The bridge to possible

The New, Encrypted Protocol Stack & How to deal with it

Adding Real Value to Networks

Andreas Enotiadis (MIG Sales CTO)
Bart Van de Velde (Sr. Director, Engineering, Networking CTO Office)

In memory of and based on the brilliant work of Mark Gallagher

(14/09/1966-17/09/2021)

Agenda

- The New Internet
- Toolbox
- Use cases

Sessio

The New Internet

The Internet Reality – circa 2020 – Major US Carrier

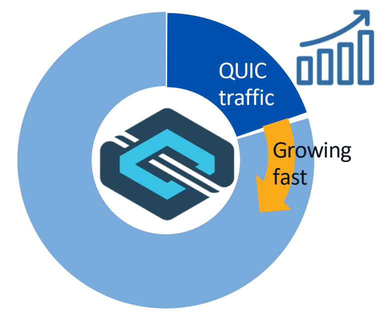
>90% of
Volume: encrypted

>70% of
Volume: to Cloud

10 Cloud sites
“Elephant destinations”
not “Elephant flows”

~50% of Flows:
DNS

>20% of Traffic:
QUIC



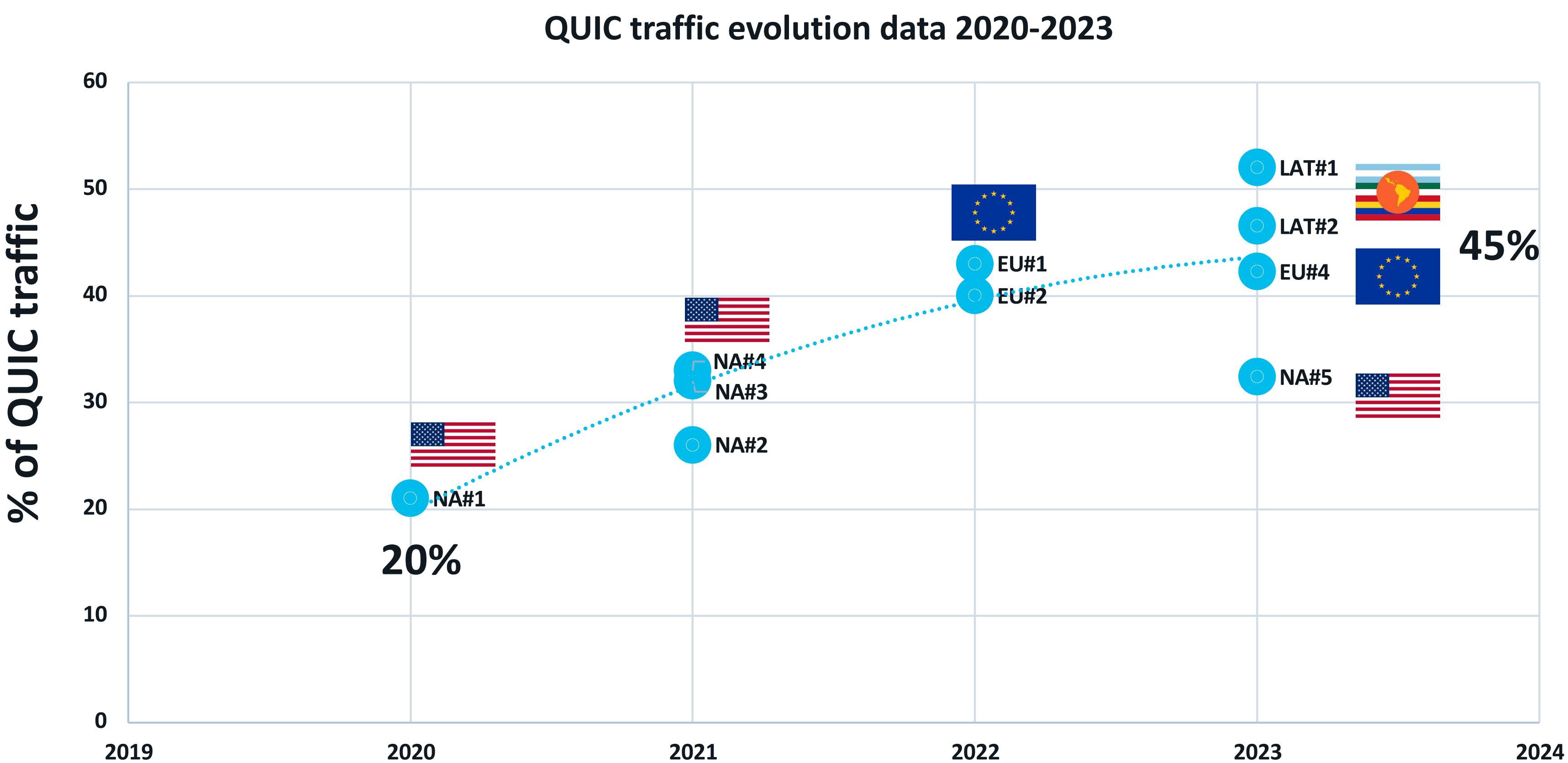
Many small flows
Micro-sessions

- Destination: all-encrypted world
- Cloud: concentrating the Internet

- Content: DNS is the load-balancer
- QUIC: Future Protocol of choice

QUIC is growing across the world

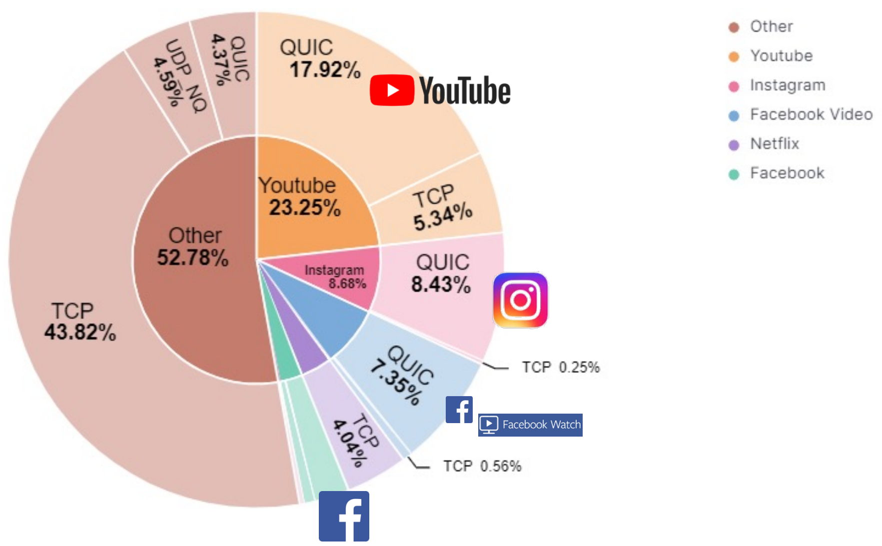
various snapshots



Network Traffic by Volume and Flows

Overall Volume by Apps

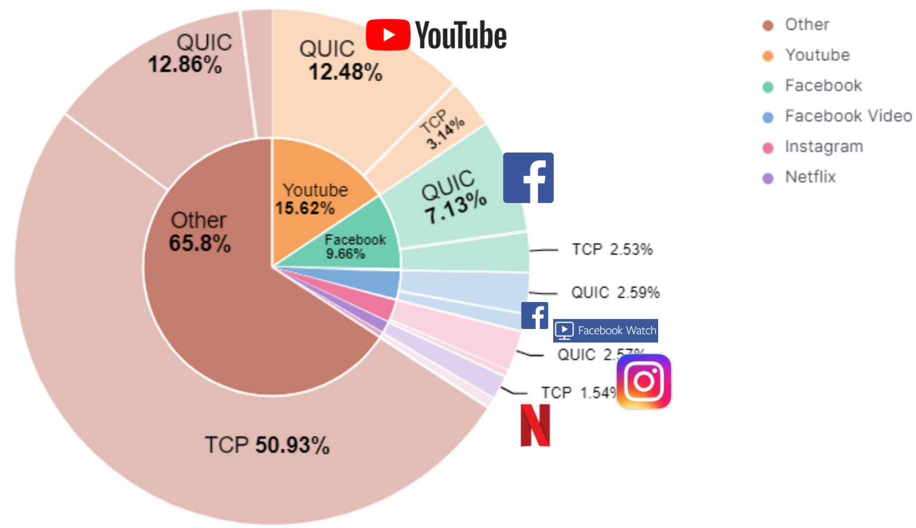
Big 5 is 48% of traffic
QUIC is 40% of traffic
"other traffic" still largely TCP, QUIC now visible (4.3%).



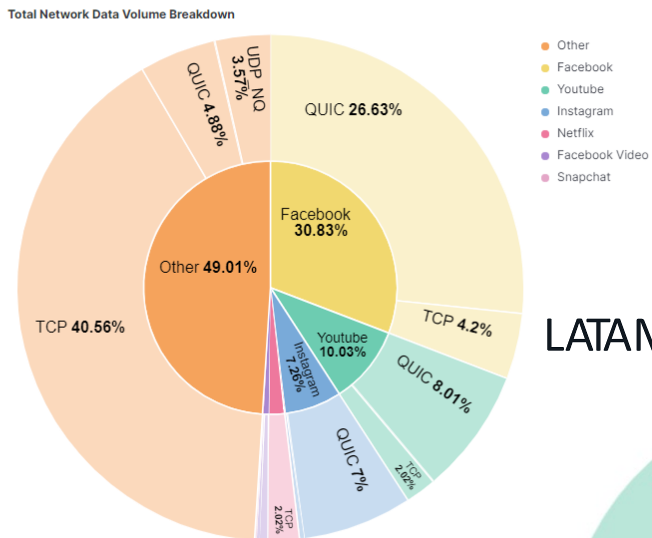
*source EU Operator 2022

Total Flows by Apps

Lots of TCP sessions (likely IOT related, transactional related)
Big 5 QUIC sessions are very targetted and high efficiency (video related behaviour)

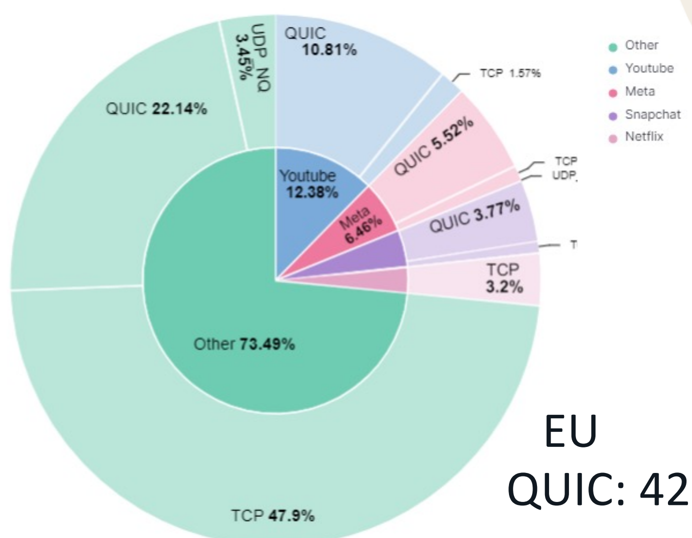


The pattern persists worldwide into 2023



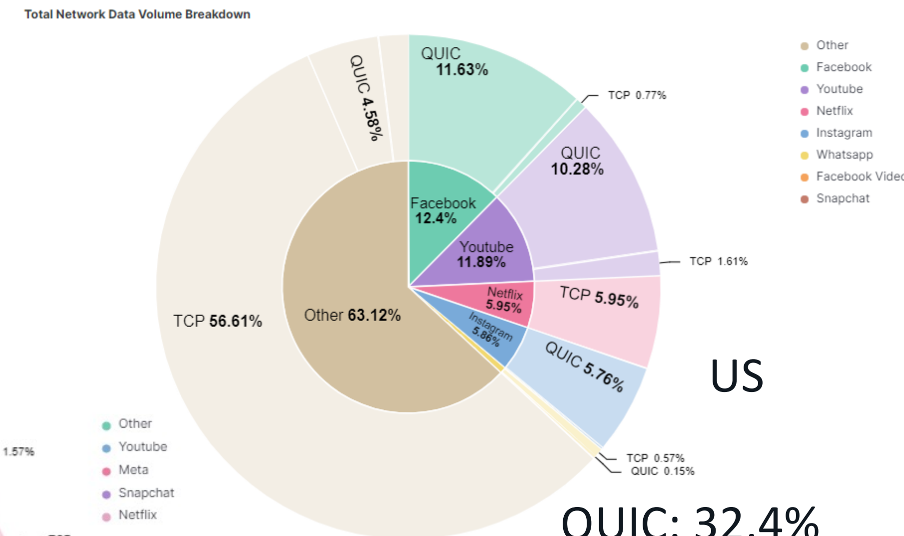
LATAM

QUIC: 46.52%



EU

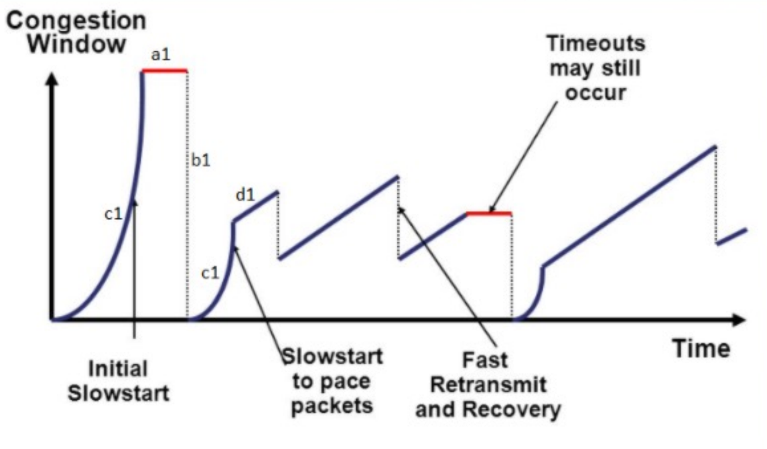
QUIC: 42.24%



US

QUIC: 32.4%

The old network design assumptions are challenged



TCP goal is network fairness

Today IP Networks are architected with TCP behaviour as implicit assumption

So when IP packets or PDUs are dropped TCP will take care of it at a higher layer

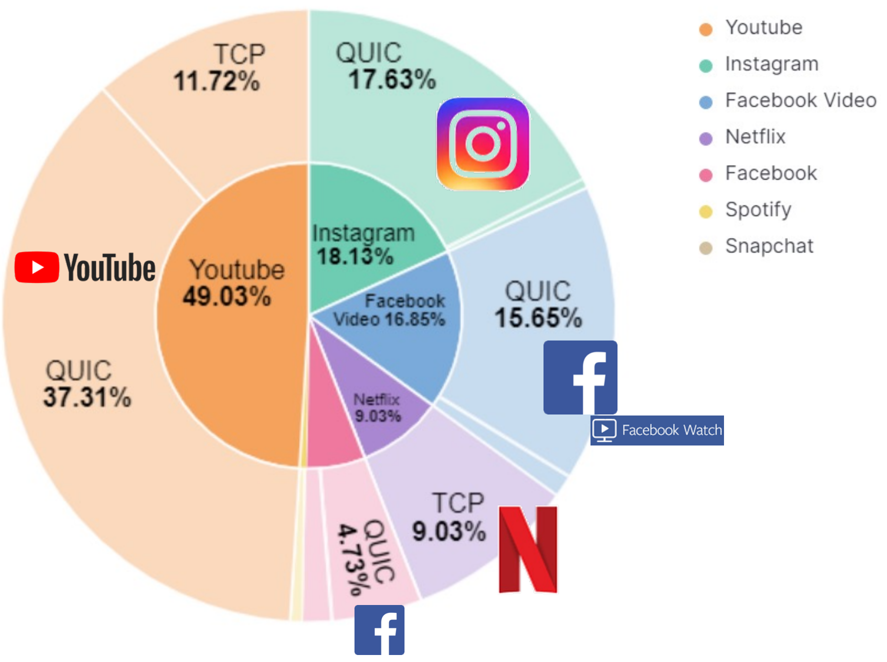
Scenario	Flow	Avg. throughput (std. dev.)
QUIC vs. TCP	QUIC	2.71 (0.46)
	TCP	1.62 (1.27)
QUIC vs. TCPx2	QUIC	2.8 (1.16)
	TCP 1	0.7 (0.21)
	TCP 2	0.96 (0.3)
QUIC vs. TCPx4	QUIC	2.75 (1.2)
	TCP 1	0.45 (0.14)
	TCP 2	0.36 (0.09)
	TCP 3	0.41 (0.11)
	TCP 4	0.45 (0.13)

*** Source : APNIC**

QUIC goal is “MY App” performance

What are the IP Network Design assumptions wrt QUIC ?

Top 5 Apps – QUIC is dominant
80/20 rule now



April 10 2022

An application driven global transition

HTTP/3 Stack = UDP+QUIC+TLS

Old App Stack

HTTP/1.1/2

TLS

TCP

IP

New App Stack

QUIC – RFC 9000
HTTP/3 – RFC9114

HTTP/3

QUIC + TLS1.3

UDP

IP

- ***Improved Security***
- ***Multi-session***
- ***Improved QoE***
- ***APP friendly design***

DoH

DoT – RFC7858
DoH – RFC8484

Application Controlled DNS
DNS Traffic not observable

Google & CloudFlare serve 50% of global DNS requests
Both support DoH
All major OSs & Browsers support DoH
(Firefox Defaults for US to CloudFlare)

eSNI / ECH

RFC8744

Target Domain is opaque / unobservable

TikTok Uber

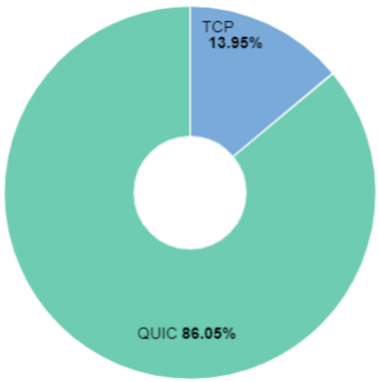
Large Scale Adoption

DPI Ineffective

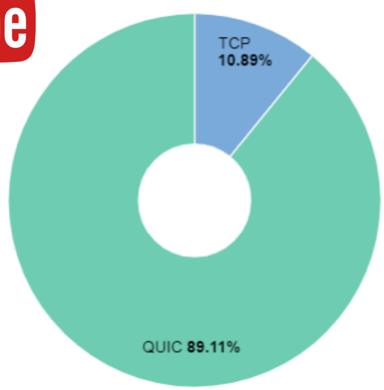
including alternative hints e.g. DNS or SNI analysis

Packet Inspection needs different approach

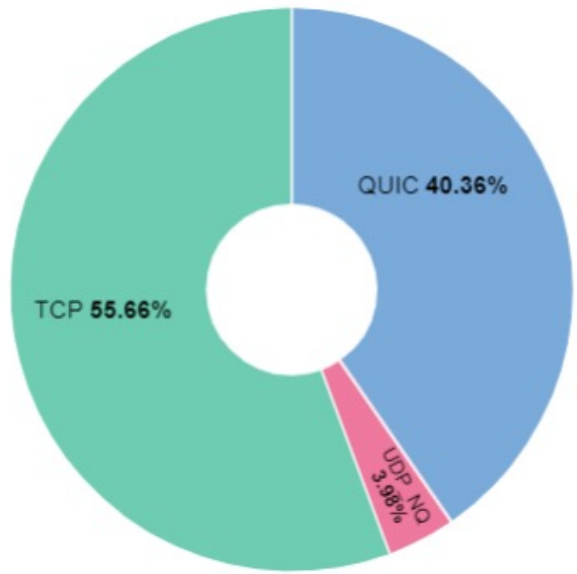
QUIC : 86%



QUIC : 90%



Overall Volume



TCP : 55%

QUIC : 40%

*source: Live Traffic USA Operator ; dd. May 2023

QUIC/H3/DoH stack is in business

Content Delivery | **Security** | **Privacy** | **Loadbalancing** | **App Infrastructure** | **App Experience**

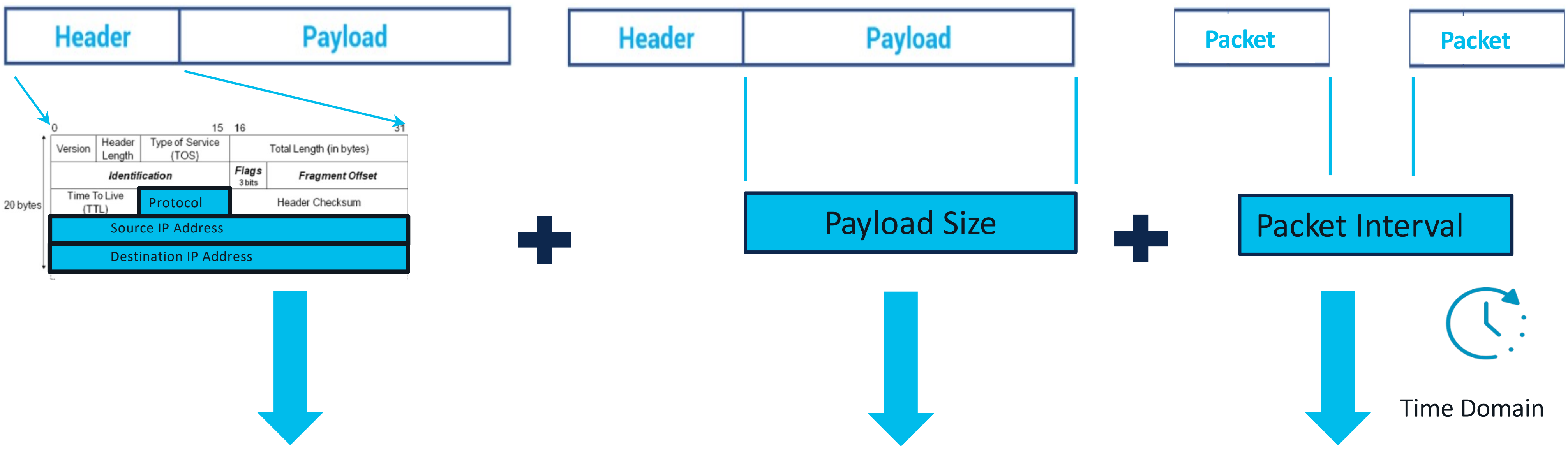
Dealing with the new reality: Toolbox & Use Cases

Customers are looking for solutions

Example Use Cases Asked

Manage video downloads vs video streaming, downloads being the priority

There is some information that will not go away

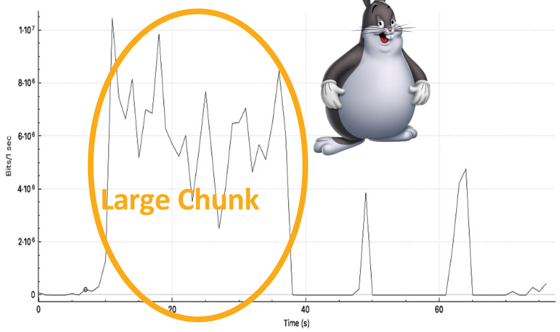


CDN Information

Traffic Volume in Time Information

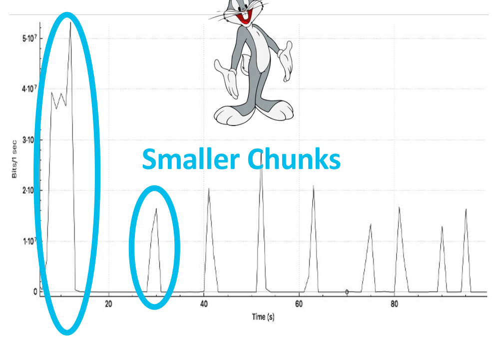
App (e.g. Video) Behavior varies by protocol and use case

TCP Video Stream Detection



TCP based ABR video players prefer **larger, sustained downloads** due to high cost of establishing the TCP session and reducing time spent in TCP slow start.
Often use HTTP/2 connection. (DASH/HLS) to fix HOL.

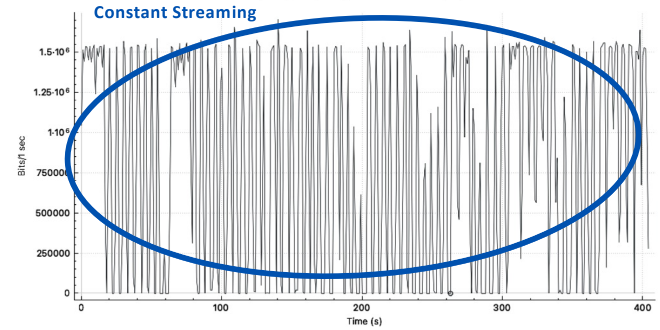
QUIC Video Stream Detection



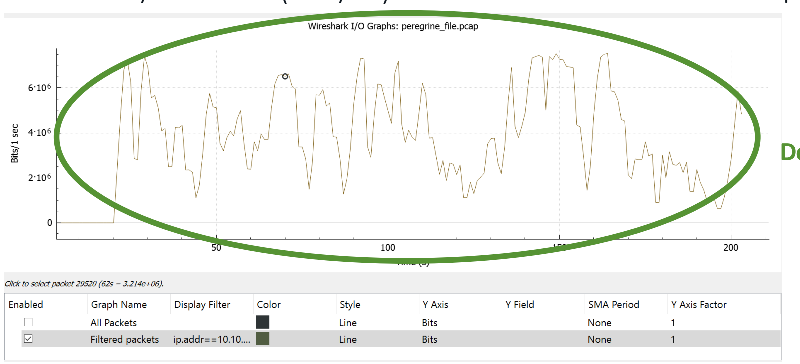
QUIC based ABR video players prefer requesting **video in smaller chunks**.

Multiple QUIC Streams in many cases to (different) servers

UDP Video Live Stream Detection



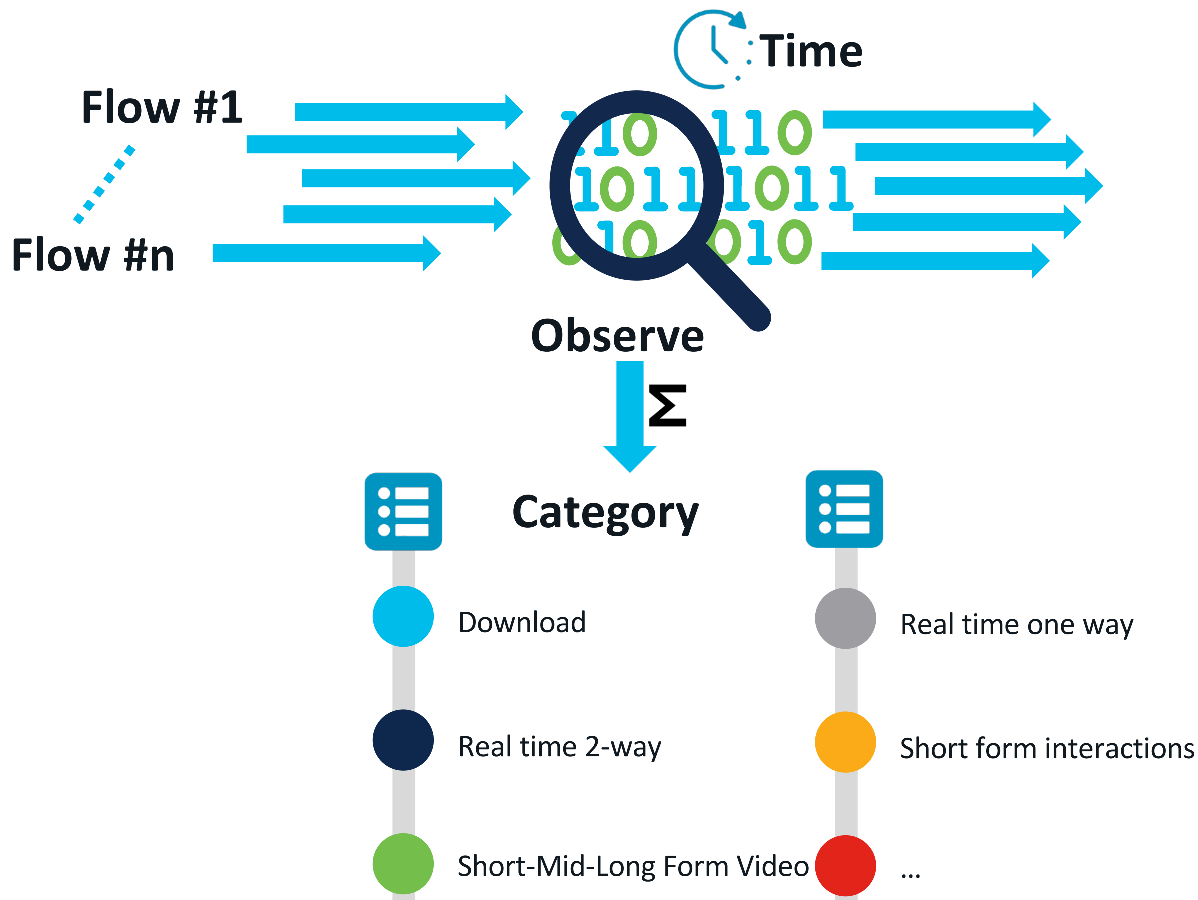
UDP based video players are extremely reliant on consistent network performance. Small buffer, sustained T'put
Applications: YouTube Live, WebEx, Microsoft Teams, Zoom



Download Stream Detection

Time Domain Flow recognition

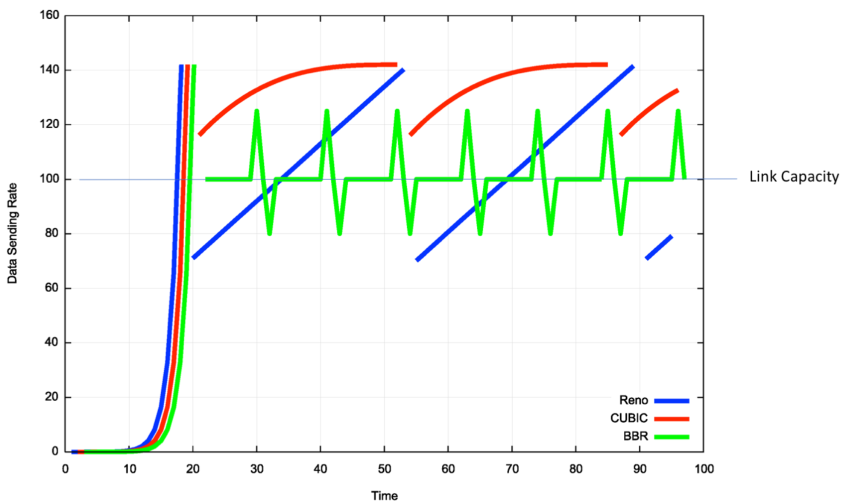
- Observe all flows
- Profile per flow (Time domain matched)
- The resulting profile will allow to distinguish the nature of the flow
  - Content Download
  - (x-Form) Streaming content
  - Real time 2 way communication
  - Video/non-video
  - Short lived flows



Inferring congestion

- Different congestion algo’s have different behaviour
- Time-domain observation + anomaly detection -> congestion inference

Reno vs CUBIC vs BBR behaviour*

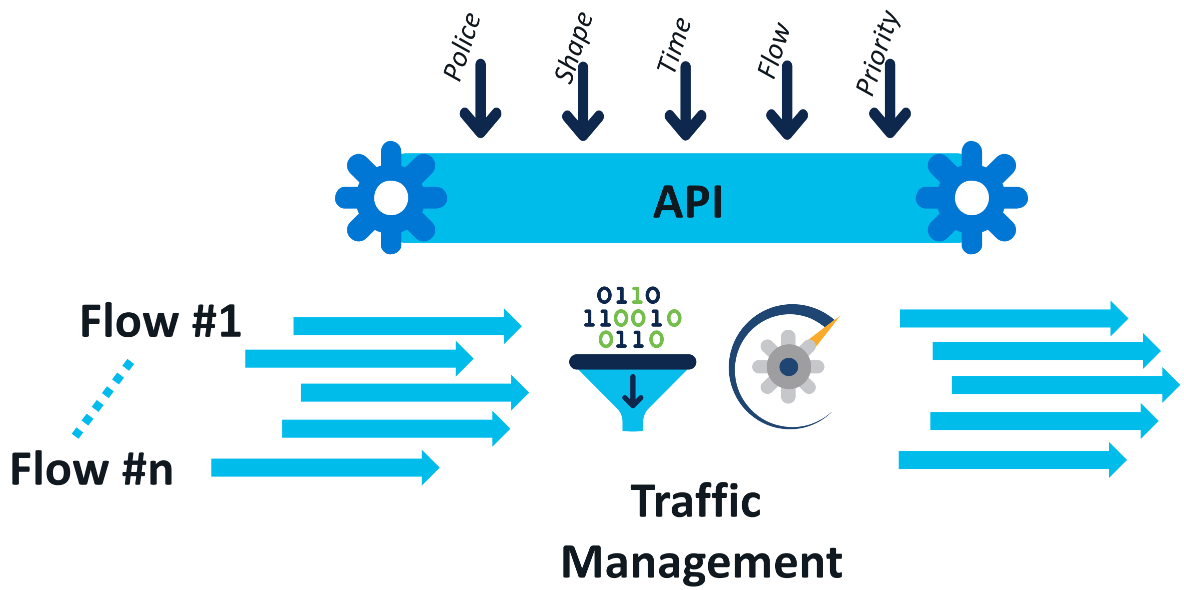


- Assessment of various flows in parallel
- Understand Protocol behaviour: congested or not
- This serves as input for Policy Application

* https://blog.apnic.net/2017/05/09/bbr-new-kid-tcp-block/

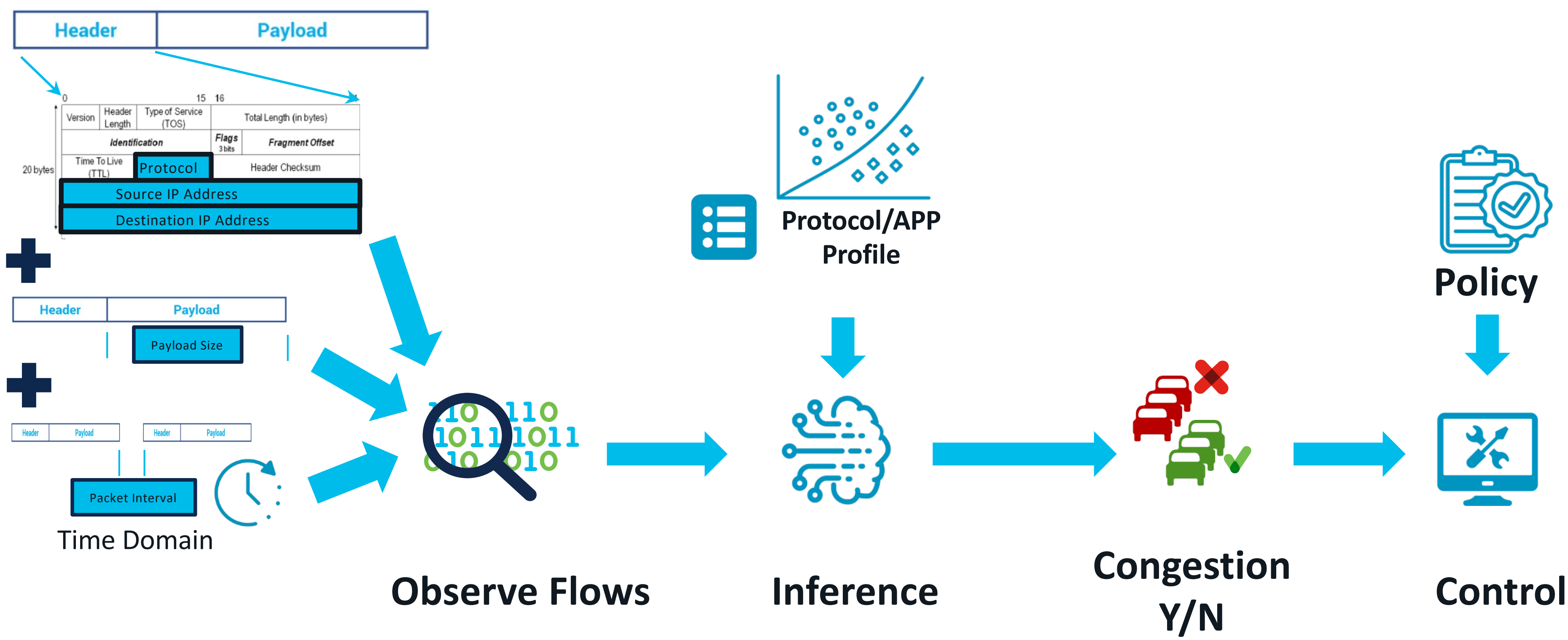
Programmable Traffic Management

- Traffic can be controlled in various ways.
  - Buffer
  - Discard
  - Flow control
  - …
- It’s also possible to pre-compile a traffic management action based on these parameters, for constant enforcement (eg. Elephant flow management)



Overall Toolbox

Basis for building use cases



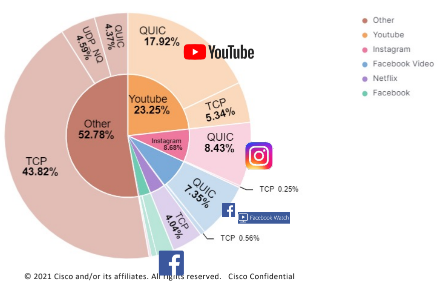
Visibility

Use Case : Monitoring and analytics

Network Traffic by Volume and Flows

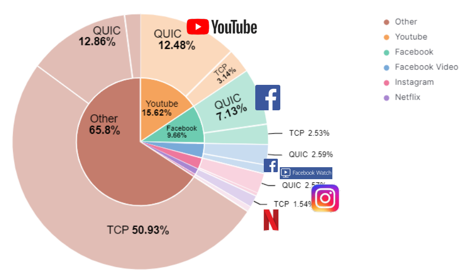
Overall Volume by Apps

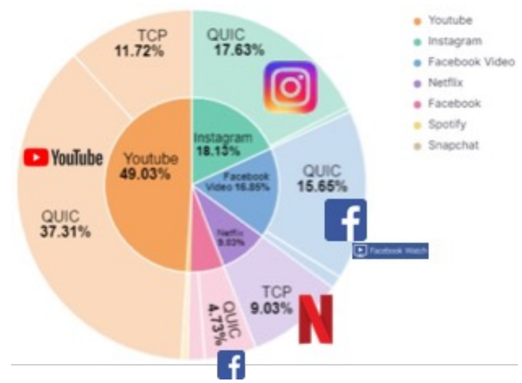
Big 5 is 48% of traffic
QUIC is 40% of traffic
"other traffic" still largely TCP, QUIC now visible (4.3%).

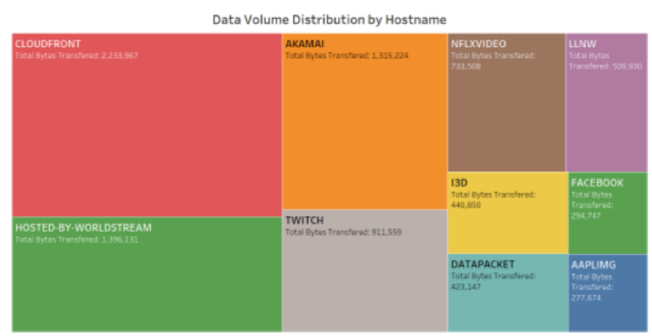


Total Flows by Apps

Lots of TCP sessions (likely IOT related, transactional related)
Big 5 QUIC sessions are very targetted and high efficiency
(video related behaviour)







CDN

Hosting

Gaming

Video Streaming

Profile aligned with Fixed Broadband traffic (browser driven traffic)

- Monitor all flows
- Infer information for Source (DNS, SNI/eSNI), CDN (ECH), Flow Type (Time domain behaviour)
- ELK (elastic Search, Logstash, Kibana) analytics engine
- Extensible to enriched CDR production

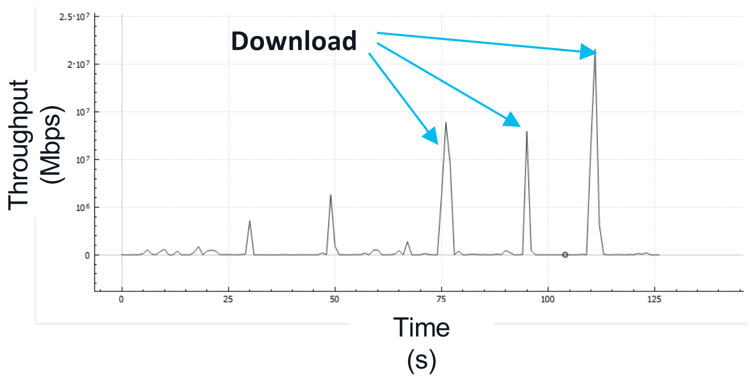
QUIC : 41%

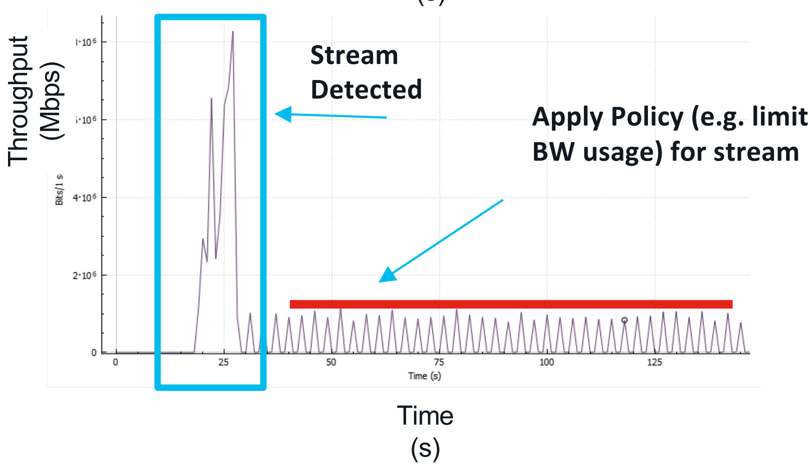
TCP: 53%

UDP (other): 6%

Custom Policy Enforcement

e.g. Differentiate between ”download” and “streaming” (within same app)



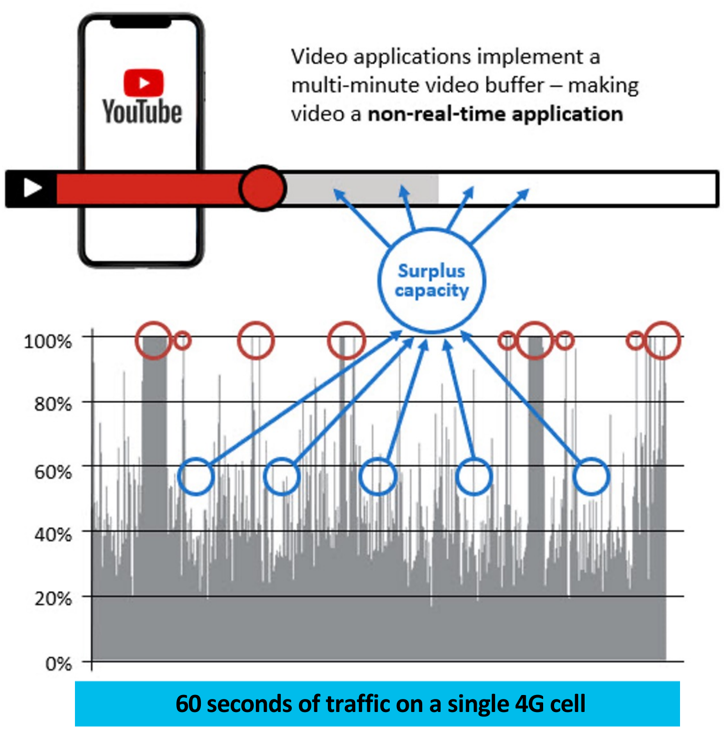


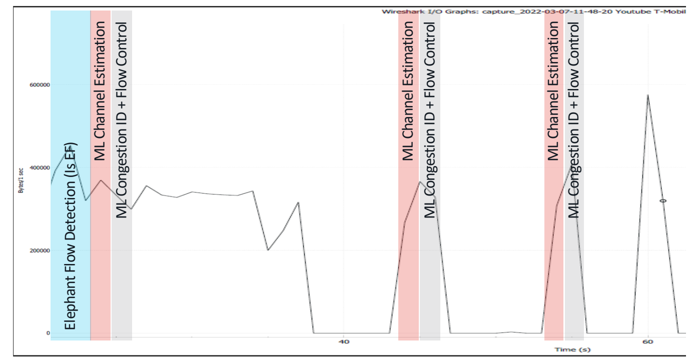
- Same Source/Destination Address
- Differentiate between download versus streaming *on the same SA/DA*
- ***Apply Policy per flow type, e.g.***
  - ***Download Policy: no action***
  - ***Streaming Policy: Limit to set BW profile (police/buffer/…)***

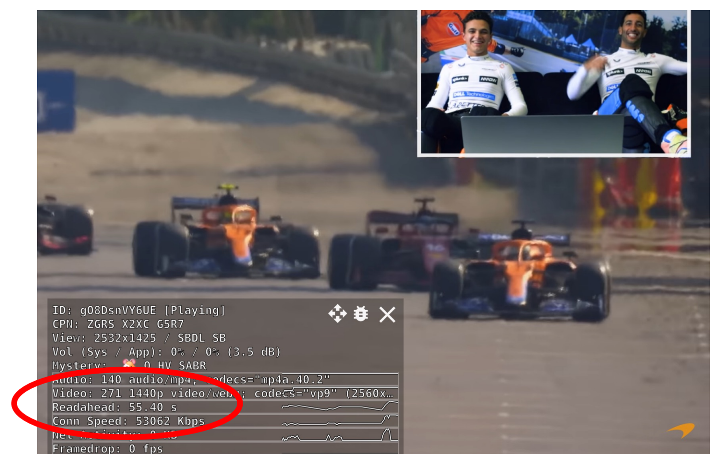
Time Domain shaping

User Experience optimization under congestion

Congestion inference determines which links are congested and which flows are impacted
Elephant Flow Detection identifies which (QUIC or not) Flows can be managed.
Then Machine Learning determines if that Flow is being delivered during congestion (red circle) and require Flow Control or not (blue circle)



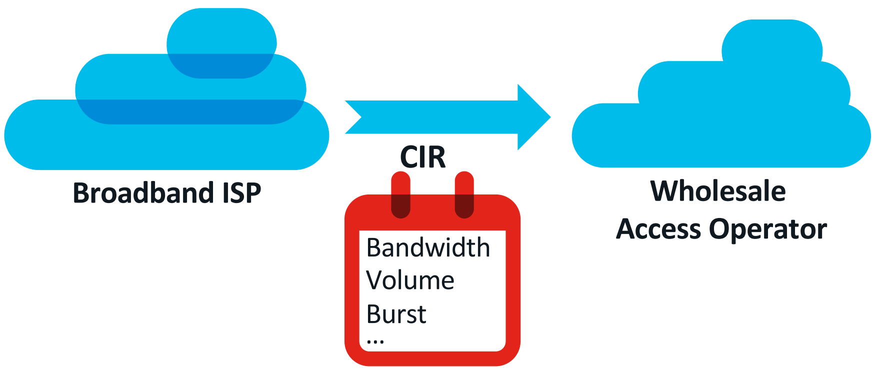




Time domain shaping

User Experience Optimization within SLA Boundaries

Situation

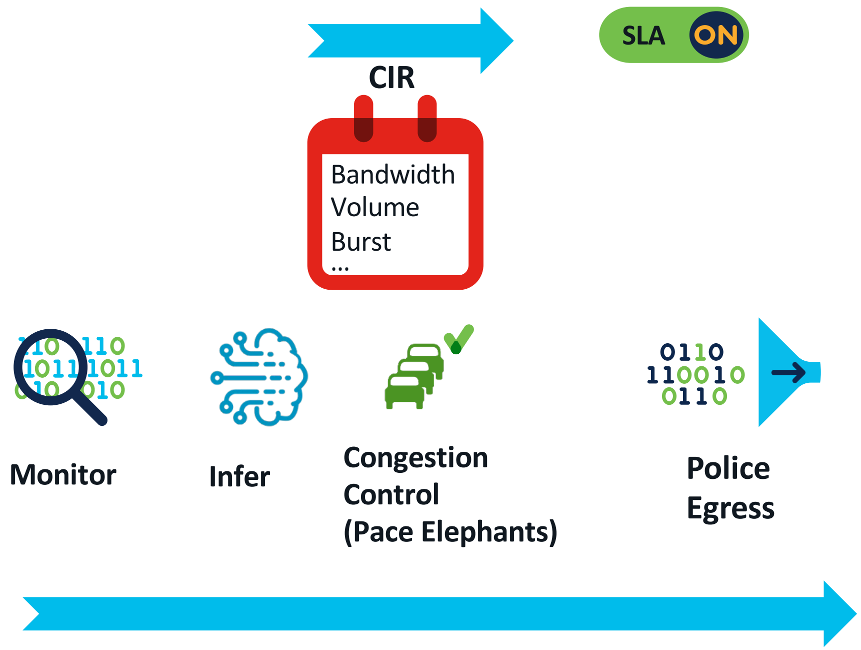


Conform to SLA results in predictable cost ✓

Violate SLA results in additional cost ✗

Indiscriminate Policing leads to bad user experience

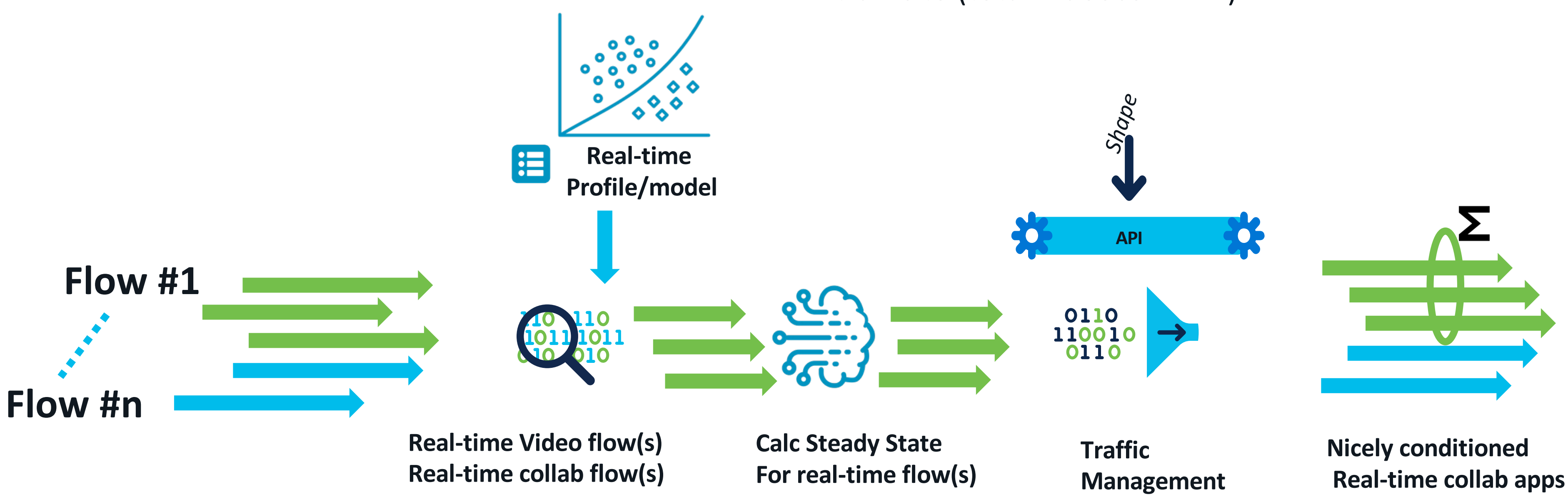
Solution



- ✓ ***Conform to SLA***
- ✓ ***Ensure QoE for every user***
- ✓ ***Fair use capability***

Use Case : Protecting Real-time Traffic

Observe traffic, detect videoconferencing stream, measure steady state Bandwith usage of video conf stream, shape traffic to (total-videoconf BW)



Real-time collab flow

Any other flow

Summary

- Traffic is encrypted, application controlled, and obfuscated
- H3/Quic/UDP/DOH stack is on the rise and here to stay
- Networks need an IP flow centric approach that scales

The bridge to possible

Thank you