# Low Latency, Low Loss, Scalable Throughput (L4S) for Video

## MHV 2024

**Presented by Glenn Deen**

**Comcast Distinguished Engineer**

COMCAST

# Agenda

- Latency, why it matters
- L4S Technology Introduction
- L4S and Video
- L4S Resources
- Q&A

# L4S at MHV 2024

L4S Talks at IETF, NANOG, etc. have focused on the network layer

Today we will be looking at L4S at the Video Application layer

# Latency: what is it & why it matters

Everything we believe about Internet performance is (mostly) wrong.

Bandwidth (marketed as speed) is important but is no longer the dominant performance factor.

Nearly all end user Internet QoE issues can be attributed to latency.

**Problem is - no one knows what latency is. (or if they do – they are wrong)**



COMCAST

# From IDLE to WORKING Latency

Changing Terms, Changing Mindsets

- When you think of “latency” – think of “**delay**”, “**responsiveness**” or “**response time**”

- **Idle Latency**: today’s artificial measure of responsiveness when a network connection is unused

- **Working Latency**: real-world measure of responsiveness when a network connection is being used (under load – normal traffic volume)

# Latency: the X-Factor of QoE

- It's a *latency* problem (not bandwidth):
  - Zoom dropping or jerky video/audio
  - Laggy/slow gaming
  - WiFi slow downs when several people are online
  - Poor VoIP quality
  - Delays loading web pages

- We used to think there was a tradeoff: EITHER high throughput OR low latency
  - Turns out we can have both!!

# Two technologies for Low Latency Networking

1. Low Latency, Low Loss, Scalable Throughput (**L4S**)
   - Intended for: high bitrate latency sensitive flows
   - Server requirements: scalable congestion control algorithm, such as TCP-Prague, so the sender will act on Congestion Experienced (CE) marks

2. Non-Queue-Building Per Hop Behavior (**NQB**)
   - Intended for: low bitrate latency sensitive flows
   - Server requirements: None

References:
https://www.rfc-editor.org/rfc/rfc9330.html
https://www.rfc-editor.org/rfc/rfc9331.html
https://www.rfc-editor.org/rfc/rfc9332.html
https://datatracker.ietf.org/doc/draft-ietf-tsvwg-nqb/
https://github.com/jlivingood/IETF-L4S-Deployment/blob/main/App-Developer-Guide.md
https://github.com/jlivingood/IETF-L4S-Deployment/blob/main/Network-Config-Guide.md



COMCAST

# Bottlenecks, Latency, Dual Queues and L4S

**Packet Queues** can be bottlenecks where congestion contributes to latency

**L4S** [RFC9330,9331,9332] marks packets with the ECT[1] bit in the ECN header

**L4S marked packets** can be managed by more appropriate approaches such as TCP Prague to better meet the application's needs

**L4S** creates a new additional L4S Queue in addition to the "Classic" Queue

**L4S Queue** exists as a **Dual Queue** in parallel to the Classic Queue

- Permits Alternate Congestion Control for L4S marked packets
- Classic congestion controls continue to manage unmarked packets

# L4S Smoother Flow

L4S Queue signaling when congested helps avoid dropping packets

- Signals back to the sender alerting of the congestion
- Enables sender to react with Self Restraint
- Removes burstiness resulting in smoother-more consistent flow

Scalable throughput

- L4S keeps things simple: a signal dual queue, no special per flow controls

Smooth reliable & consistent application traffic flow is the result

- including for high bandwidth flows
- Exactly what video applications want – especially for live

# Dual Queue Basics

- There is always a bottleneck link on the end-to-end path
- Two fundamental traffic types = two separate network queues at bottlenecks
- Can be incrementally deployed – does not depend on full internet-wide deployment
- Loose coupling between layers (between network & app developer)

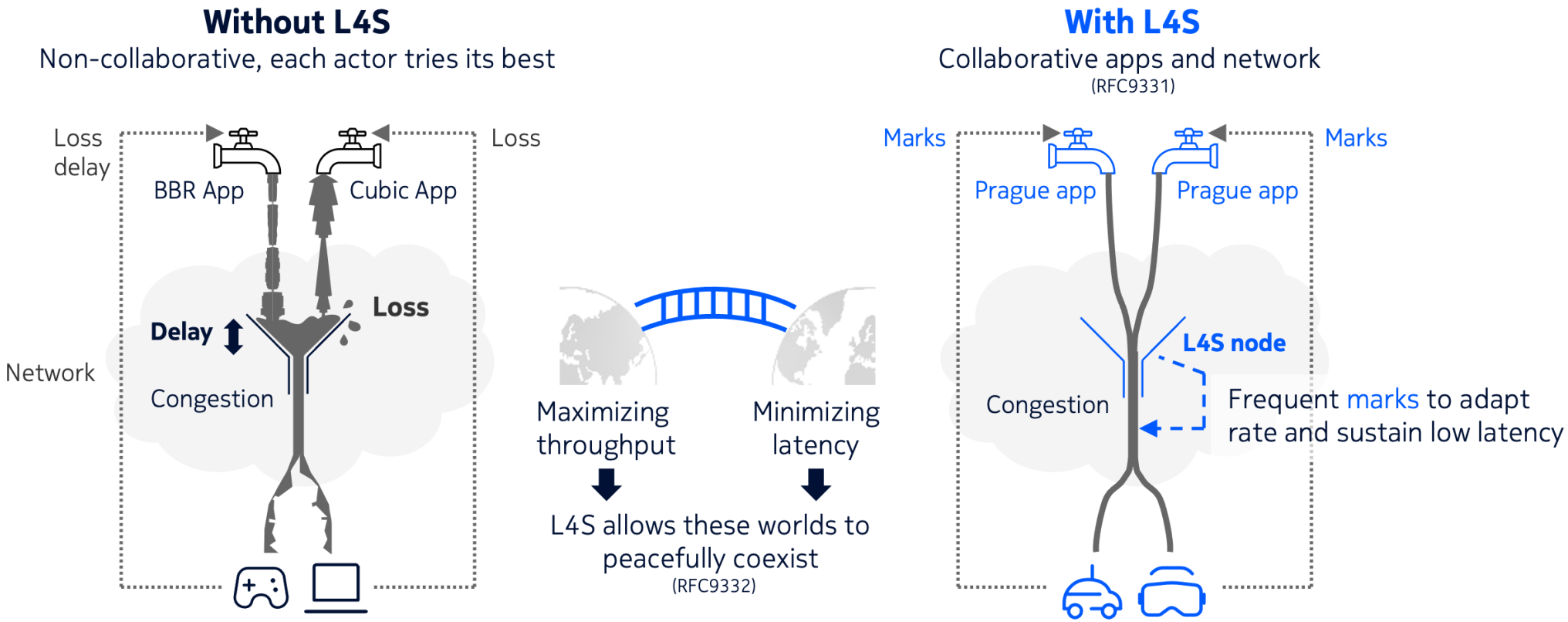
**Deployment Principles:**

- Only apps mark traffic, not the (access) network (e.g., with a DPI middlebox)
- Any app/edge provider should be able to use it
- Network operators just need to let the ECN and/or DSCP marks to flow across the network without bleaching or other modification

# Dual Queue Basics

Establish 2nd network queue at bottlenecks, with a shallower queue depth

(image from Nokia)

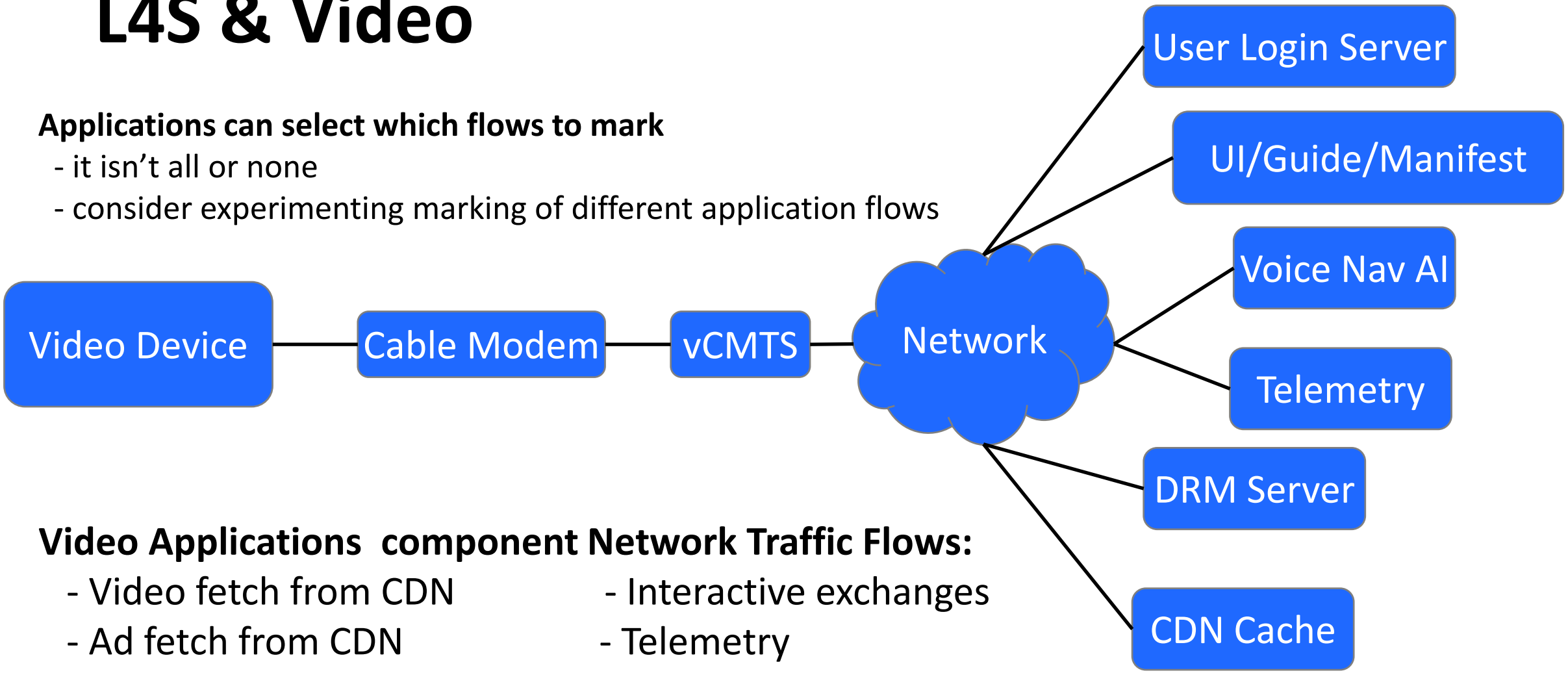





COMCAST

# L4S & Video

**Applications can select which flows to mark**
- it isn't all or none
- consider experimenting marking of different application flows




**Video Applications component Network Traffic Flows:**

- Video fetch from CDN
- Ad fetch from CDN
- DRM service exchanges
- UI/Guide/Manifests.
- Interactive exchanges
- Telemetry
- User Authentication
-Voice Nav AI



COMCAST

# L4S Resources

L4S Technical Specification IETF RFCs
https://www.rfc-editor.org/rfc/rfc9330.html
https://www.rfc-editor.org/rfc/rfc9331.html
https://www.rfc-editor.org/rfc/rfc9332.html

IETF Non-Queue-Building Per Hop Behavior (**NQB**)
https://datatracker.ietf.org/doc/draft-ietf-tsvwg-nqb/

L4S Guides
https://github.com/jlivingood/IETF-L4S-Deployment/blob/main/App-Developer-Guide.md
https://github.com/jlivingood/IETF-L4S-Deployment/blob/main/Network-Config-Guide.md

L4S on iOS
https://developer.apple.com/documentation/network/testing_and_debugging_l4s_in_your_app