

# Agenda

- 14:00 **Welcome & Start** -Gorry Fairhurst, University of Aberdeen
- 14:05 Status of QUIC
  - **Overview of Standards, and operation of QUIC, G Fairhurst (15 mins)**
  - Q+A
- 14:25 Overview of QUIC Satellite Performance
  - **QUIC Performance (8 mins) Lorena Albiol Schnitger, Indra**
  - **QUIC SATCOM Operations (8 mins) Chi-Jiun Su, HNS**
  - Q+A
- 14:45 Overview QUIC Satellite-Activities
  - **ACK Reduction (8 mins) Ana Custura, University of Aberdeen**
  - **QUIC and Next Gen SATCOM (8 mins) Joerg Deutschmann**
  - **Accelerating Start-up (8 mins) Nicholas Kuhn, CNES**
  - Q+A
- 15:10 Open discussion on challenges and opportunities (20 mins)

# QUIC: Its impact on satellite equipment vendors and operators

Gorry Fairhurst  
Aberdeen University



**QUIC** :

a new ~~web~~ transport

04

# What is QUIC?

Explaining the Origins of QUIC

# What is QUIC?



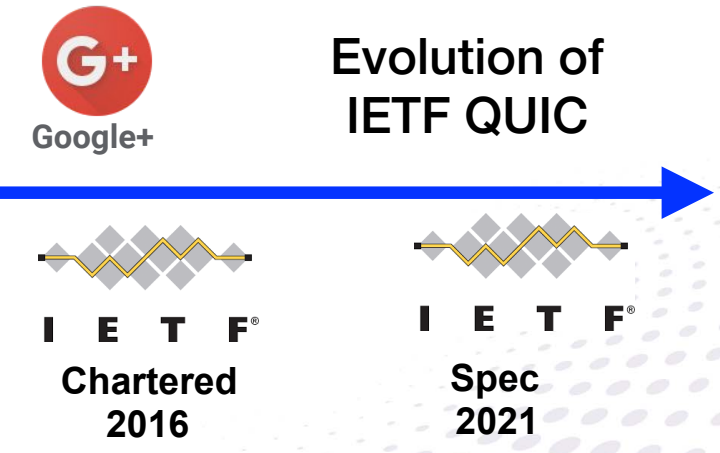
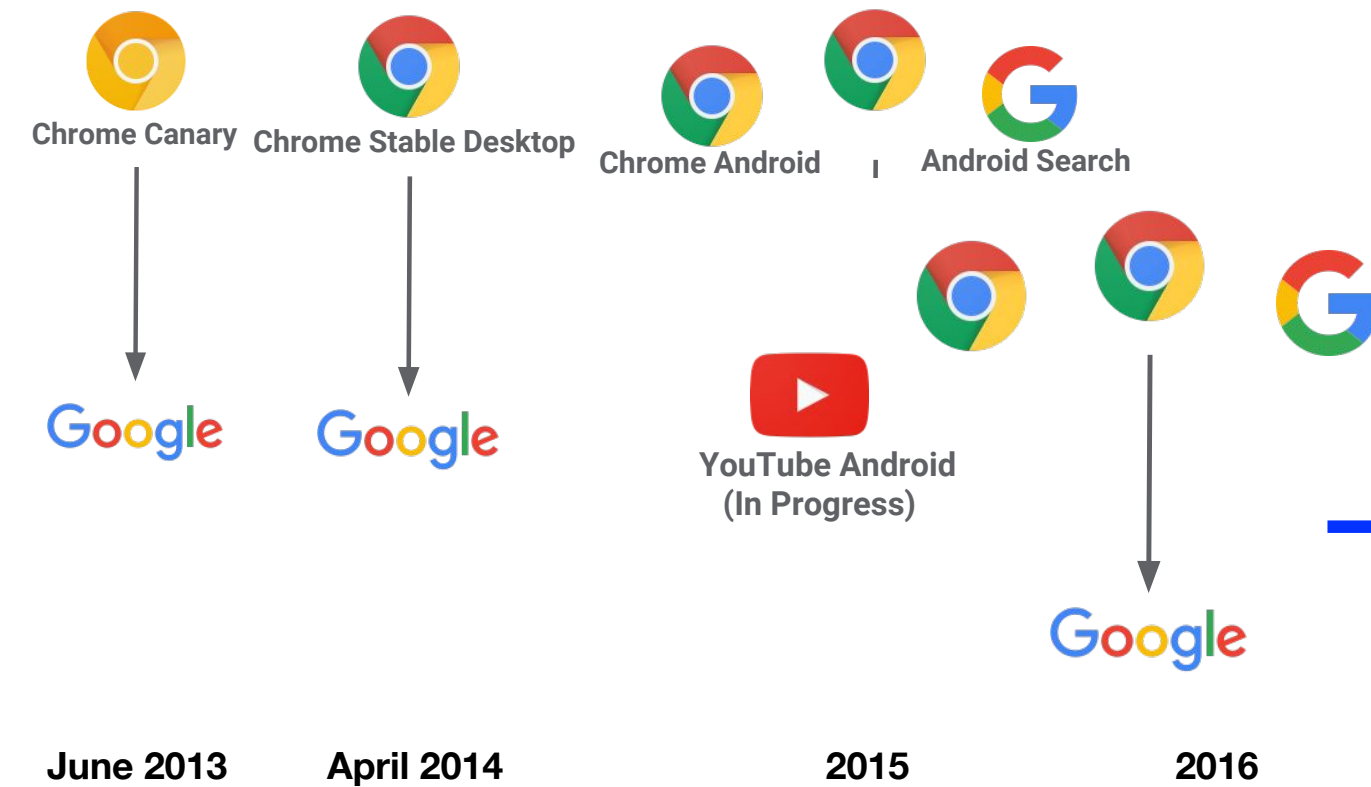
	HTTP/1	HTTP/2
Year	(1.1) 1991	2015
Data	ASCII	Binary
Security	TLS optional	TLS 1.2 always
Transport	TCP	TCP
Stream	1 stream over a TCP Connection	Multiple streams over one connection

TCP Spec started 40 years ago



# Evolution of QUIC...

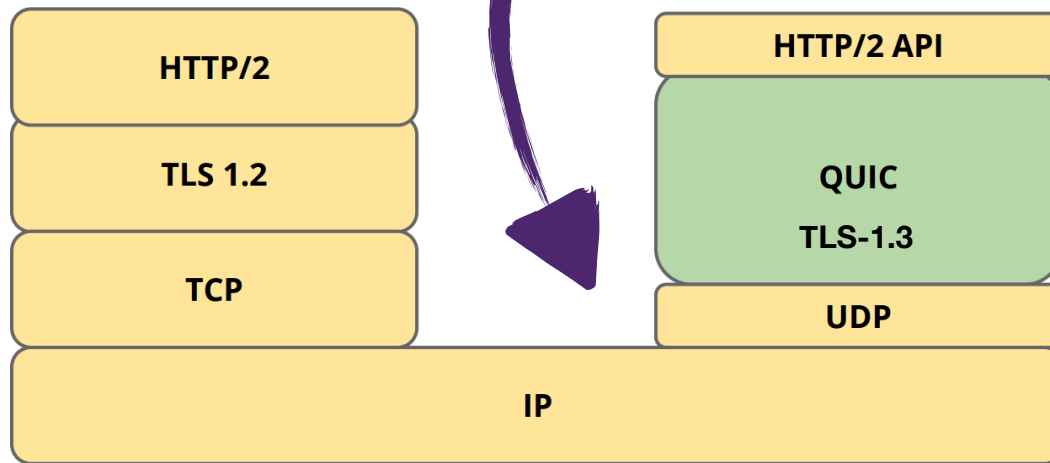
## Evolution of Google Chromium



>90% bytes from Chrome are now QUIC

# QUIC replaces TCP with UDP and TLS-1.3

UDP has allowed rapid update and deployment

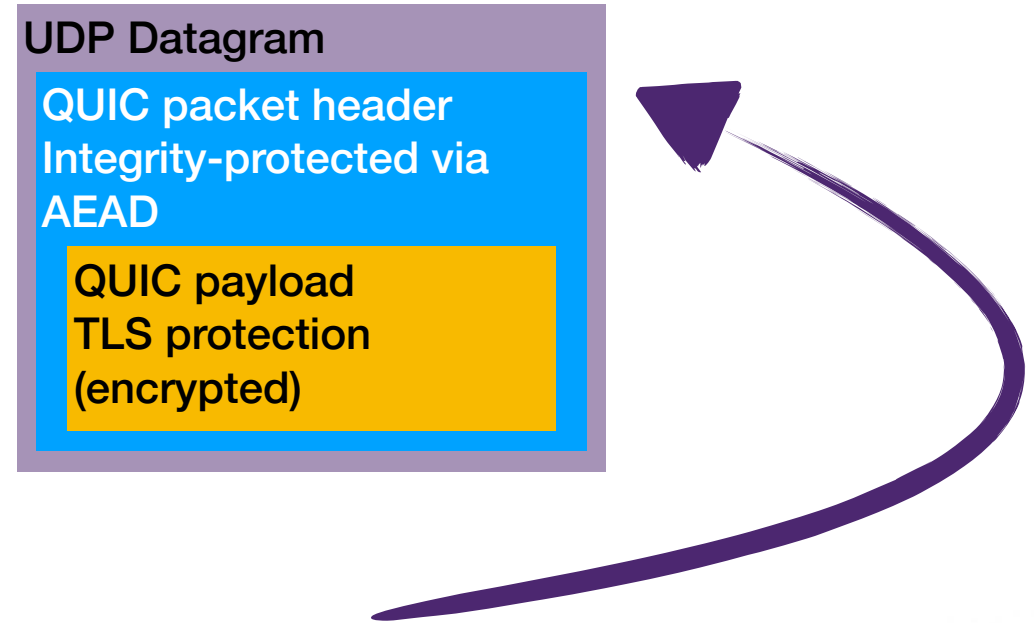


QUIC provides security and privacy protection  
Encryption prevents unauthenticated middleboxes changing packets

# What is QUIC?

	HTTP/1	HTTP/2	HTTP/3
Year	(1.1) 1991	2015	2021
Data	ASCII	Binary	Binary
Security	TLS optional	TLS 1.2 always	TLS 1.3 within IETF QUIC
Transport	TCP	TCP	QUIC over UDP
Stream	1 stream over a TCP Connection	Multiple streams over one connection	Multiple streams over multiple transport streams

# QUIC Packets



A **QUIC packet** is sent in the payload of a UDP Datagram

The payload is *always encrypted* using TLS-1.3

Many packets can be *coalesced* into a datagram (there are rules)

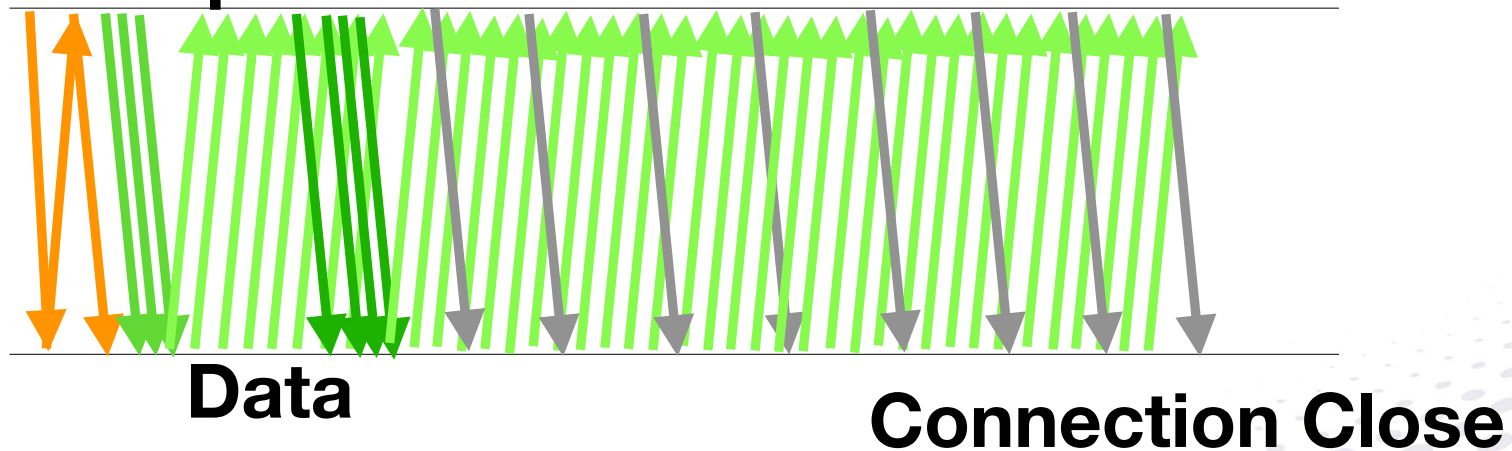


# QUIC Connections

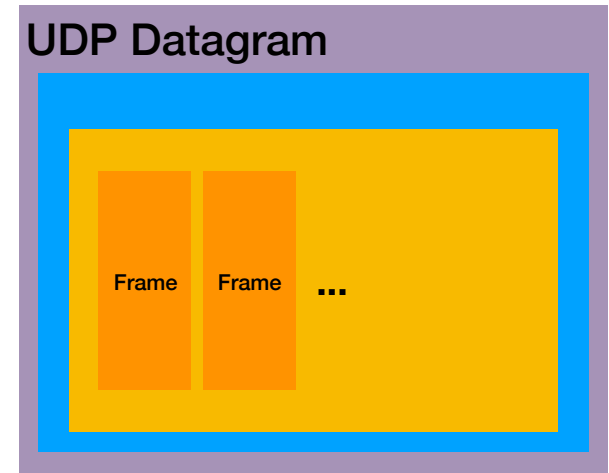
QUIC uses *connections*

Each connection starts with a *handshake phase*, similar to TCP  
A connection can migrate to another network path after it connects

## Handshake Request



# QUIC Packets carry Frames

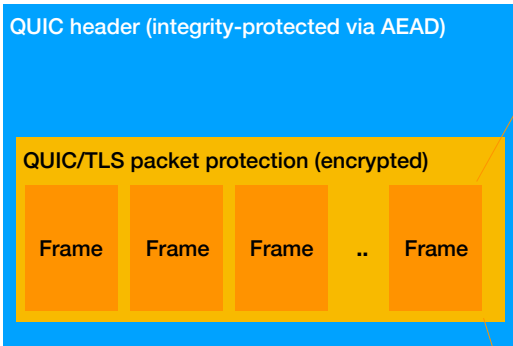


One or more **QUIC Frames** are sent in a QUIC Packet

Data is sent in **STREAM Frames**

Packets can also carry other types of **QUIC Frames**

# QUIC Frames

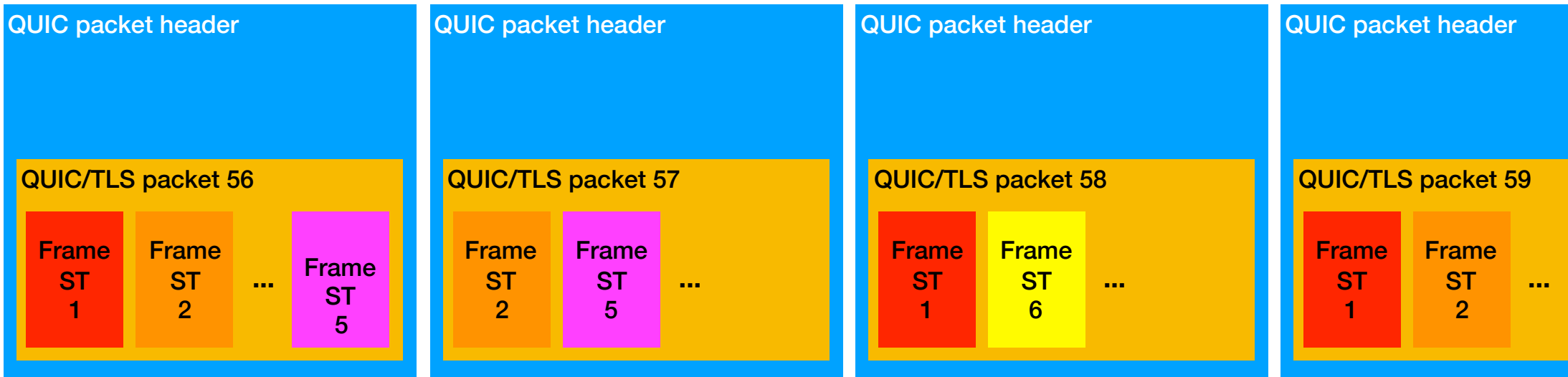


Type	Name	
0x00	PADDING	IH01   NP
0x01	PING	IH01
0x02	ACK	IH 1   NC
0x03	ACK	IH 1   NC
0x04	RESET_STREAM	01
0x05	STOP_SENDING	01
0x06	CRYPTO	IH 1
0x07	NEW_TOKEN	1
0x08-0x0f	STREAM	01   F
0x10	MAX_DATA	01
0x11	MAX_STREAM_DATA	01
0x12-0x13	MAX_STREAMS	01
0x14	DATA_BLOCKED	01
0x15	STREAM_DATA_BLOCKED	01
0x16-0x17	STREAMS_BLOCKED	01
0x18	NEW_CONNECTION_ID	01   P
0x18	RETIRE_CONNECTION_ID	01   P
0x1a	PATH_CHALLENGE	01   P
0x1b	PATH_RESPONSE	ih01
0x1c-1d	CONNECTION_CLOSE	1

Packets  
 I: Initial  
 H: Handshake  
 0: 0-RTT  
 1: 1-RTT

Types  
 ih: Only in Initial or Handshake  
 N: not ack-eliciting  
 C: do not count for CC  
 P: can probe new paths  
 F: flow controlled

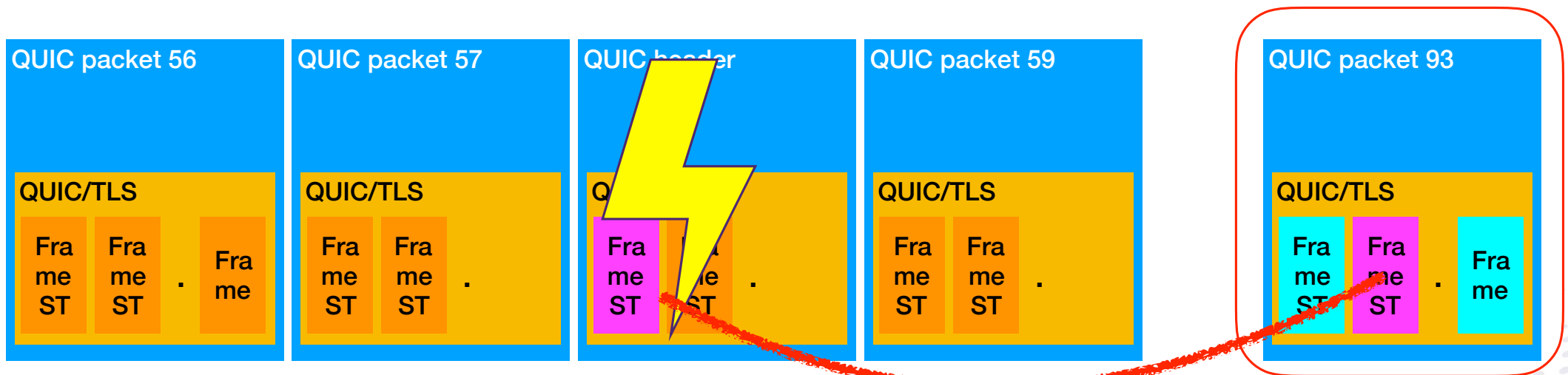
# QUIC: Multi-Streaming



Stream can be created, and deleted as needed - they are never re-used  
STREAMS are **Congestion Controlled** (normally)  
STREAMS are **Flow Controlled**

# All STREAMS are Acknowledged

Endpoints *all packets* received are *acknowledged*  
Not all QUIC Streams are *reliable*



**Packets are NOT retransmitted** (the frames carried in packets CAN be retransmitted)



2020 v1 Spec  
Nov 2020 IETF Last Call  
2021 Publish as RFCs

250+ pages

***draft-ietf-quic-applicability***

*Applicability of the QUIC Transport Protocol*

***draft-ietf-quic-http***

*Hypertext Transfer Protocol (HTTP) over QUIC*

***draft-ietf-quic-invariants***

*Version-Independent Properties of QUIC*

***draft-ietf-quic-manageability***

*Manageability of the QUIC Transport Protocol*

***draft-ietf-quic-qcram***

*Header Compression for HTTP over QUIC*

***draft-ietf-quic-recovery***

*QUIC Loss Detection and Congestion Control*

***draft-ietf-quic-tls***

*Using Transport Layer Security (TLS) to Secure QUIC*

***draft-ietf-quic-transport***

*QUIC: A UDP-Based Multiplexed and Secure Transport*

<https://tools.ietf.org/wg/quic/>

# QUIC v1 is being deployed



>15 implementations on their way:

e.g.

chrome (**google**)

mvfst (**Facebook**)

AppleQUIC (**Apple**)

ats (**Apache**)

lsquic

mozquic (**Mozilla**)

ngtcp2

ngx\_quic

Pandora

f5

picoquic

quant

quiche(**Cloudflare**)

quicly (**Fastley**)

quicr & Quinn

Winquic (**Microsoft**)

quic-go

nghq

nghttp3

ls-qpack

nghttp3

# What's could be in QUIC 2?



## **Good things on the way:**

Alternative congestion controllers

Partial Reliability - e.g. datagrams for multimedia

Secure Tunnels and Proxies (see new IETF Masque Working Group )...

Migration Support (already in Spec)

Multipath support (?To be discussed at IETF-109?)

Forward Error Correction (?Maybe?)

## **Logging, and other useful stuff**

**Considerable interest in using QUIC for other applications**

Still all being discussed - no decisions yet.

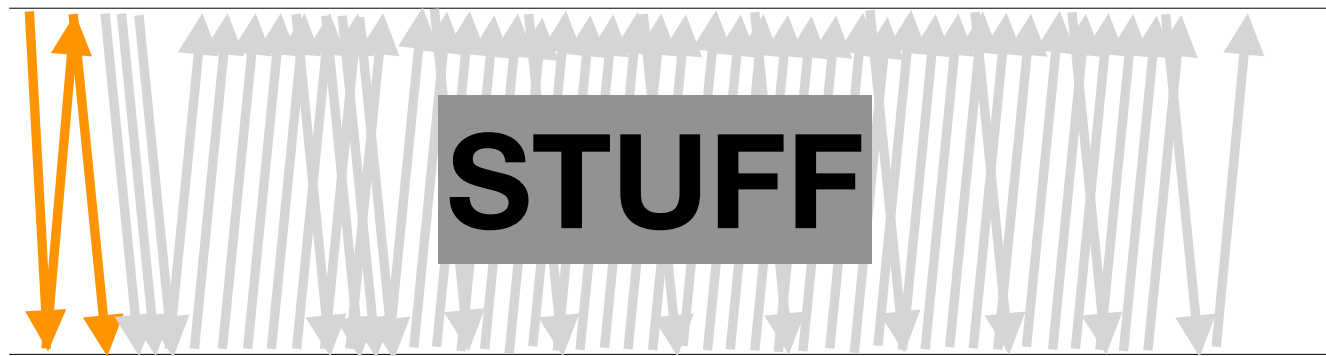


# What do I need to do now that everyone (?) uses QUIC

A Set of Slides Explaining Network Implications

# The network can't inspect QUIC traffic

Handshake

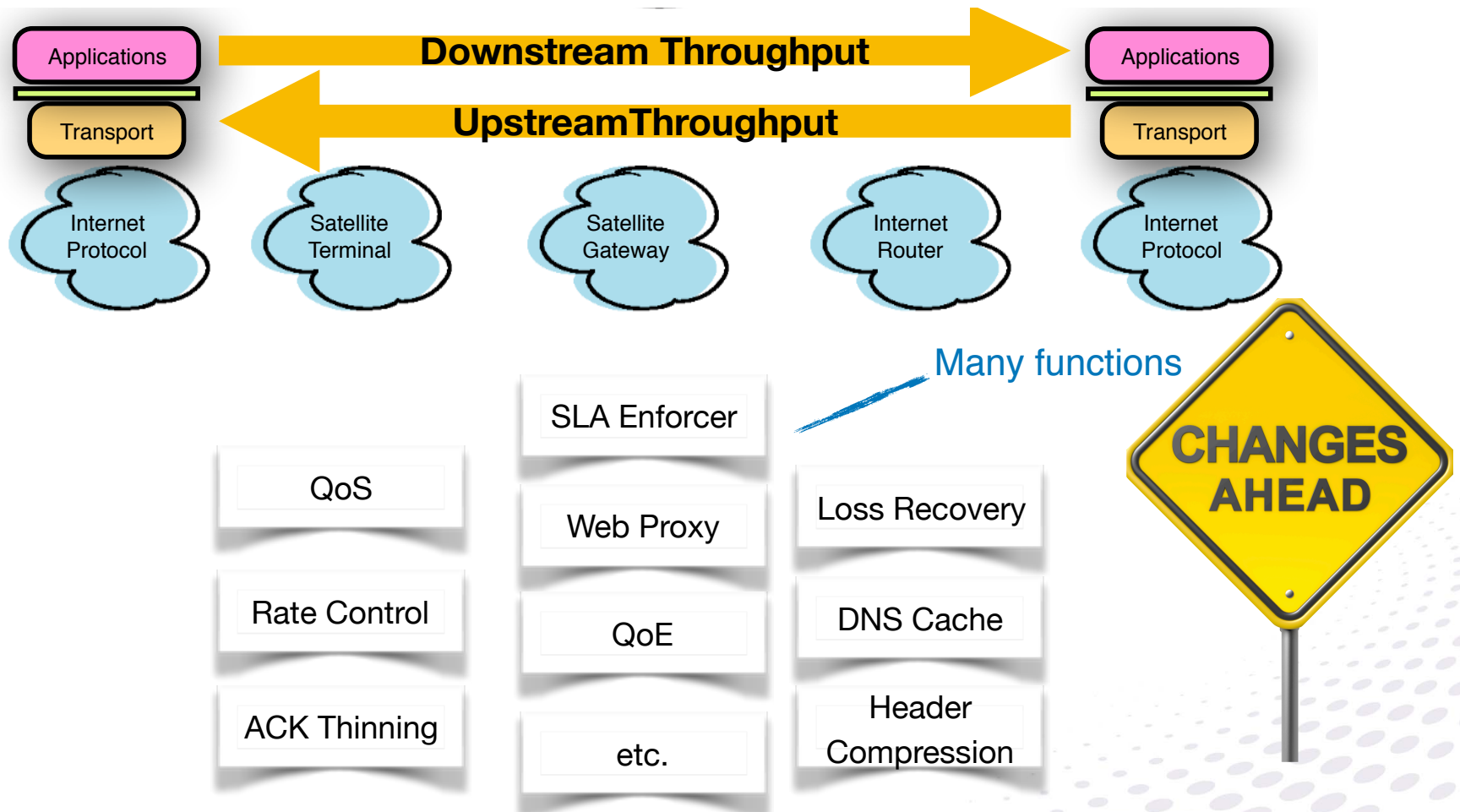


The benefits of encryption have side-effects:

- Can't debug or measure with common network tools

- Need access to keys from the handshake, or server logs

# TCP Performance Enhancing Proxies



Middleboxes need explicit authentication!

# QUIC changes the way networks are used and how they need to be operated

## Operators no longer control trade-offs

PEP methods do not work as they have done

Firewalls and Malware detection

Network Management

Compliance and Troubleshooting

Passive Measurement

Legal Intercept

Still can use pattern recognition of traffic (for now, QUIC allows padding)



[draft-ietf-tsvwg-transport-encrypt](#)  
[draft-ietf-quick-manageability](#)

# QUIC and Broadband Satellite

😊	QUIC has latest transport improvements - can be easy updated
😊	QUIC has latest http improvements in H3 - can be easy updated
😞	Current QUIC optimised for main use-case
😊	QUIC will continue to evolve, and this may benefit satellite broadband users
😞	Operators no longer control trade-offs - most changes need updates to clients/servers

## What next?

**The QUIC WG is exploring new proposals and extensions**

**Features helpful for satellite could be discussed/developed/deployed**

**There might be synergy here with mobile use-cases....**

**.... What issues are most important?**

**.... How do we move forward?**