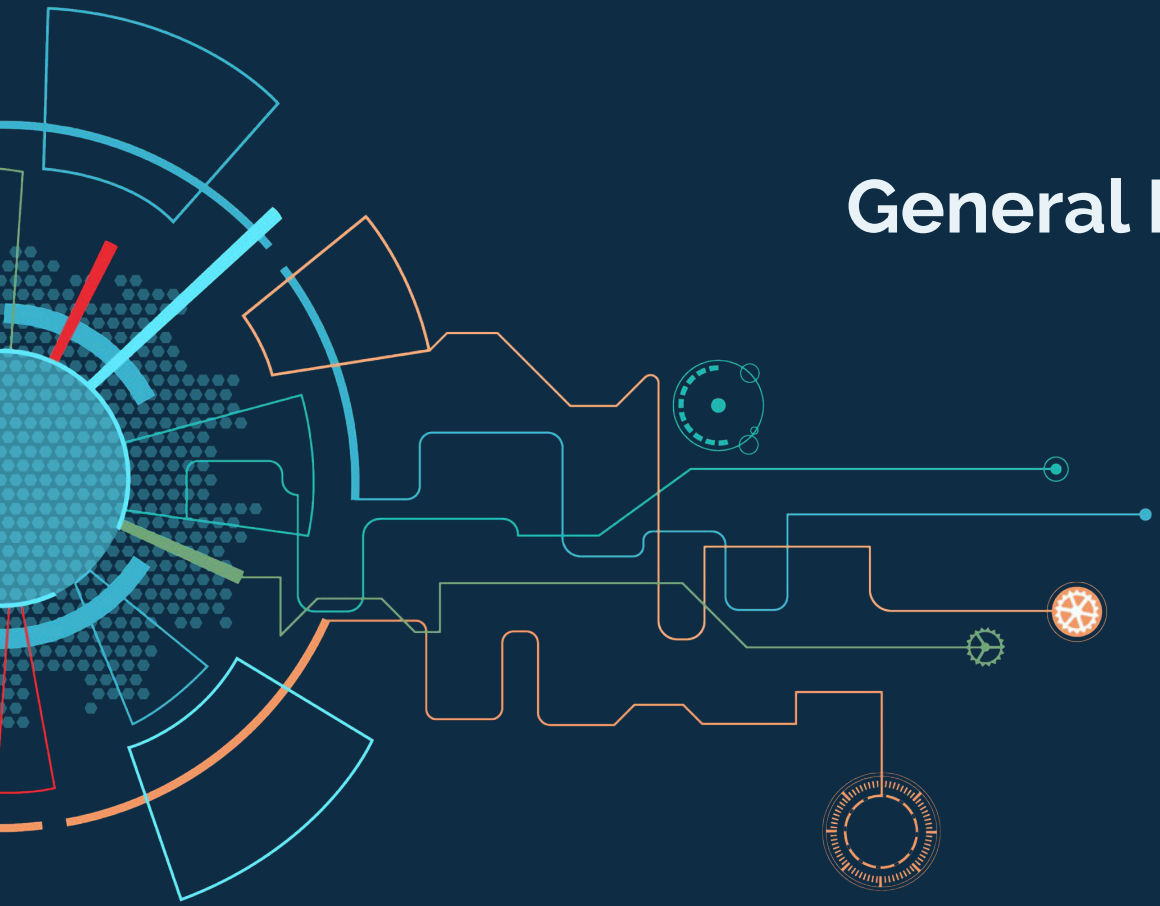
- **32 x 100G/400G** via QSFP28/QSFP-DD
- 128 x 100G when 100G split mode is activated
- Each port can be used simultaneously as input and output and is totally independent of other ports
- Non-blocking architecture
  - 25,6 Tbit/s throughput
  - 6B pps packet forwarding
- All ports are included and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

G5 Plus - Highlights

- 32 x 100G, 64 x 100G and 32 x 400G
- 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

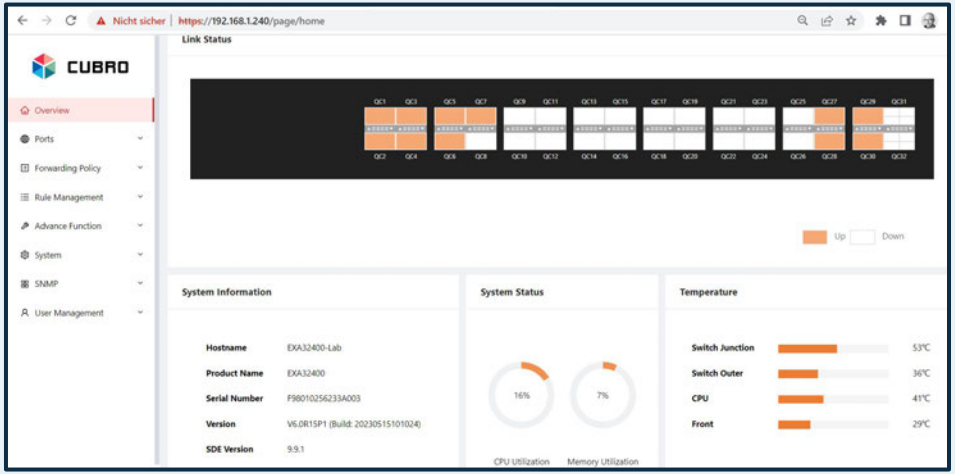
Best in-class Advanced NPB

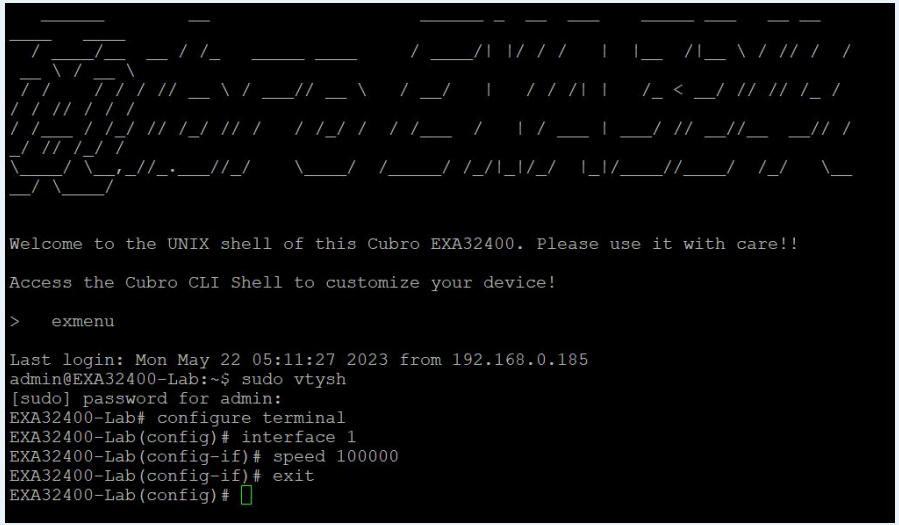


General Features and Functions

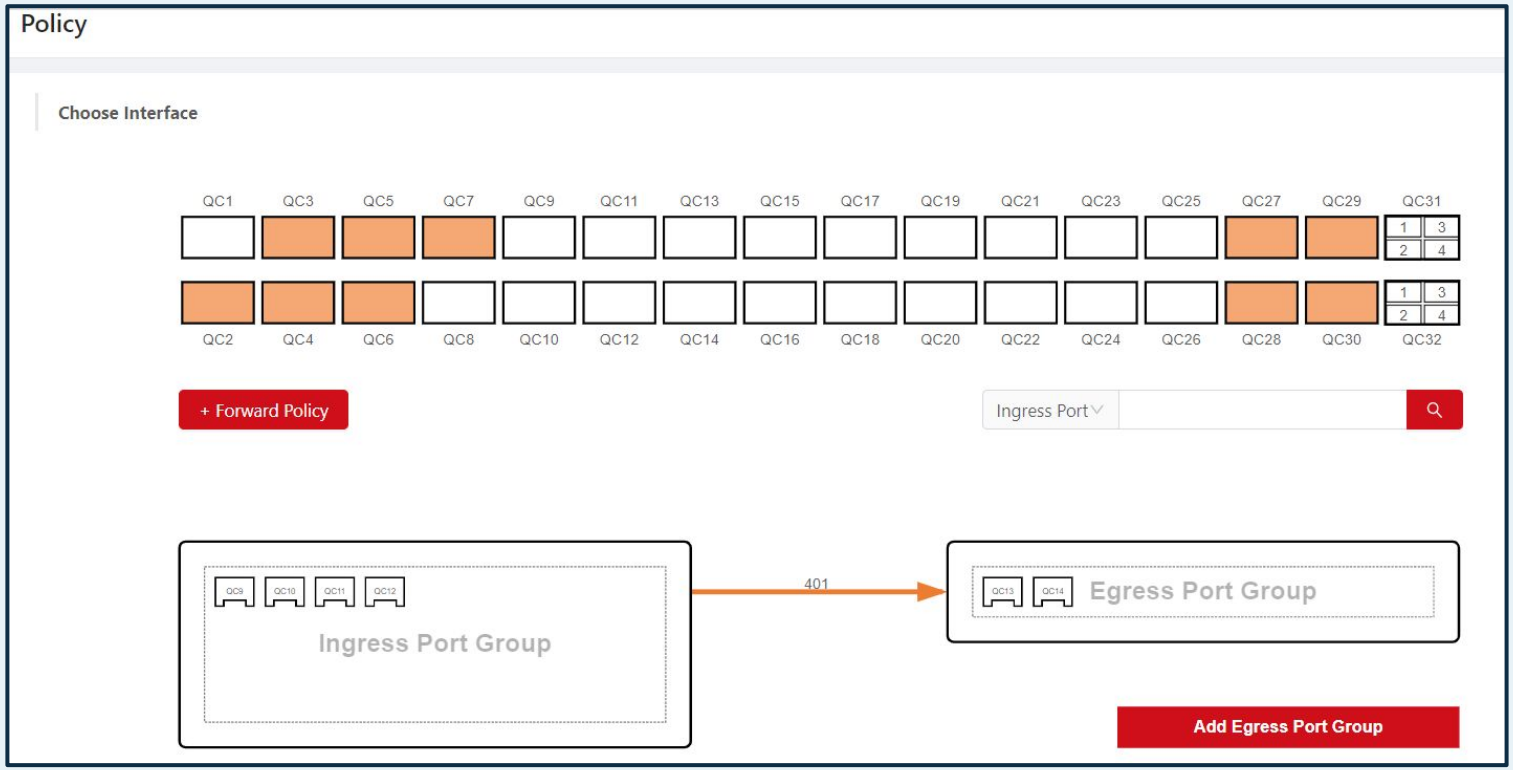
Straight-forward operation via WebUI or CLI

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





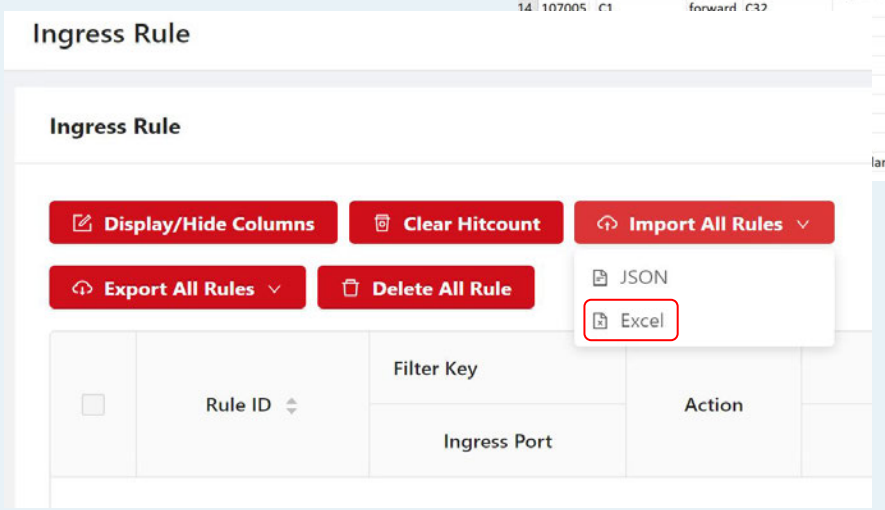
Forwarding Policy via drag & drop



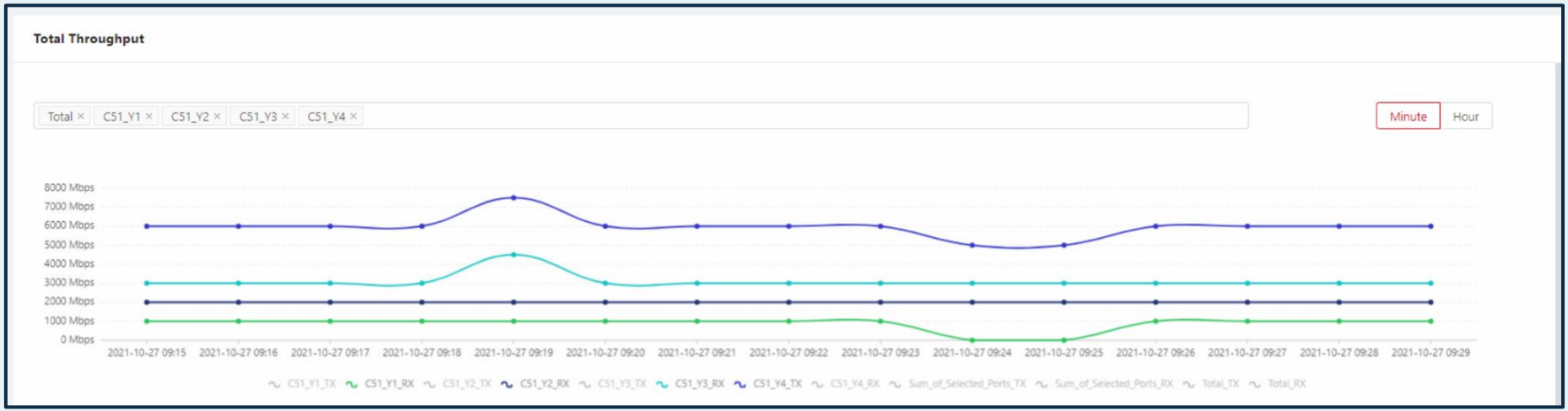
Create filters with MS Excel® & upload to G5plus

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

	A	E	F	G	K	L	M	N	O	P	Q	R	S	T
1	ace_id	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				



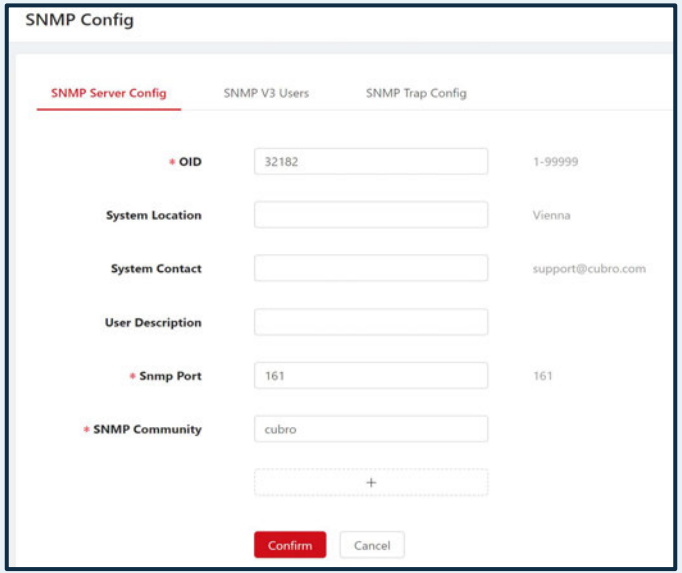
Graphical Throughput per port

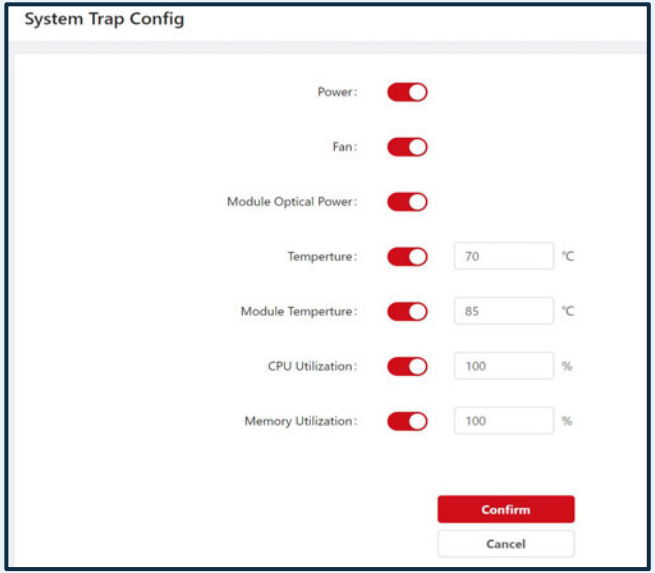


Port utilization over time to visualize traffic trends early.

SNMP management integration and supervision

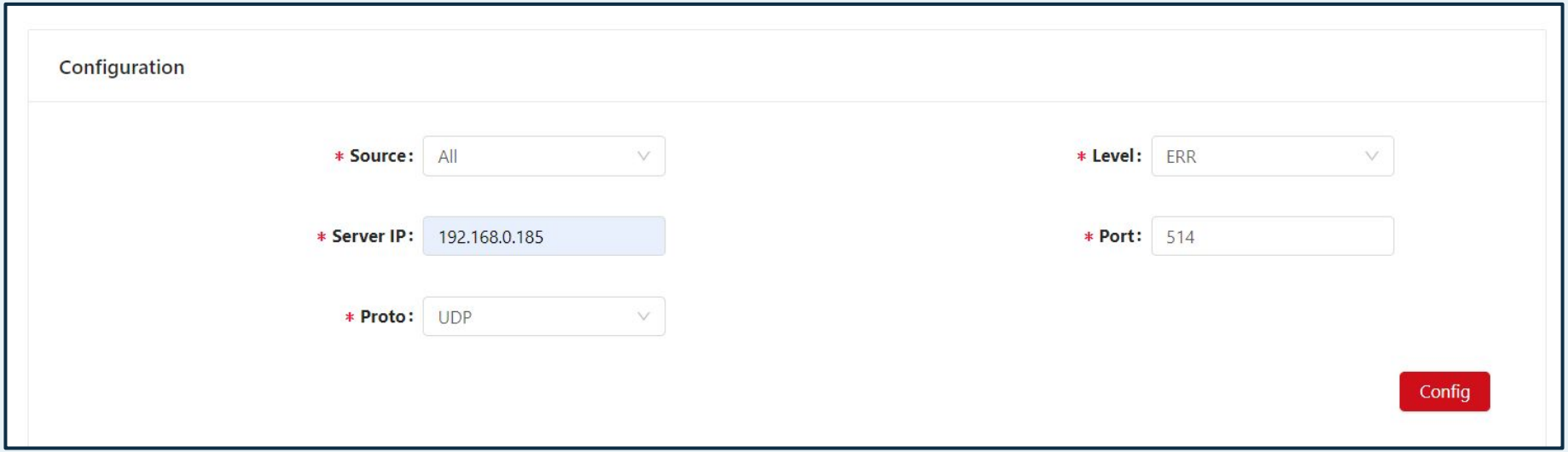
SNMPv2c and v3 is supported and thus G5 plus can be easily integrated into any SNMP supervision system. MIB file is provided by Cubro.





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

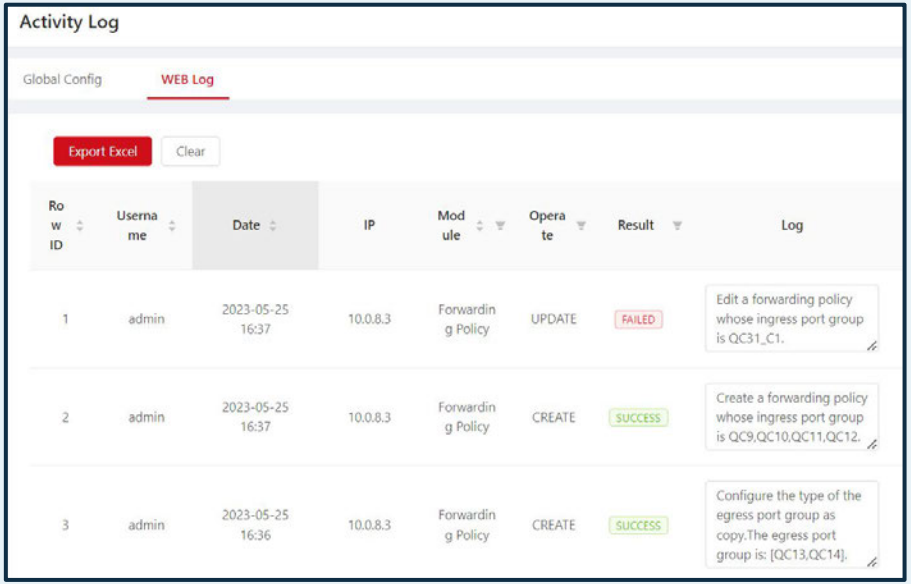
Remote Syslog



Completely user configurable Syslog

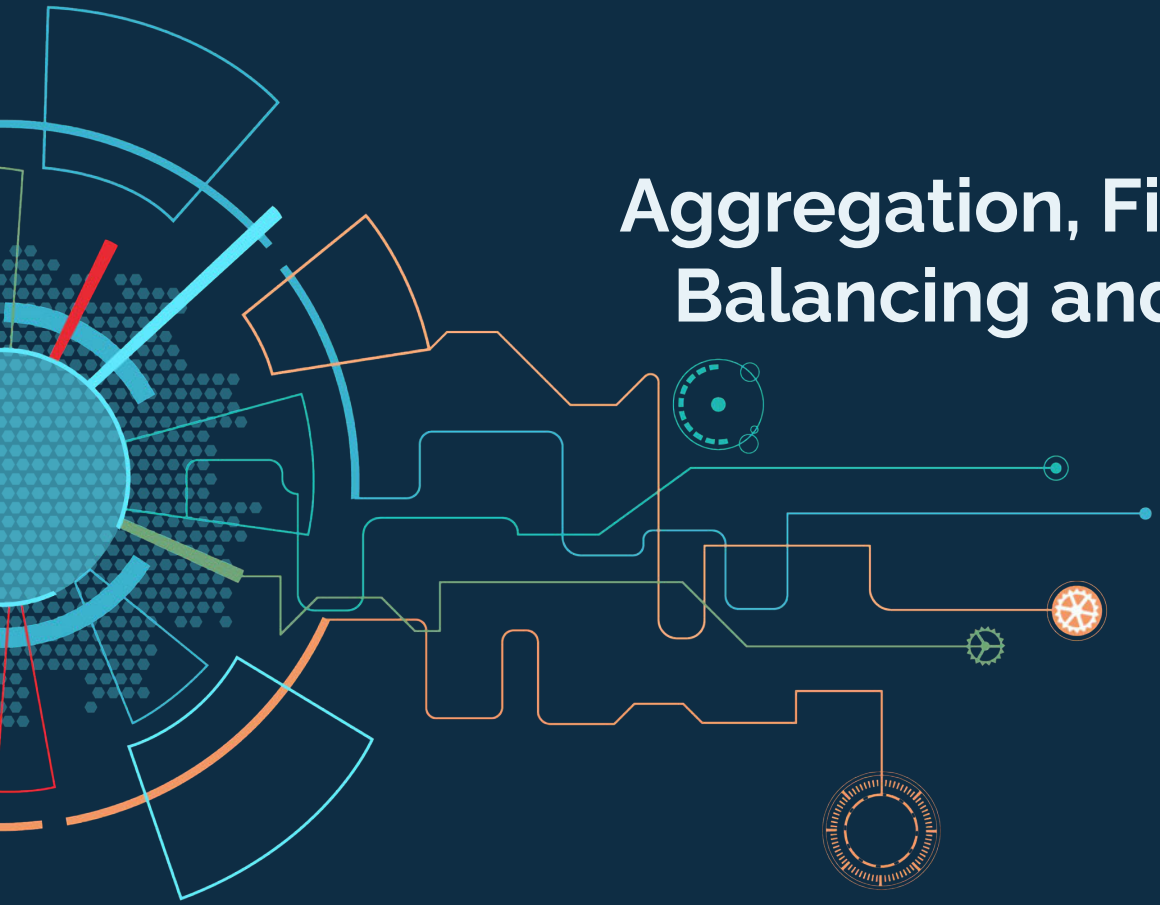
Other platform features

- 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



CUBRO
NETWORK VISIBILITY

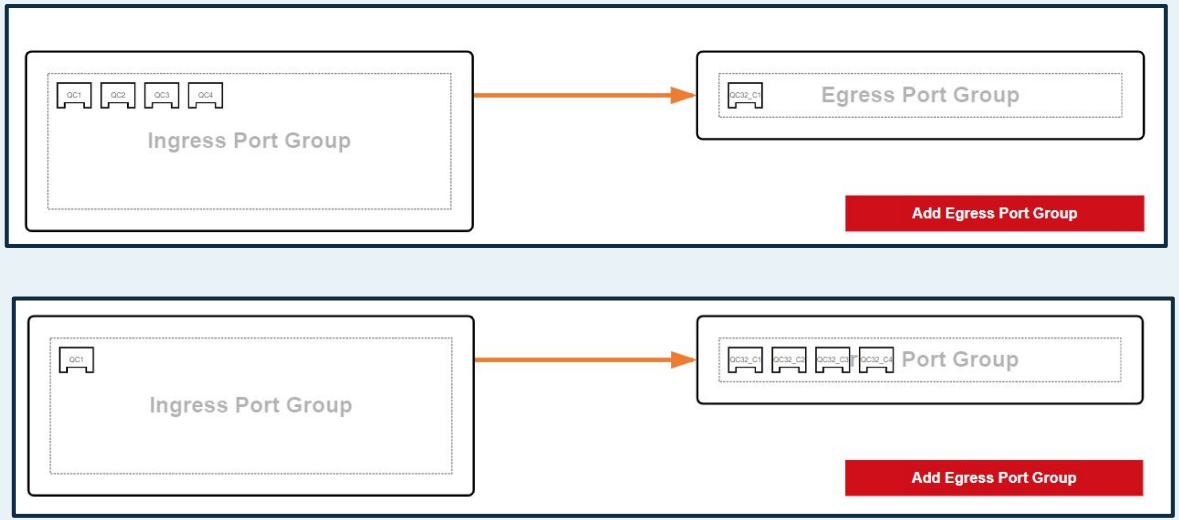
Aggregation, Filtering, Load Balancing and much more



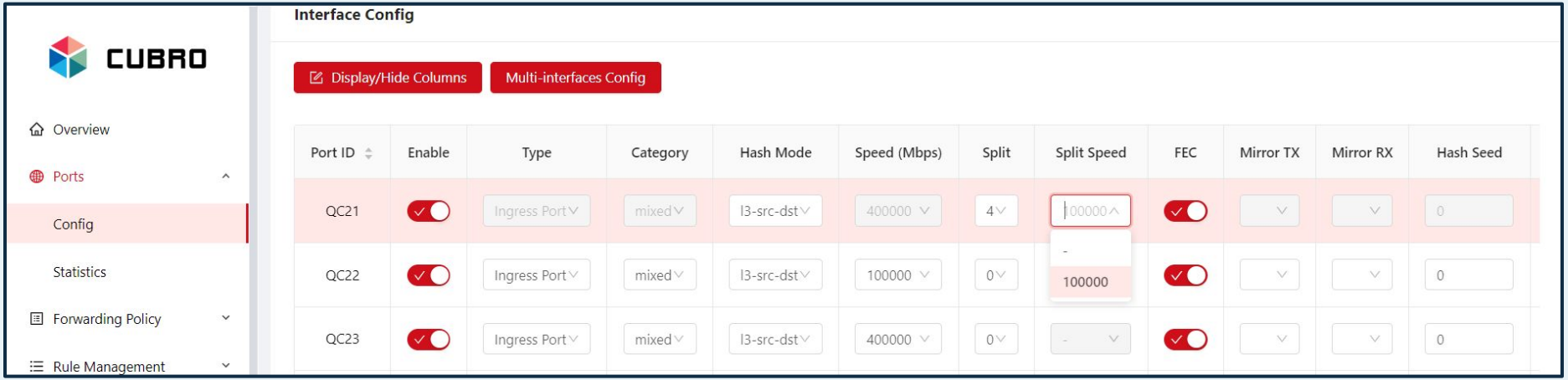
Aggregation

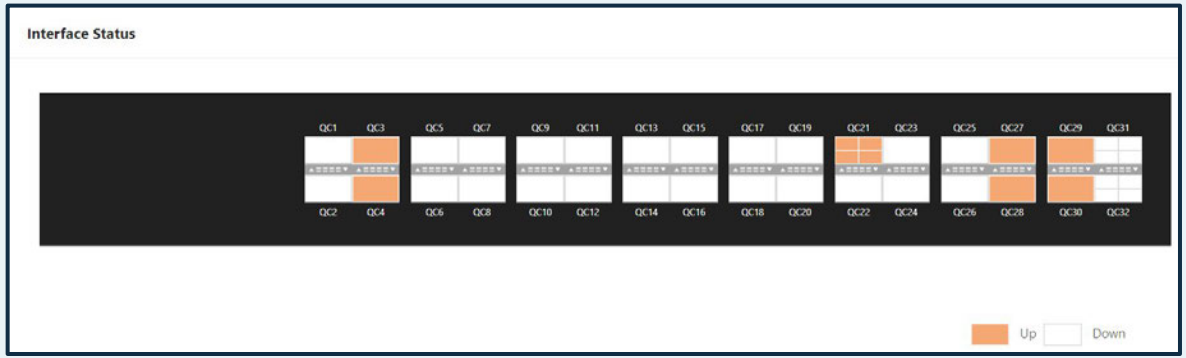
All kinds of aggregation supported :

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



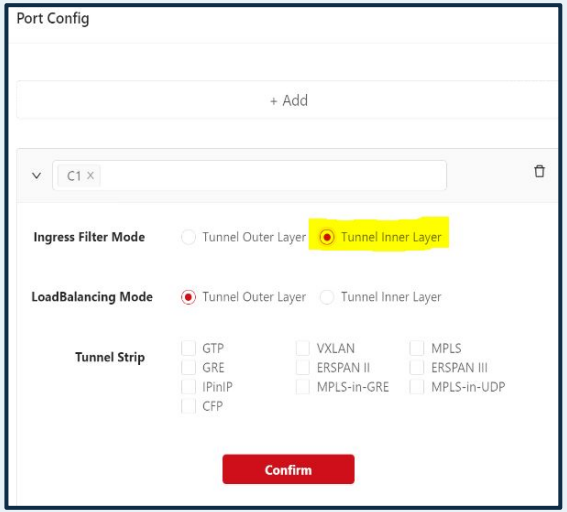
Split mode - E.g. 400G into 4 x 100G





Filtering parameters

- 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



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

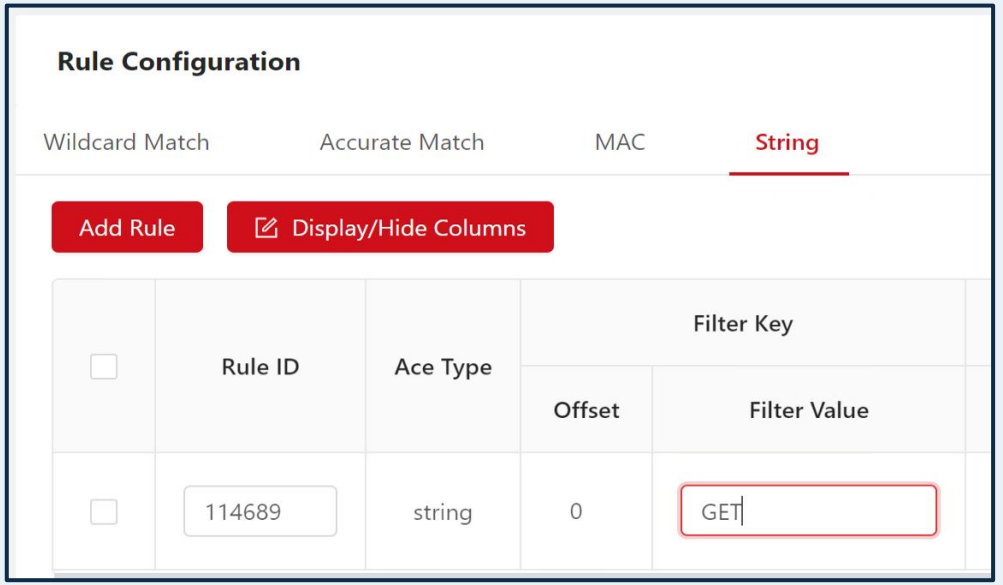
Feed only relevant traffic to the probe/analyzers

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

ASCII string / Hex pattern filtering inside payload

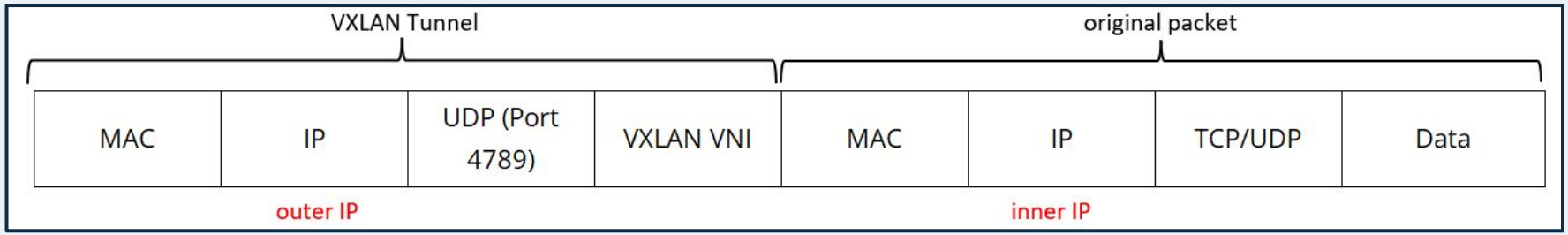
- Filter not only on packet header fields like MAC Address, IP Address or TCP/UPD port numbers but also inside the payload.
- The ASCII string filter functions allows searching for keywords or hex patterns at a defined offset
- E.g. filter out all http “GET” messages from a packet stream.



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.

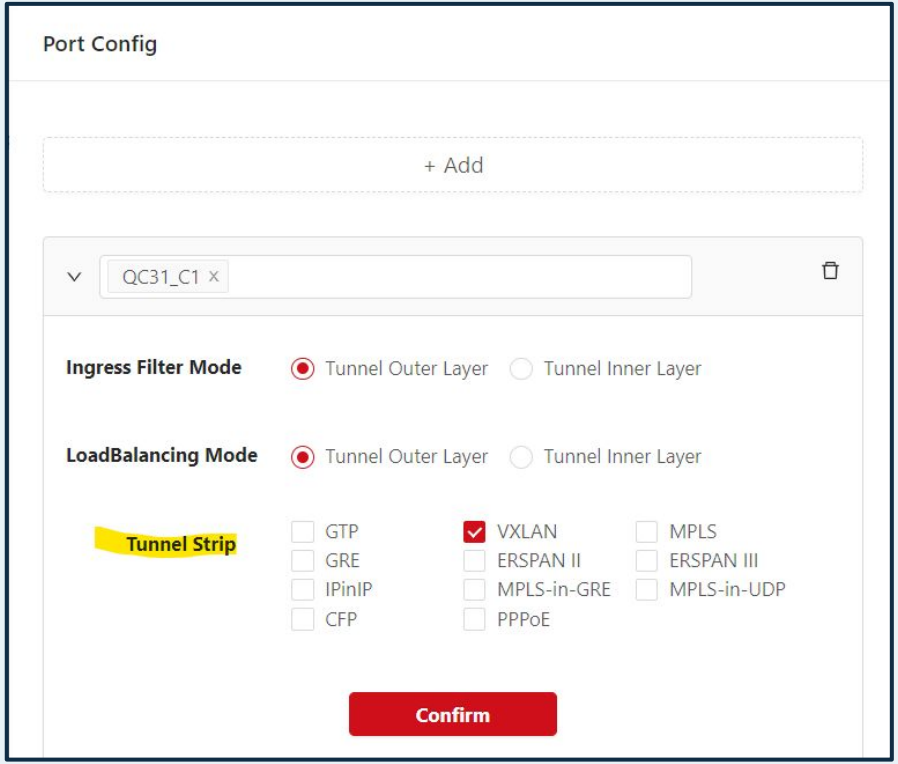


Challenges & Solutions

- 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 can not 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.

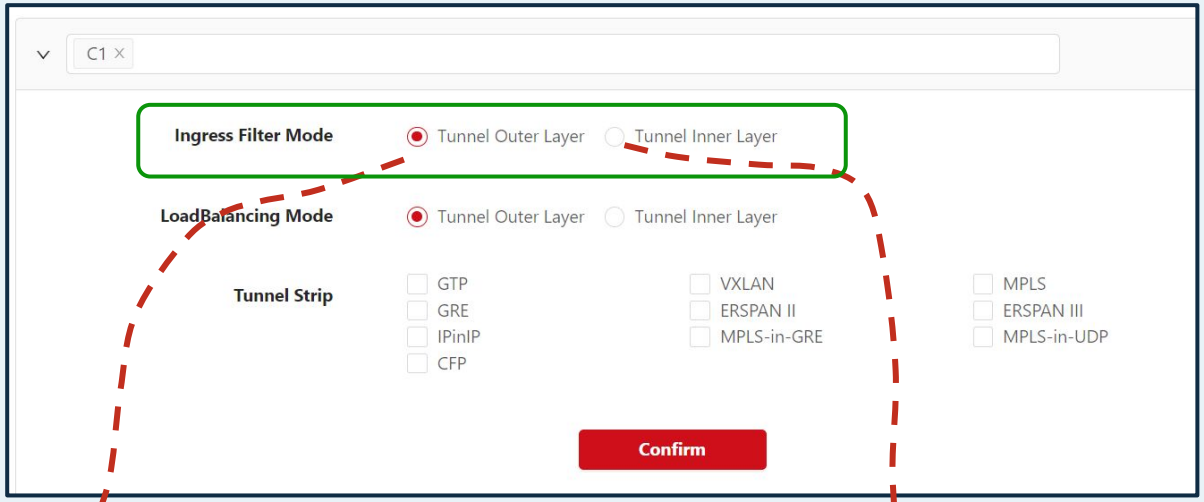
Tunnel Removal

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.



Outer or inner tunnel filtering

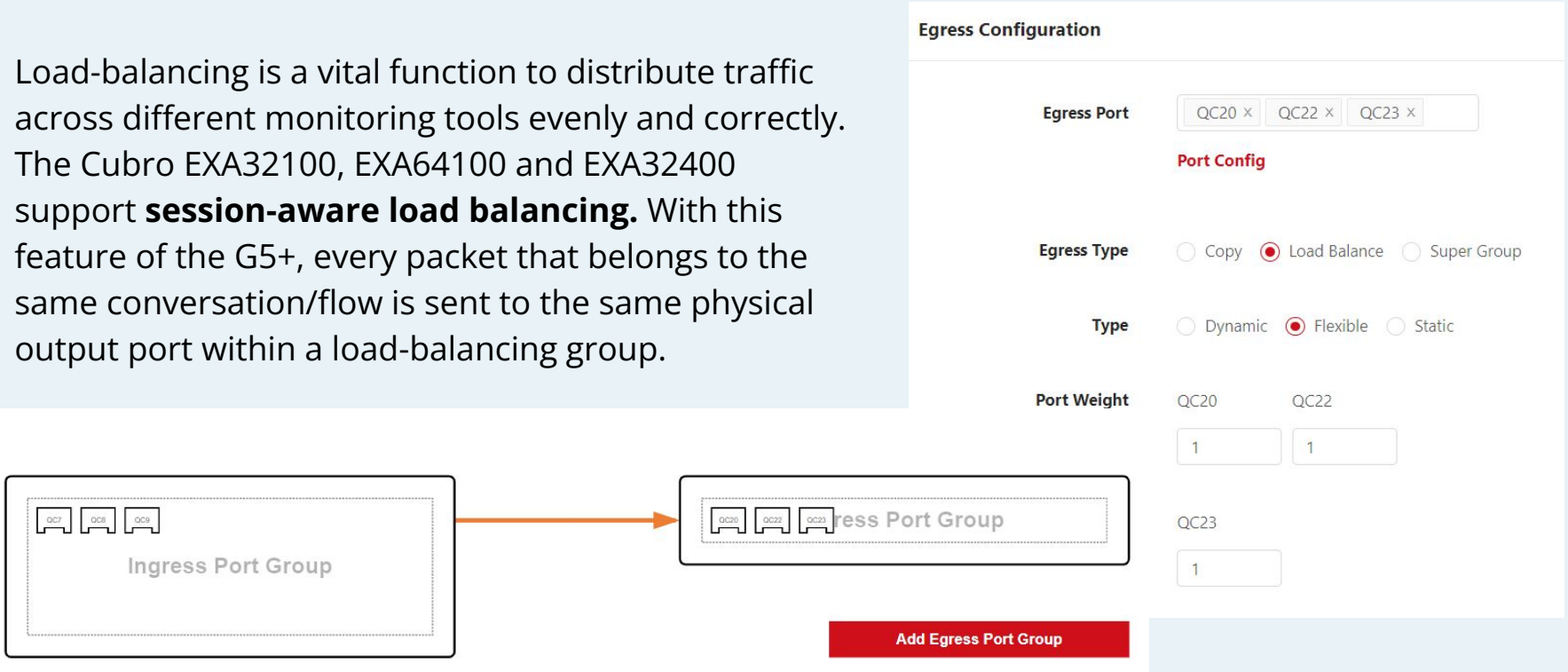
G5 plus series provides support for filtering on outer or inner tunnel packet parameters.



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

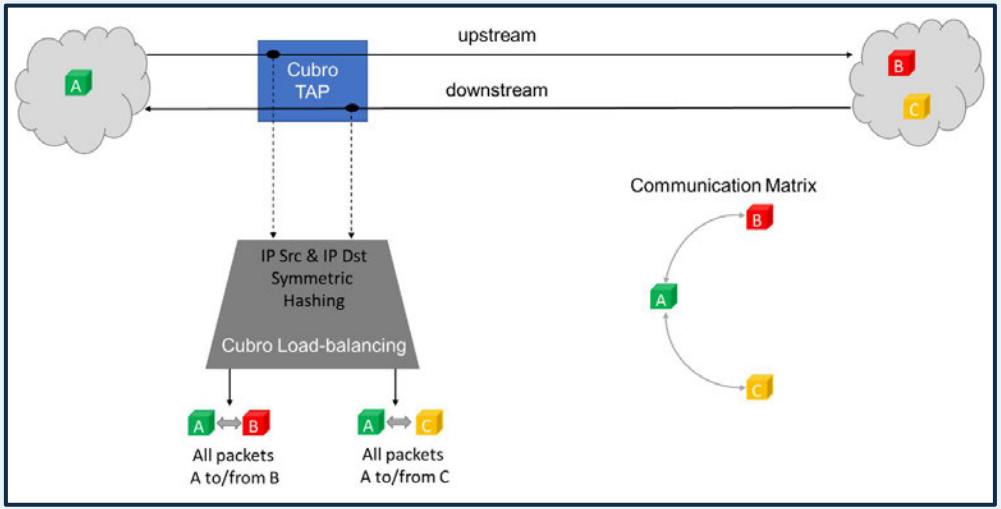
Load-balancing

Load-balancing is a vital function to distribute traffic across different monitoring tools evenly and correctly. The Cubro EXA32100, EXA64100 and EXA32400 support **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.



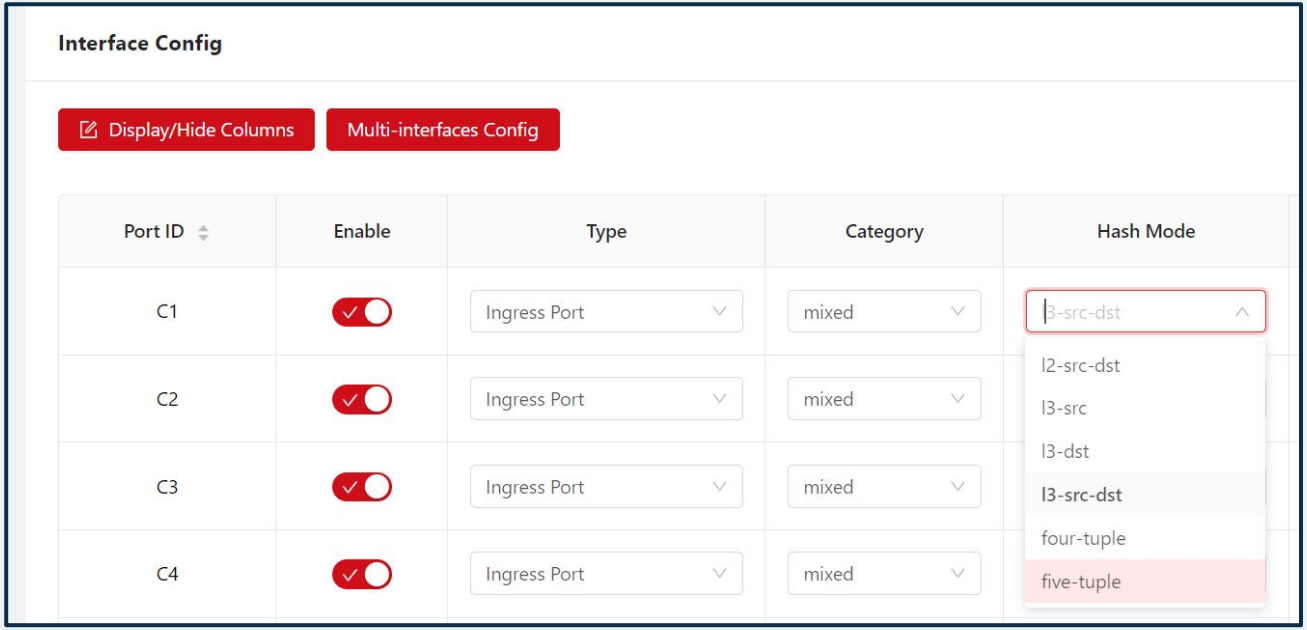
Session-aware load balancing & Hash-key calculation

Hash-keys are used to define the load-balancing behavior 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



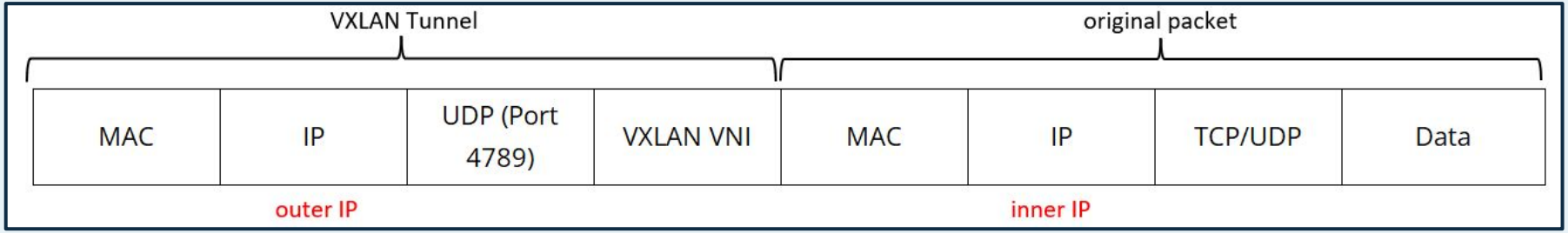
- Full flexibility to cope with all needs
- Individual setting per port

Hash-key calculation methods

Hash-key calculation method	Hash-key calculation based on	Remark
l2-scr-dst	full MAC Src & MAC Dst Addr	
l3-src	full IP Src Addr	
l3-dst	full IP Dst Addr	
l3-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.

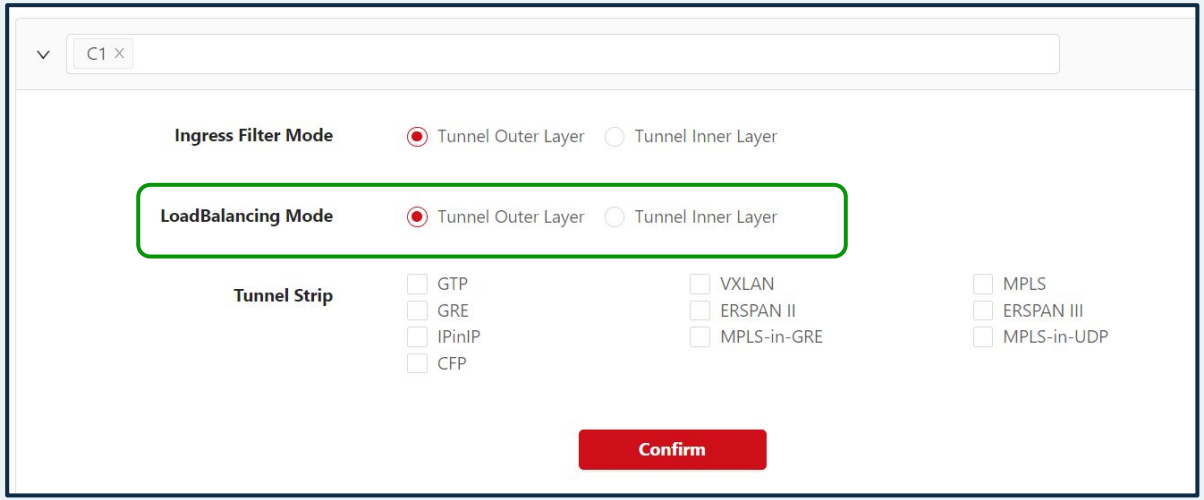


Outer IPs should not be used as hash-criteria - they are just the IP addresses of the overlay network. Load-balancing based on outer IP may encounter limitations in numerous scenarios. Load-balancing performs optimally when a wide range of packets (IP combinations) are accessible, whereas outer IP combinations are often limited.Load-balancing effectiveness improves when there are greater variations to consider. However, if only a small number of IP addresses are available, the load balancing could become highly asymmetric, leading to significant imbalances in traffic distribution.

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

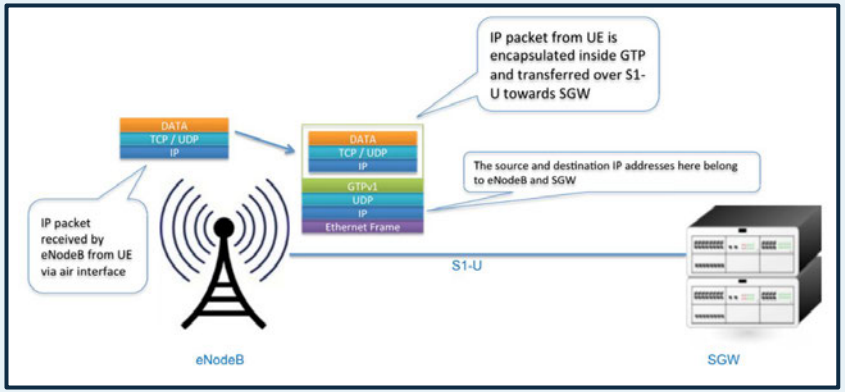
Cubro offers the choice

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



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

Load-balancing mobile GTP-U traffic

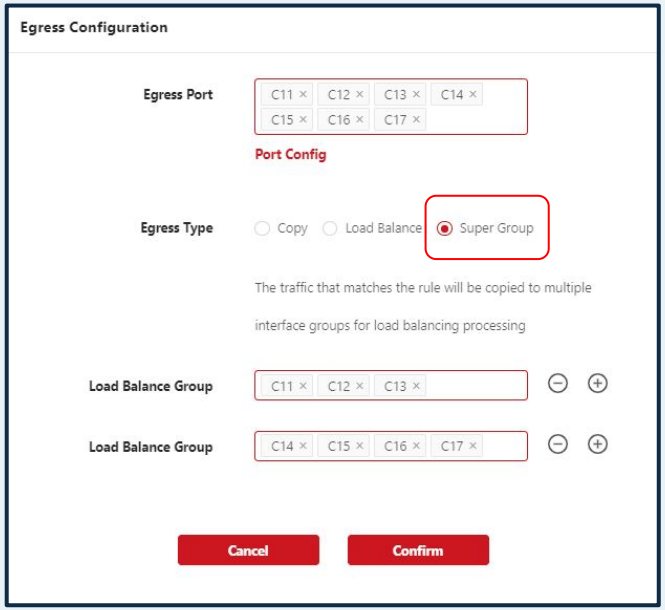


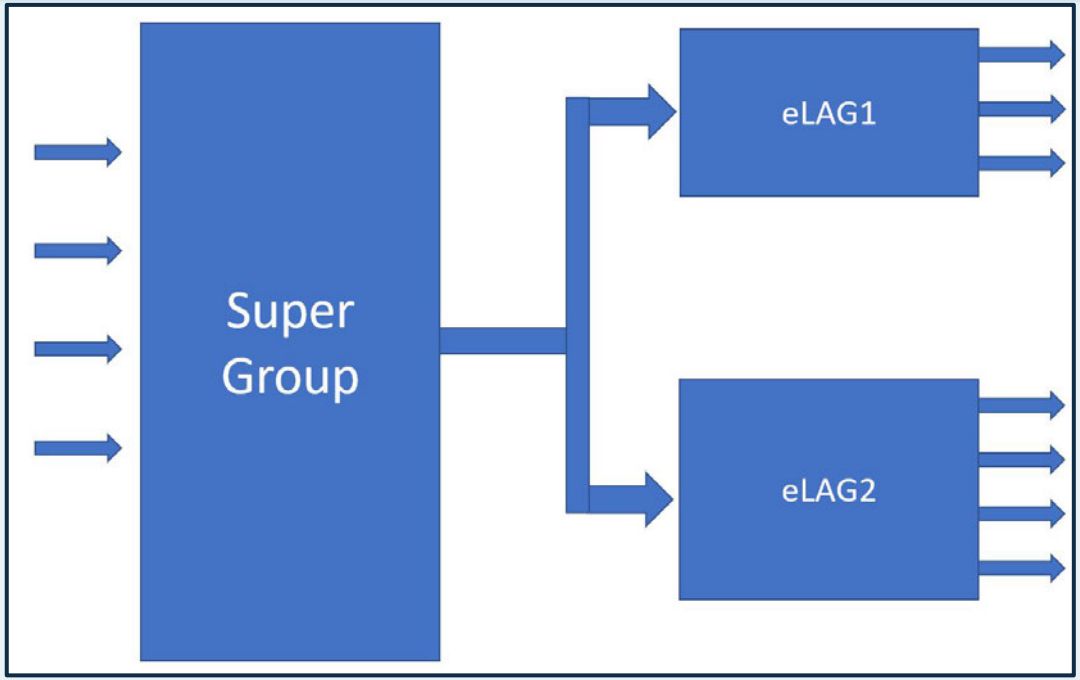
GTP is used in mobile networks to transport packets from the NodeB to the internet via an IP tunnel. Load-balancing could be based on outer IP Addresses which are the IP Addresses of the eNodeBs and SGWs. The problems using the outer tunnel for the hash-key calculation are:

- Small amount of IP Addresses when the outer IP is used → the **LB could be very asymmetric** resulting in uneven distribution of the traffic.
- The result of hash-key calculation will change once the user is moving. When the IP Address of the outer tunnel is changing (e.g. eNodeB change) the session will be moving to another output port and thus will be available at a different port of the monitoring appliance. Thus, **load-balancing will not be session-aware** from a user perspective causing more processing power required to do correlation/call analysis.

Solution: Use inner IP Address = user IP

Load-balancing to multiple groups

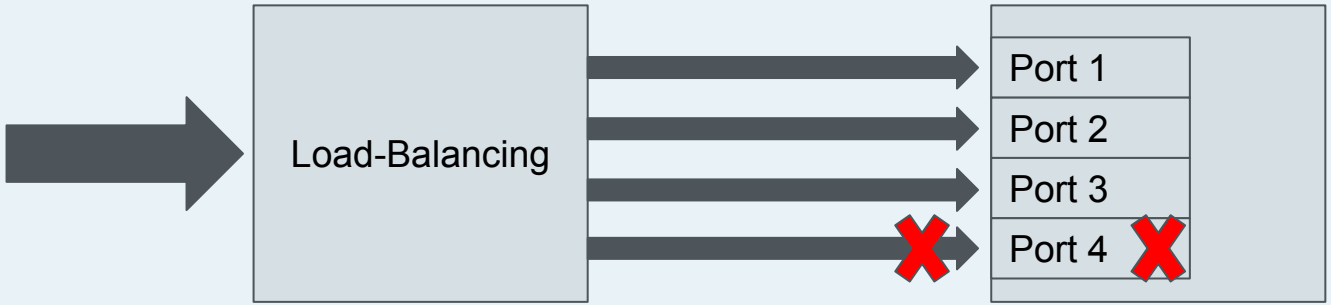




Distribute traffic to parallel analytic tools

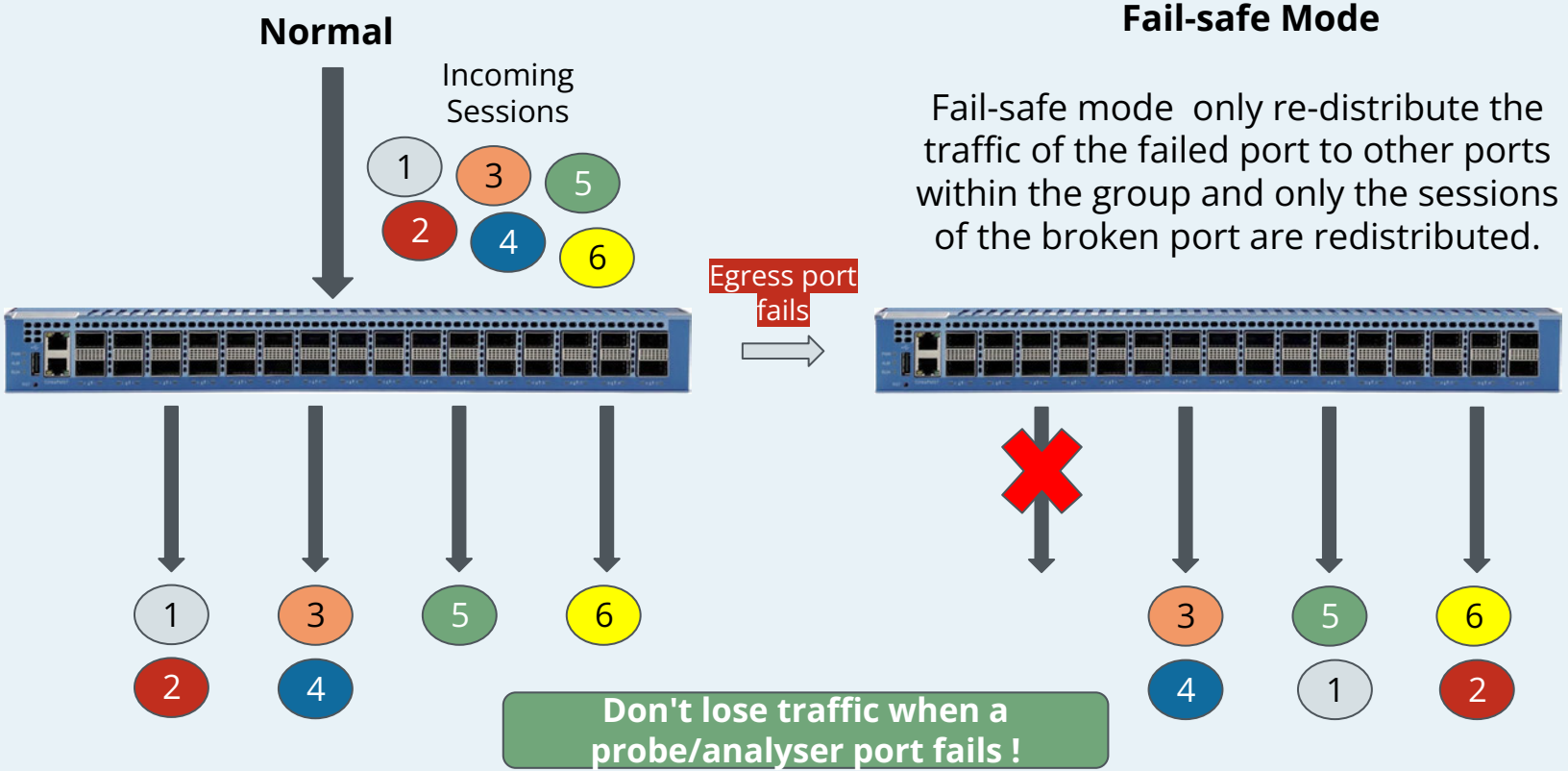
Traffic handling when an output ports fails

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



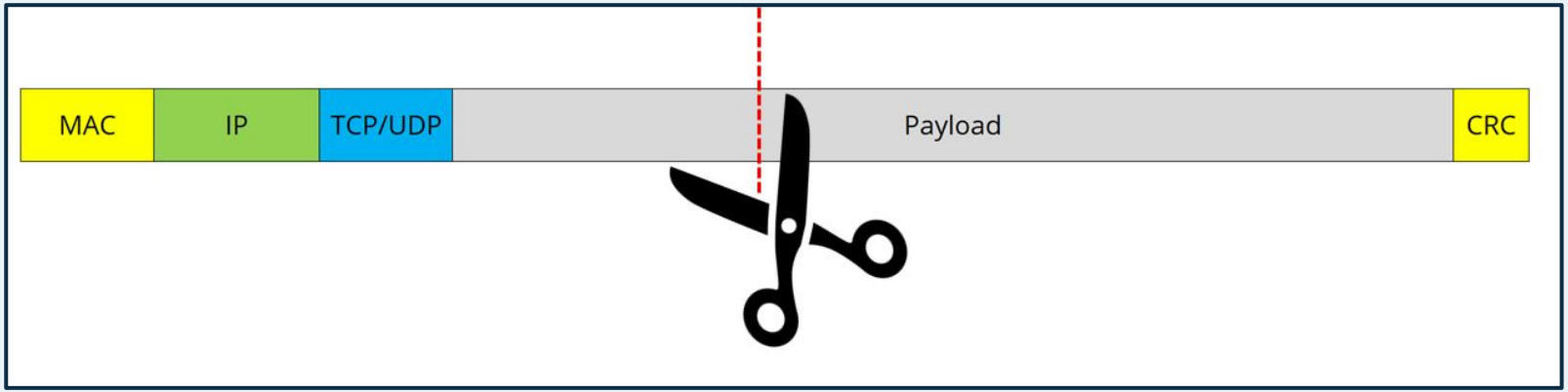
Don´t lose traffic when a probe/analyser port fails

Fail-safe Load Balancing



Slicing for any packet size to reduce output bandwidth

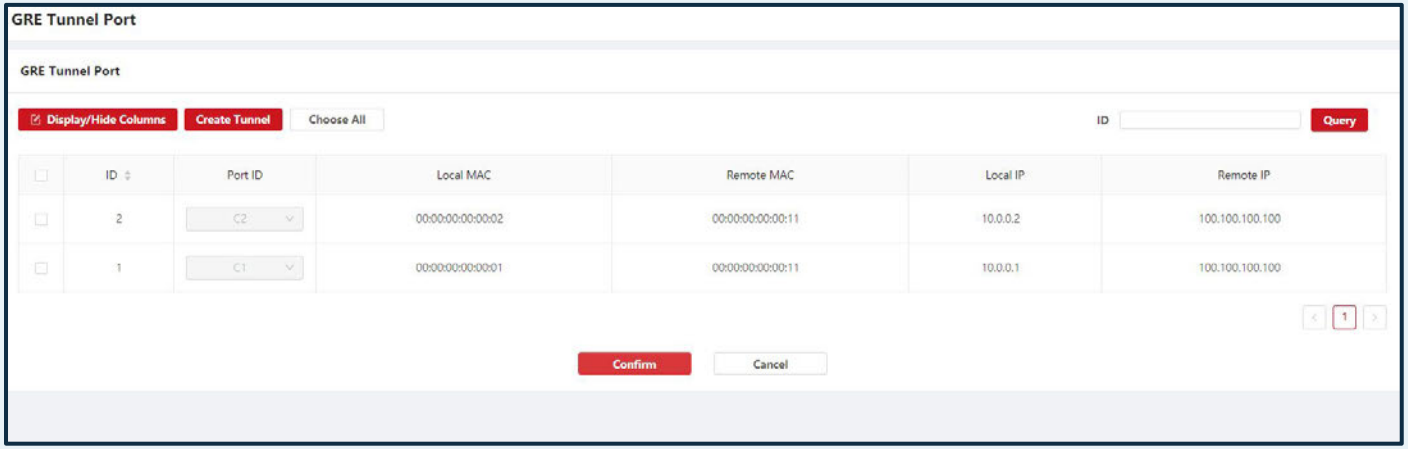
- Cubro G5+ Advanced NPBs allow to set the slicing size to **any value between 64B and 9192 Byte.**
- FCS is automatically corrected; all other fields inside the packet stay unchanged.



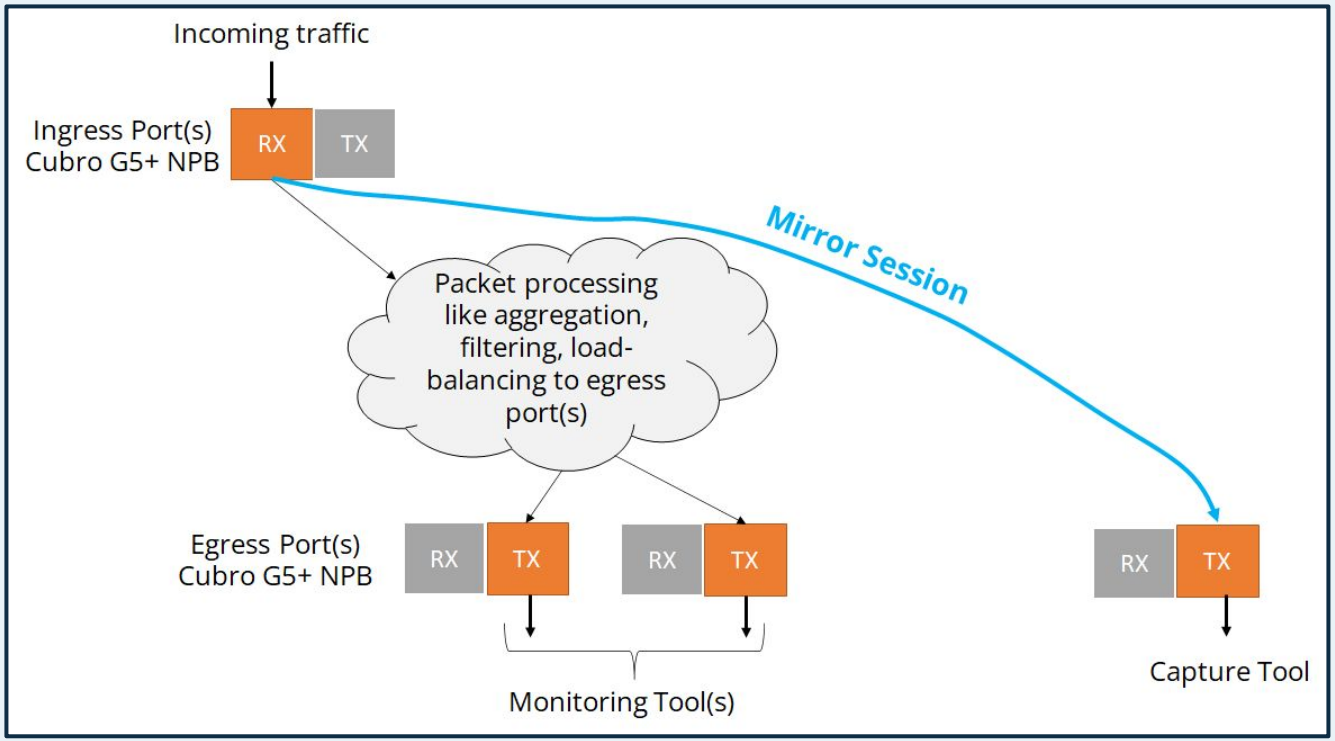
Reduces the output bandwidth sent to analytics and probing by removing parts of a packet that are not needed.

Active Tunnel Endpoint and encapsulated GRE output

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



Mirror function to easily add an output port for troubleshooting purposes

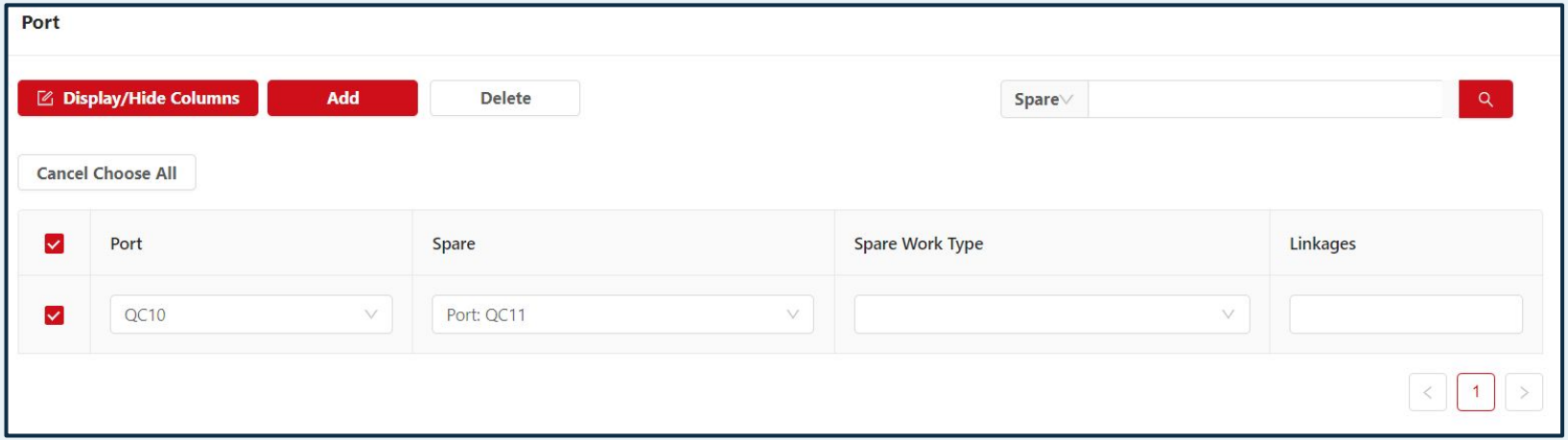


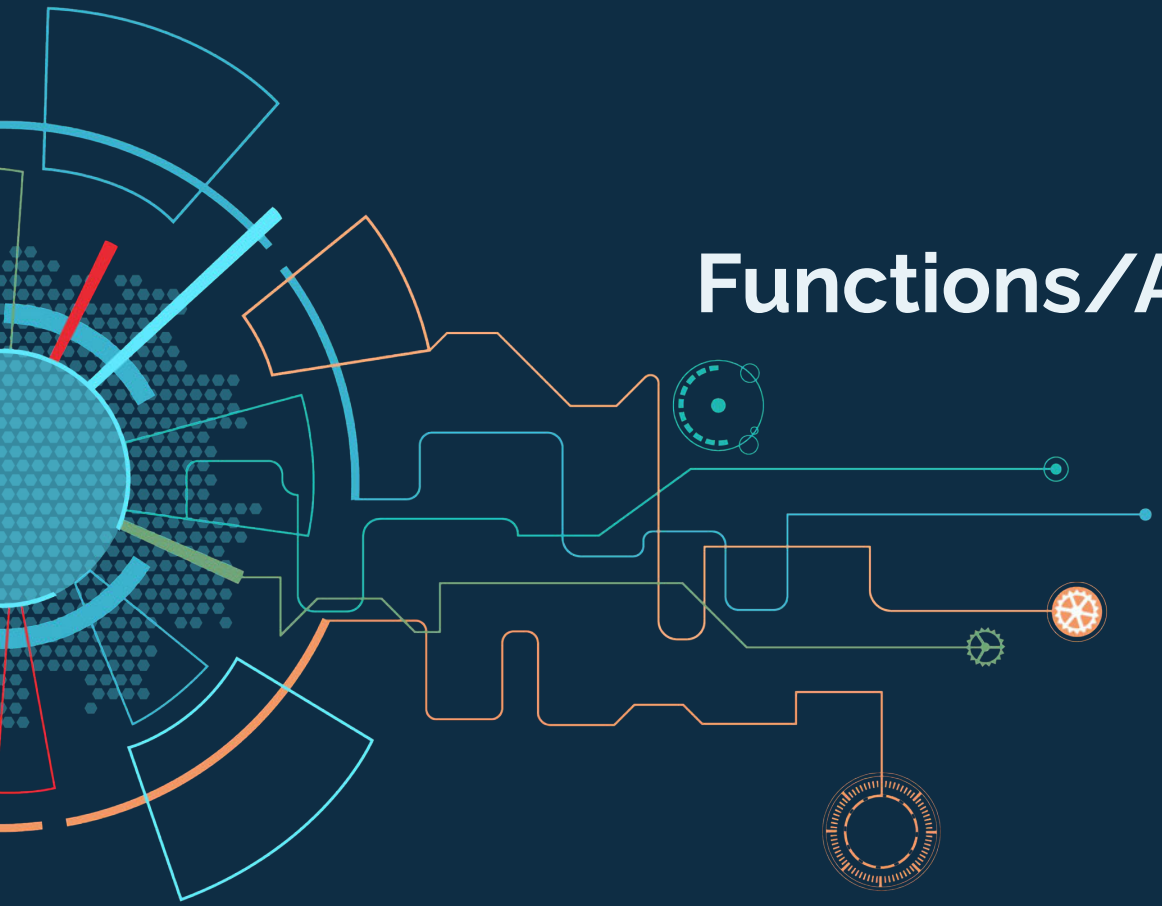
- Mirror RX port to output
- Mirror TX port to output
- Reduce mirrored output via filtering to reduce traffic load

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

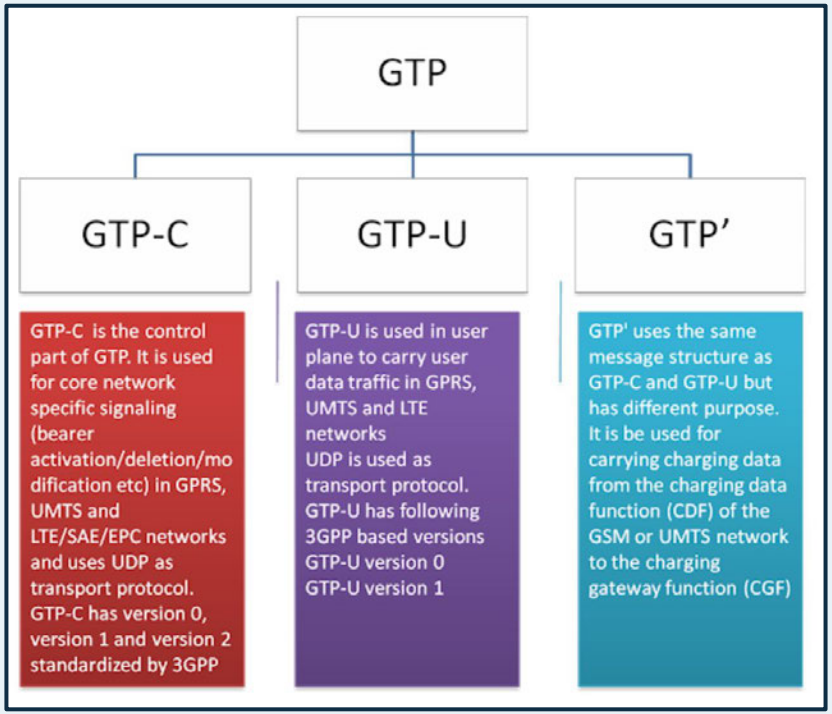


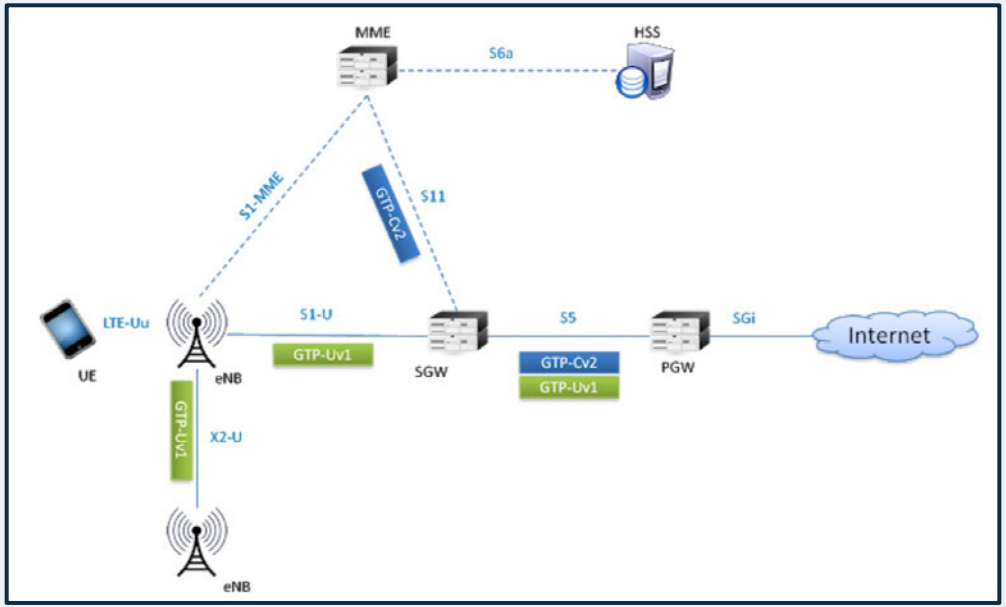


G5+ GTP Functions/Applications

GTP Overview

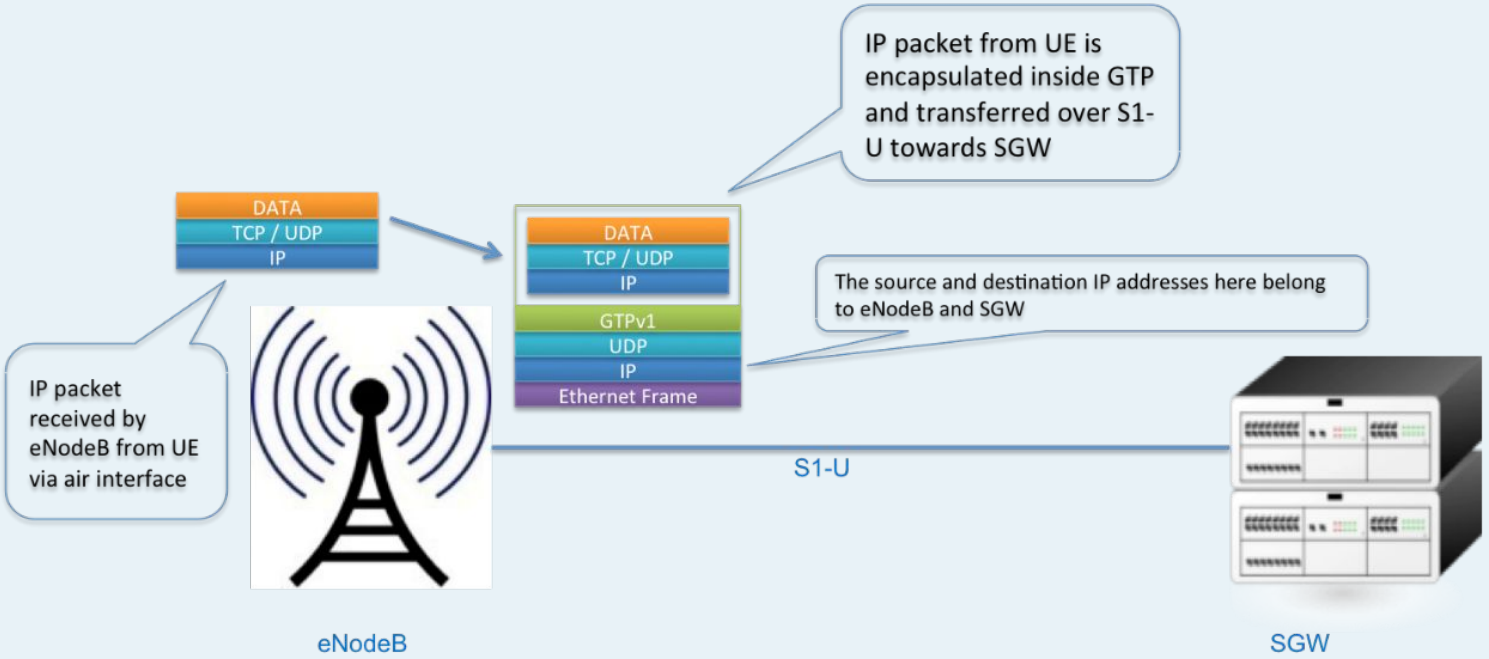
GTP = GPRS Tunneling Protocol





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
```

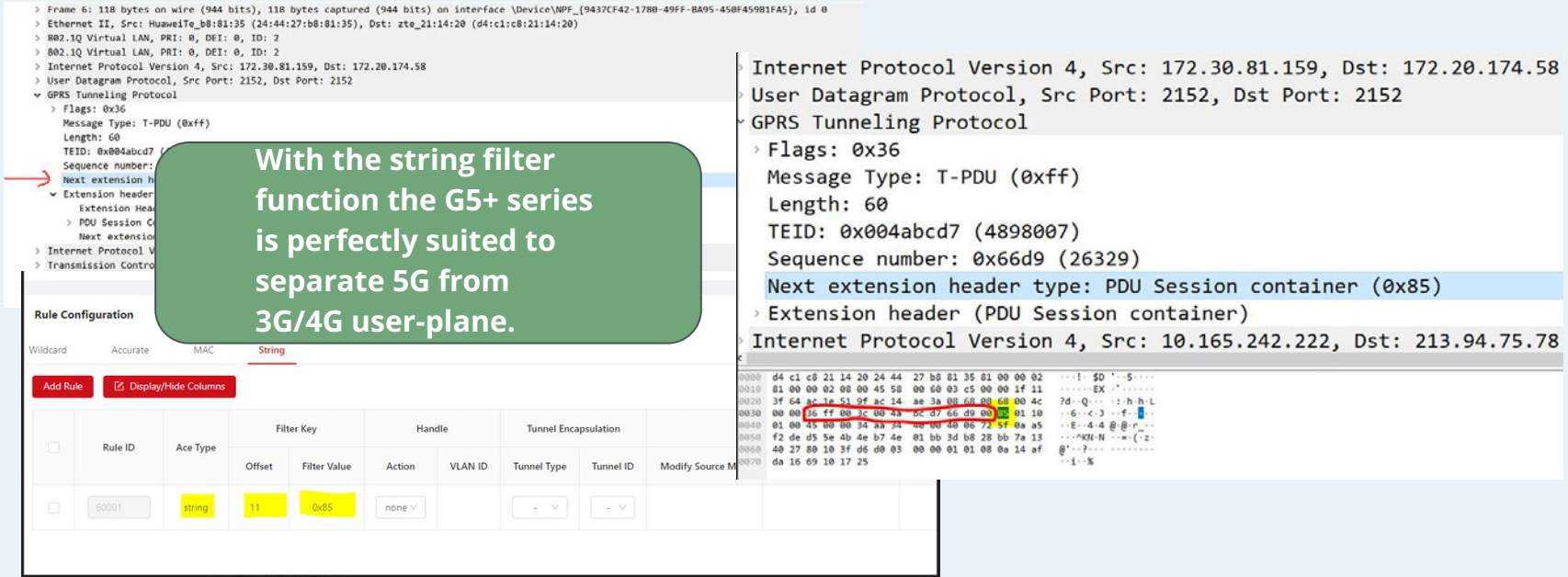
GTP-C = is the control plane of the protocol; Note that GTP-C does not have an inner IP

```
▷ 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 Userplane

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:



Cubro G5 Advanced GTP Applications

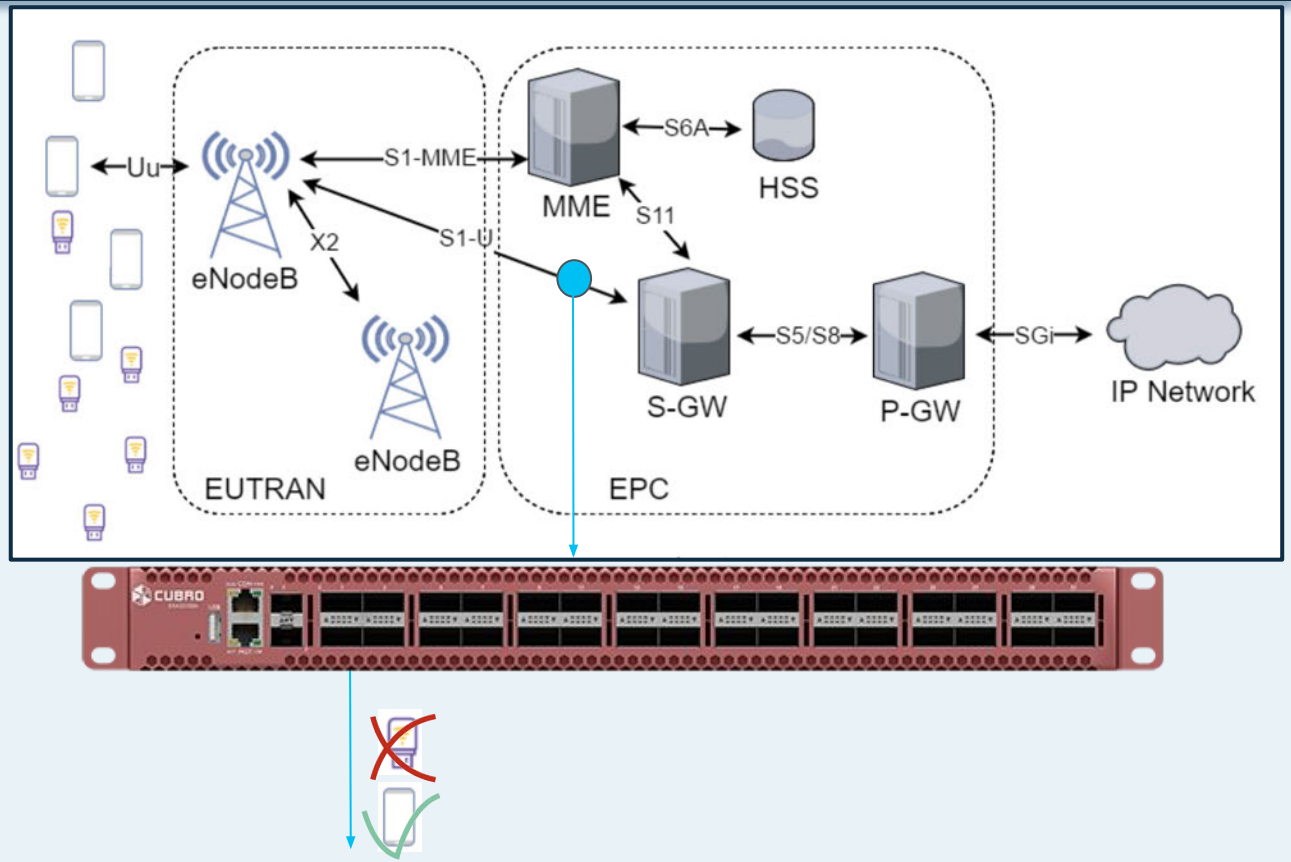
- GTP-U tunnel termination
  - Remove GTP-U tunnel header
- GTP-U Inner IP filtering including IP range filtering
  - Drop traffic by simple inner IP filtering to avoid overload on monitoring probes
- GTP-U Inner Layer 4 (application) filtering
  - Filter directly on S1-U interface and feed the traffic 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.

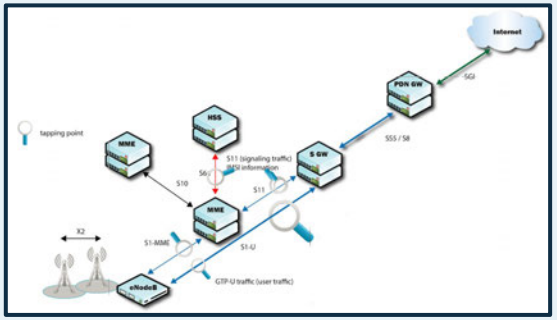
Filter on GTP inner IP Address range to drop traffic from/to LTE modes.

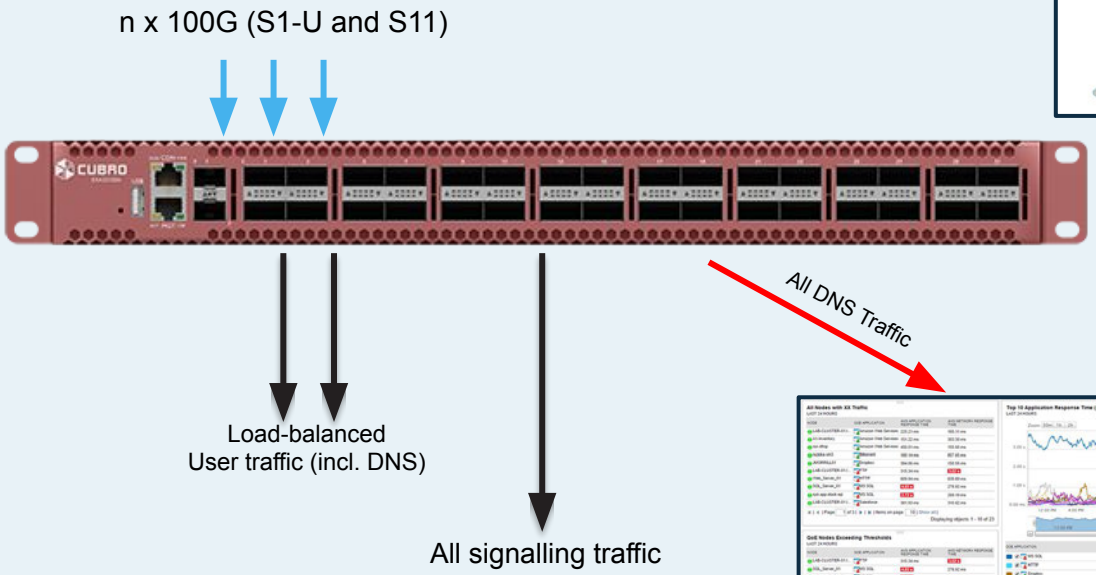


Application filtering inside GTP Tunnel

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

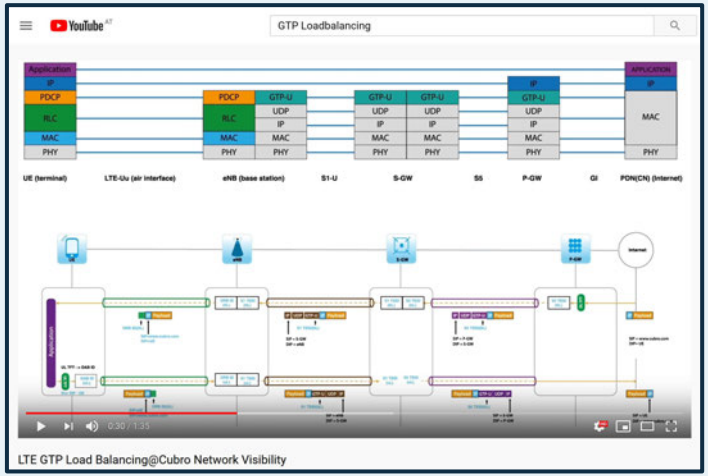
This is a simple and scalable solution to offload irrelevant traffic from the probes and thus saves costs.





GTP-U load-balancing

- 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.



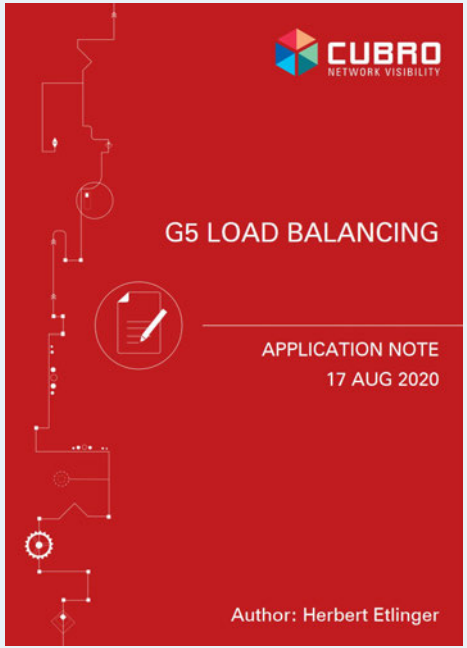
Load-balancing - some more details

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:

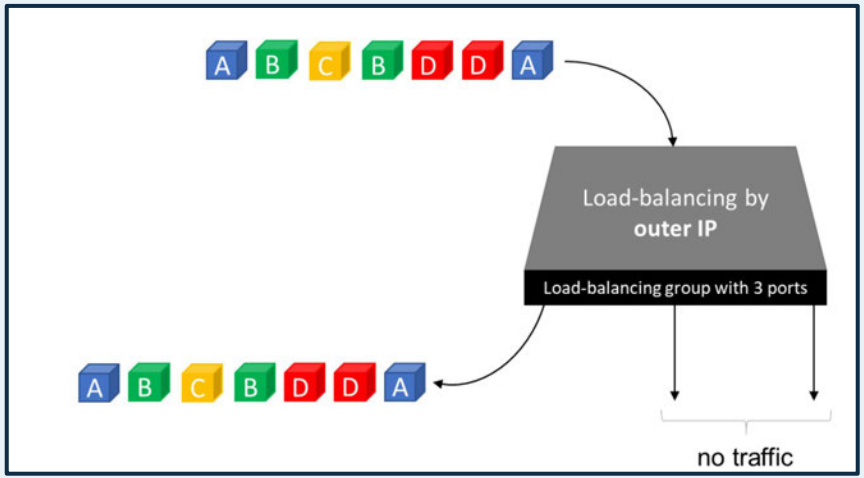
Hash-key calculation method	Hash-key calculation based on	Remark
l3-src	full IP Src Addr	
l3-dst	full IP Dst Addr	
l3-src-dst	full IP Src & IP Dst Addr	Upstream & downstream direction give DIFFERENT hash results -> upstream & downstream is split apart -> not session aware E.g. 10.0.0.1 talks to 10.0.0.2: Hash result = x 10.0.0.2 talks back to 10.0.0.1: Hash result = y
l4-src-dst	full Layer 4 Src & Dst Port	
four-tuple	full IP Src & Dst Addr & Layer 4 Src & Dst Port	
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	
l3-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	
l3-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
l4-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	
l3-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.

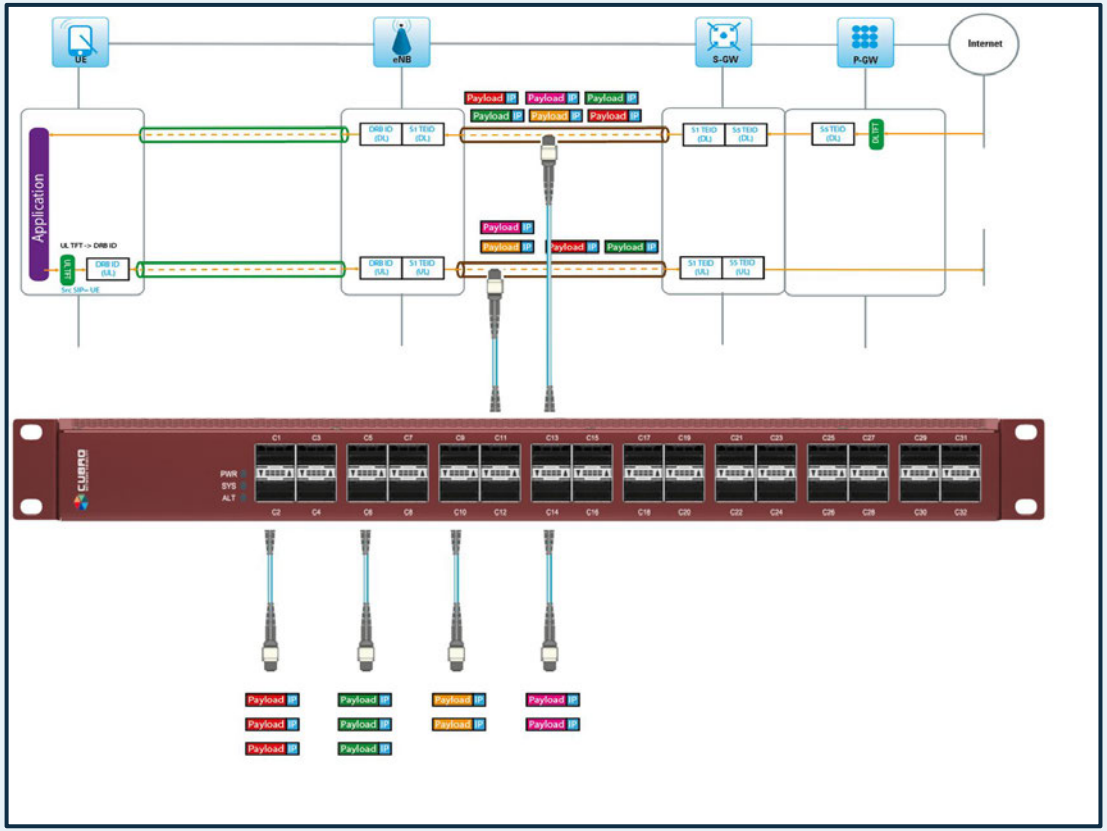


Problems caused by using outer IP for load-balancing

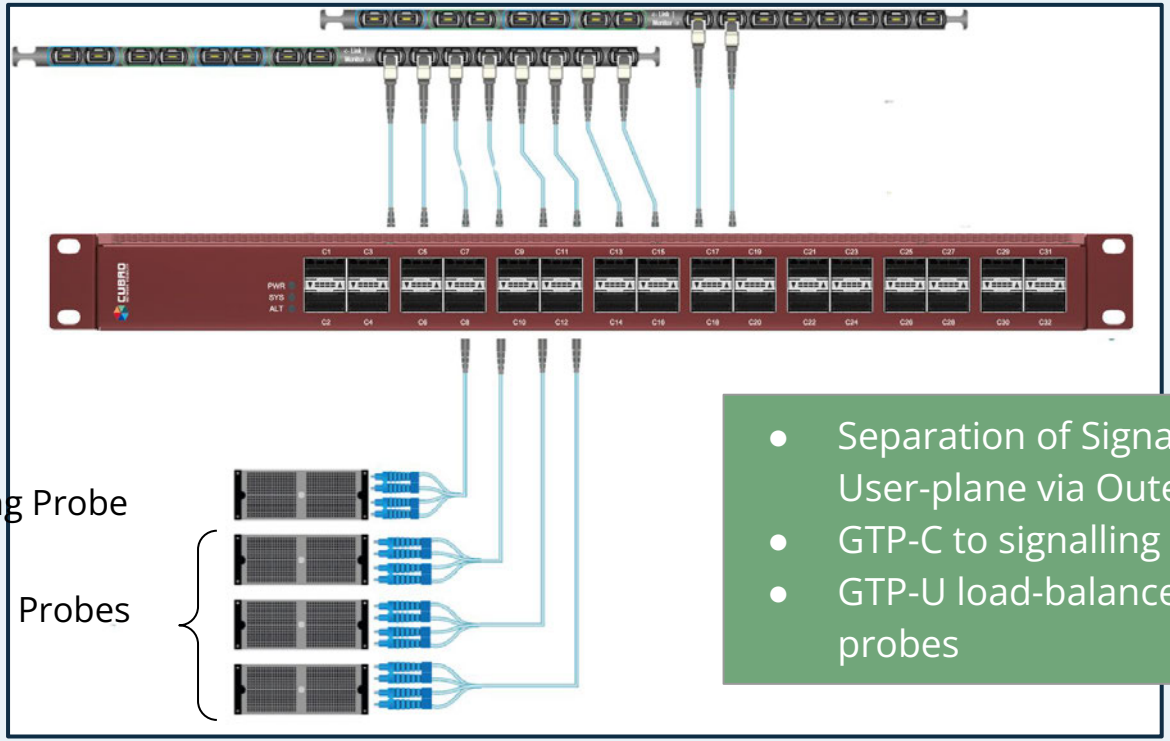
- The monitoring session for a user will be interrupted when the customer is moving to another location.
- Due to the small amount of outer IPs, the load-balancing could be asymmetric. This means the output ports can be overloaded which causes packet drop and thus bad monitoring quality.



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

Summary

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

We have operations in all time zones.
Reach us at: support@cubro.com