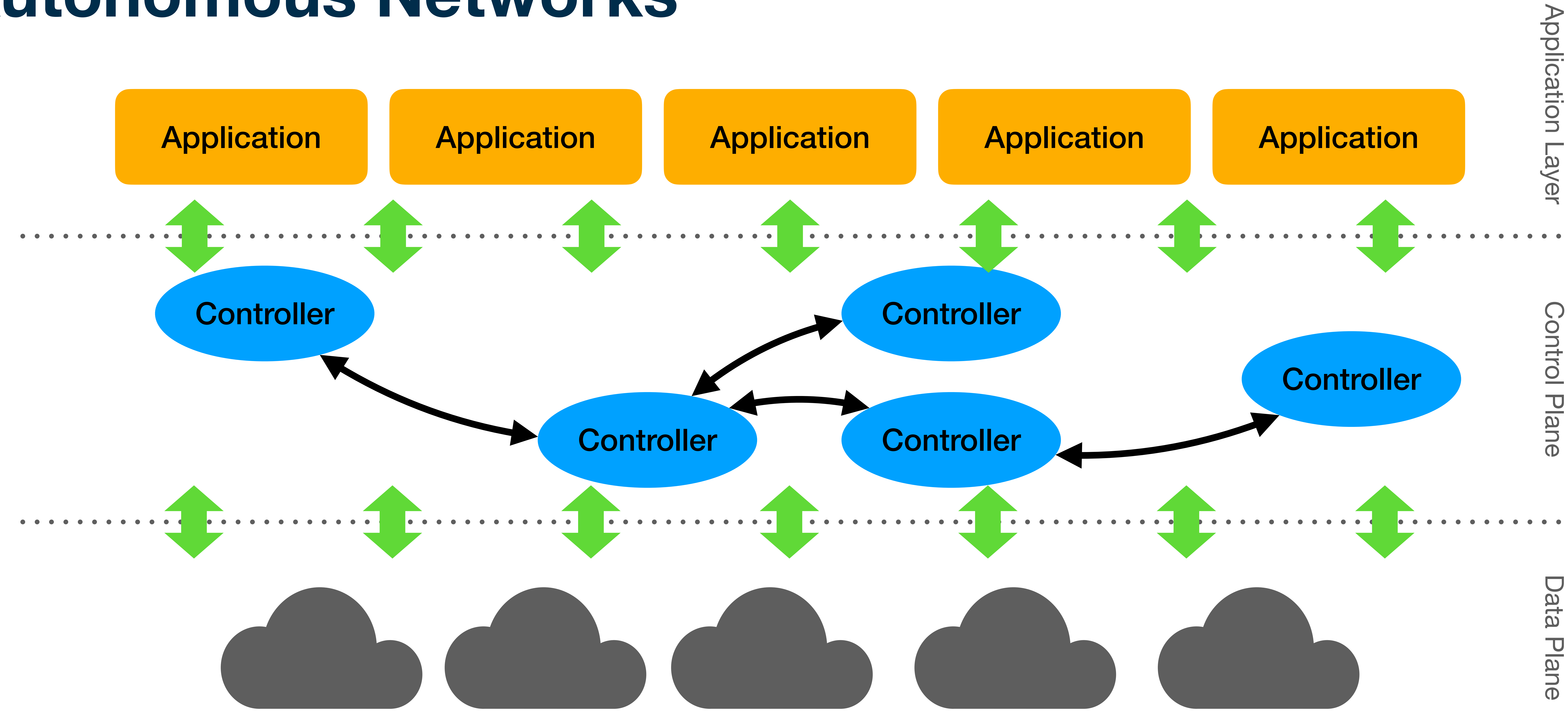
Supporting Evolution, Experimentation, and Adaptation with Parseable Standards Documents

Stephen McQuistin

Scottish Autonomous Networked Systems (SANS) Workshop
12th—13th December 2022

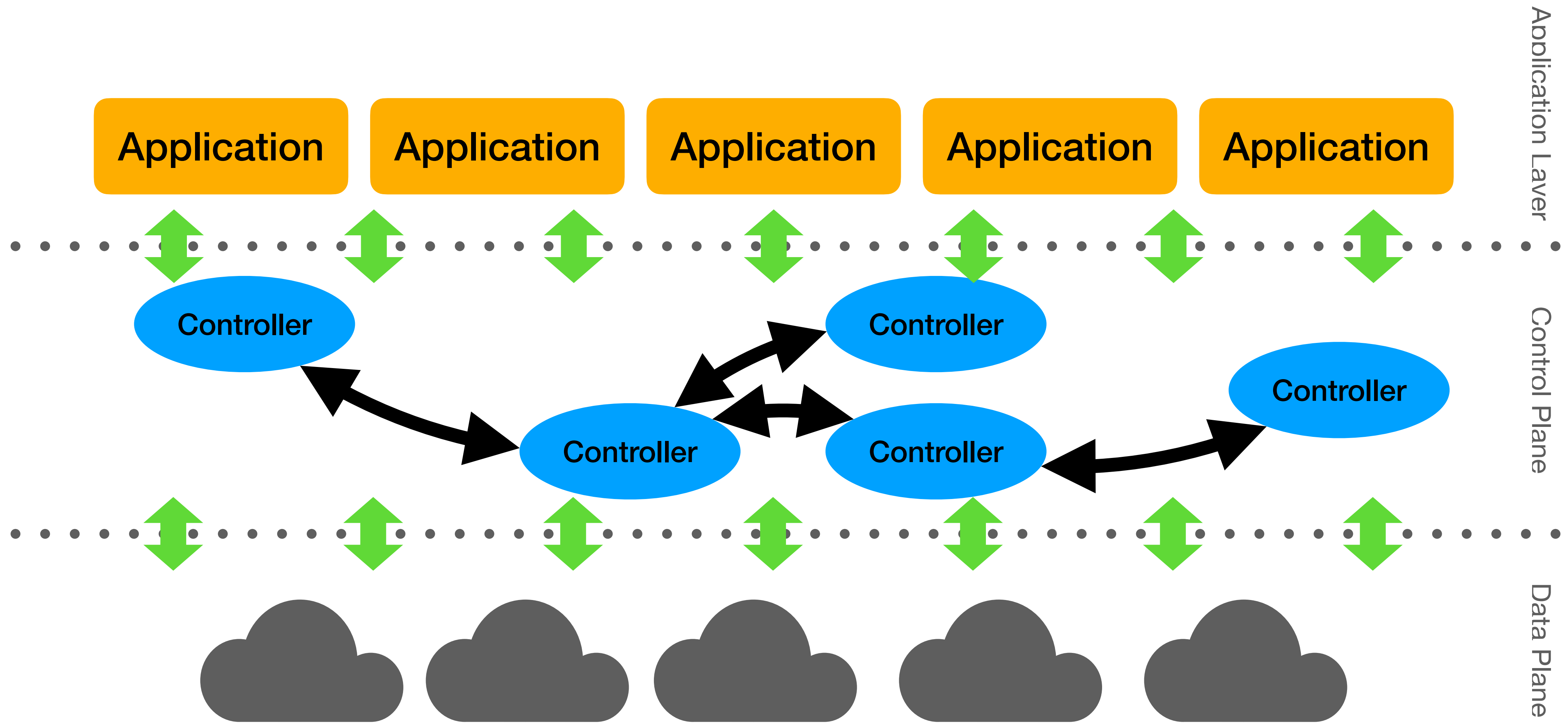
Autonomous Networks



Autonomous Networks

Adaptation

Evolution

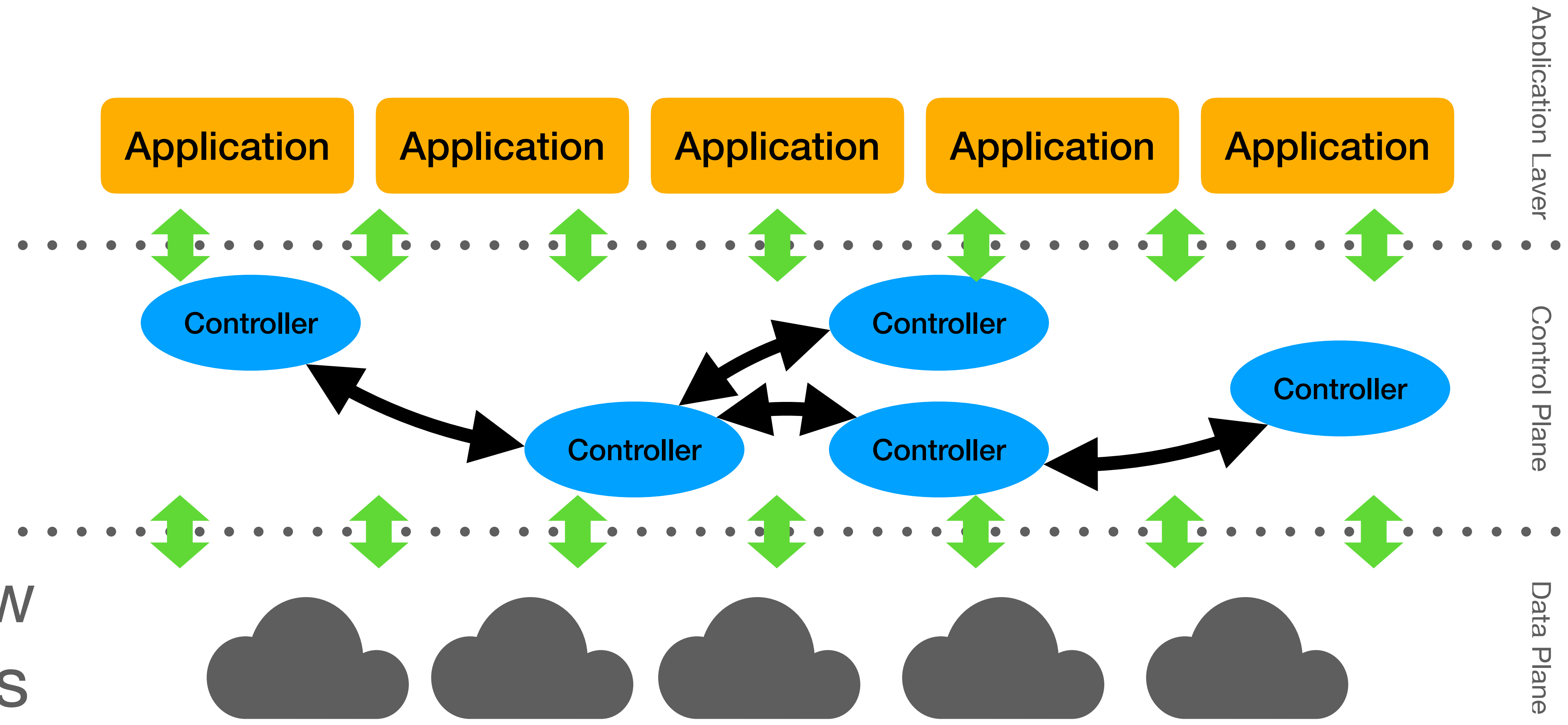


Experimentation

Autonomous Networks

Responding to changes in the environment to maintain service

Adaptation



Evolution

Improving over time as new applications and use cases emerge

Experimentation

Trying new approaches to improve performance and other metrics

Autonomous Networks

Responding to changes in the environment to maintain service

Adaptation

Evolution

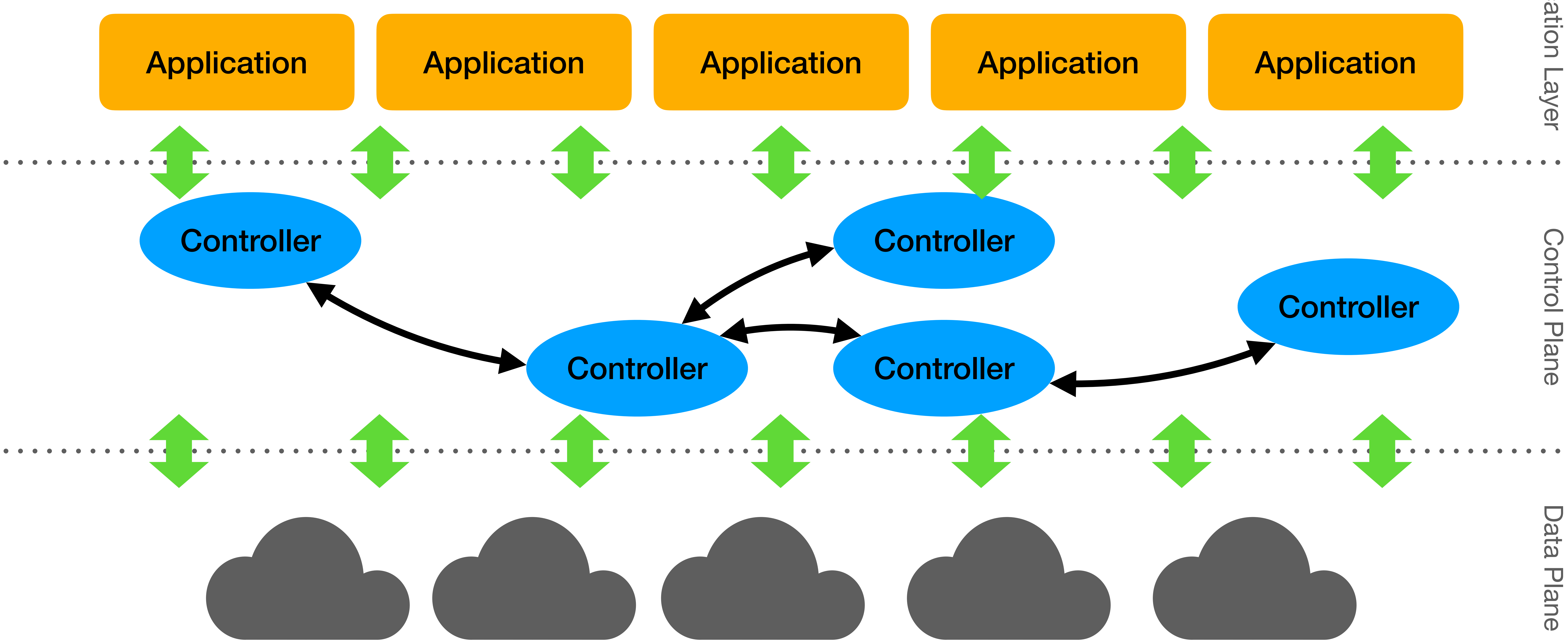
Improving over time as r
applications and use cas
emerge

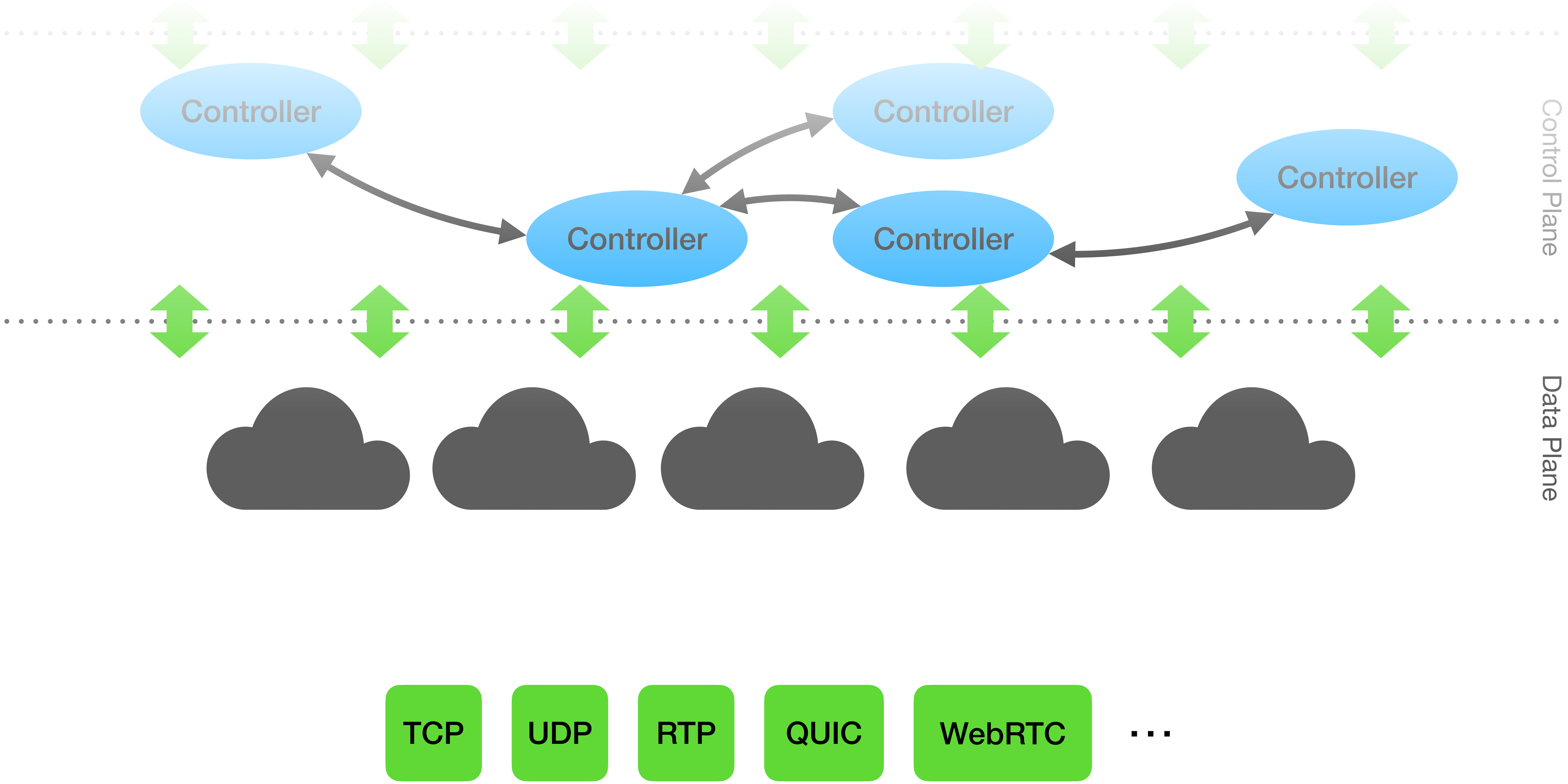
The core attributes of autonomous networks require rapid deployability

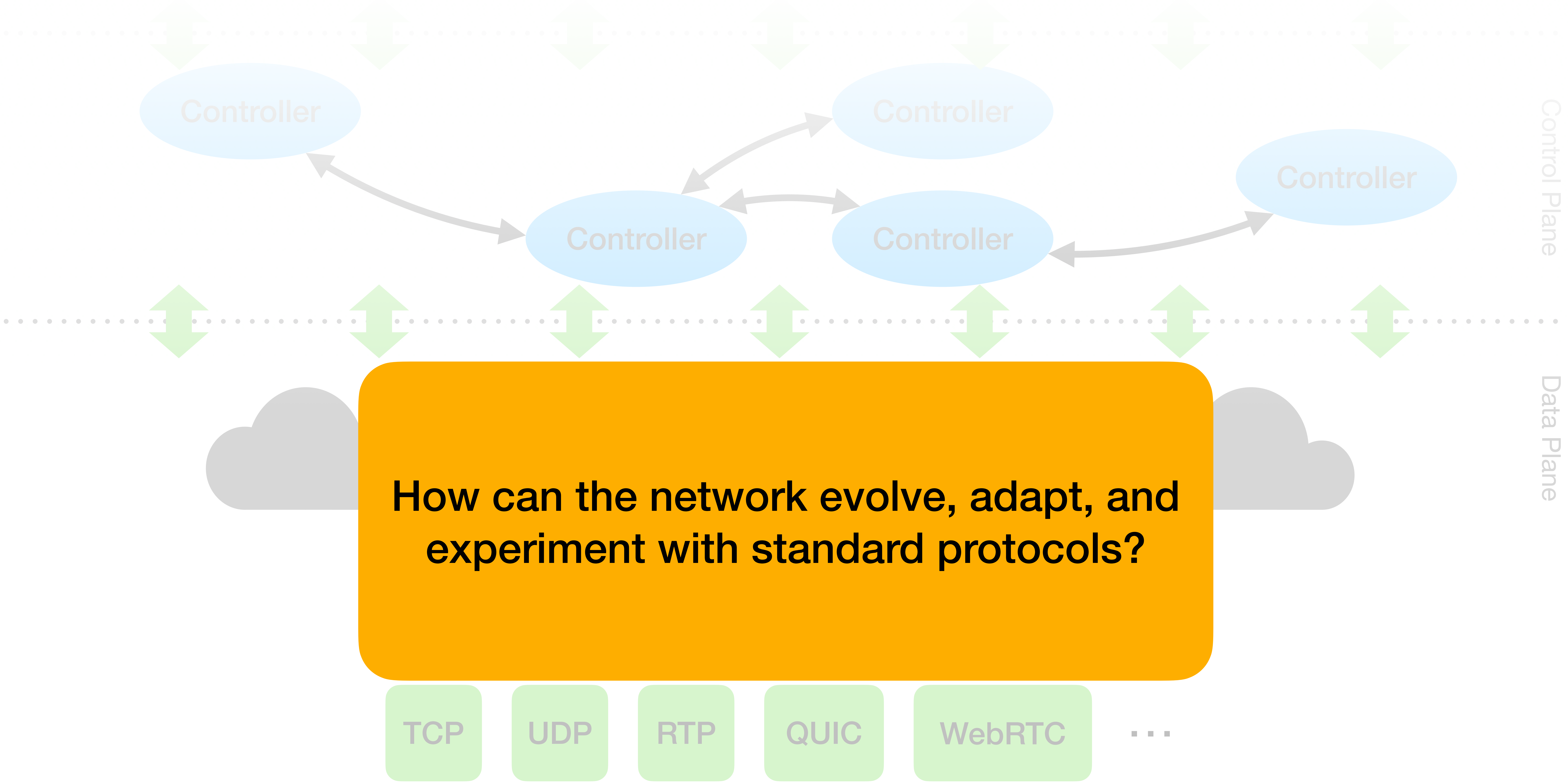
Experimentation

Trying new approaches to improve performance and other metrics

Autonomous Networks

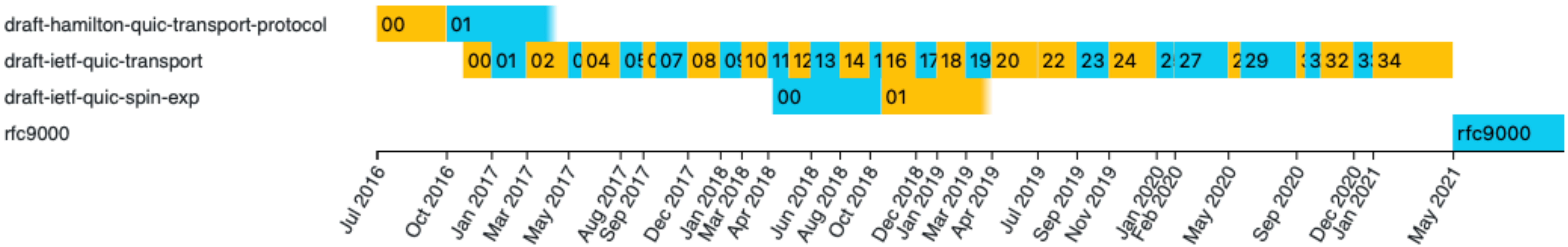



Controller
Controller
Controller
Controller
Controller
Control Plane
Data Plane
TCP
UDP
RTP
QUIC
WebRTC
...

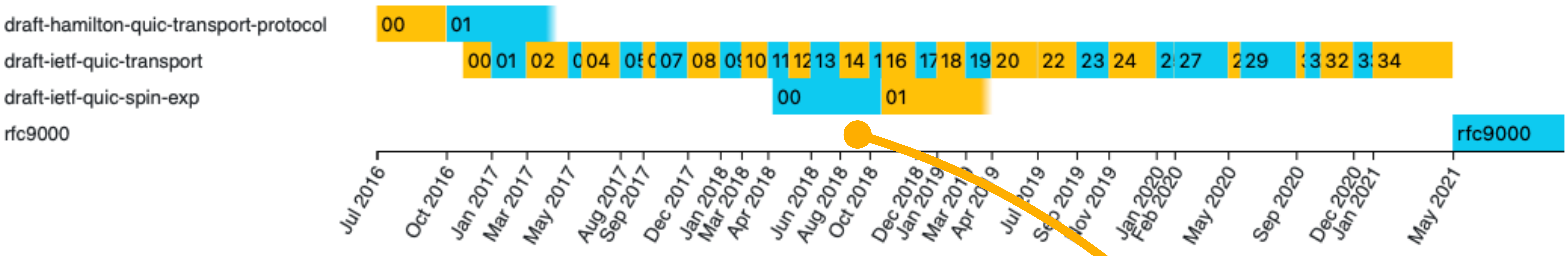


How can the network evolve, adapt, and experiment with standard protocols?

Protocol standardisation takes time



Protocol standardisation takes time



QUIC took almost five years to work its way through the IETF

… and produce standards that look like this

```
Internet Engineering Task Force                             W. Eddy, Ed.
Internet-Draft                                               MTI Systems
Obsoletes: 793, 879, 2873, 6093, 6429,                  October 27, 2020
           6528, 6691 (if approved)
Updates: 5961, 1122 (if approved)
Intended status: Standards Track
Expires: April 30, 2021


           Transmission Control Protocol (TCP) Specification
                     draft-ietf-tcpm-rfc793bis-19

Abstract

   This document specifies the Transmission Control Protocol (TCP).  TCP
   is an important transport layer protocol in the Internet protocol
   stack, and has continuously evolved over decades of use and growth of
   the Internet.  Over this time, a number of changes have been made to
   TCP as it was specified in RFC 793, though these have only been
   documented in a piecemeal fashion.  This document collects and brings
   those changes together with the protocol specification from RFC 793.
   This document obsoletes RFC 793, as well as RFCs 879, 2873, 6093,
   6429, 6528, and 6691 that updated parts of RFC 793.  It updates RFC
   1122, and should be considered as a replacement for the portions of
   that document dealing with TCP requirements.  It also updates RFC
   5961 by adding a small clarification in reset handling while in the
   SYN-RECEIVED state.  The TCP header control bits from RFC 793 have
   also been updated based on RFC 3168.

   RFC EDITOR NOTE: If approved for publication as an RFC, this should
   be marked additionally as "STD: 7" and replace RFC 793 in that role.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
```

… and produc… this


```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |       |C|E|U|A|P|R|S|F|                               |
| Offset| Rsrvd |W|C|R|C|S|S|Y|I|            Window             |
|       |       |R|E|G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Data                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Note that one tick mark represents one bit position.

Figure 1: TCP Header Format

Each of the TCP header fields is described as follows:

Source Port: 16 bits

The source port number.

Destination Port: 16 bits

The destination port number.

Sequence Number: 32 bits

The sequence number of the first data octet in this segment (except when the SYN flag is set). If SYN is set the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.

Acknowledgment Number: 32 bits

If the ACK control bit is set, this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established, this is always sent.

Data Offset: 4 bits

… and produc his


```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Source Port          |       Destination Port        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Sequence Number                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Acknowledgment Number                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Data |       |C|E|U|A|P|R|S|F|                               |
   | Offset| Rsrvd |W|C|R|C|S|S|Y|I|            Window             |
   |       |       |R|E|G|K|H|T|N|N|                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Checksum            |         Urgent Pointer        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Options                    |    Padding    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             Data                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Note that one tick mark represents one bit position.

Figure 1: TCP Header Format

Each of the TCP header fields is described as follows:

Source Port: 16 bits

The source port number.

Destination Port: 16 bits

The destination port number.

Sequence Number: 32 bits

The sequence number of the first data octet
when the SYN flag is set). If SYN is set t
the initial sequence number (ISN) and the f
ISN+1.

Acknowledgment Number: 32 bits

If the ACK control bit is set, this field c
next sequence number the sender of the segme
receive. Once a connection is established, this is always sent.

Data Offset: 4 bits

Implementation, debugging, and deployment take time too

… and produc[…]his


```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgment Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Data |       |C|E|U|A|P|R|S|F|                               |
| Offset| Rsrvd |W|C|R|C|S|S|Y|I|            Window             |
|       |       |R|E|G|K|H|T|N|N|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Checksum            |         Urgent Pointer        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

How can the network evolve, adapt, and experiment with standard protocols?

The destination port number.

Sequence Number: 32 bits

The sequence number of the first data octet […] when the SYN flag is set). If SYN is set t[…] the initial sequence number (ISN) and the f[…] ISN+1.

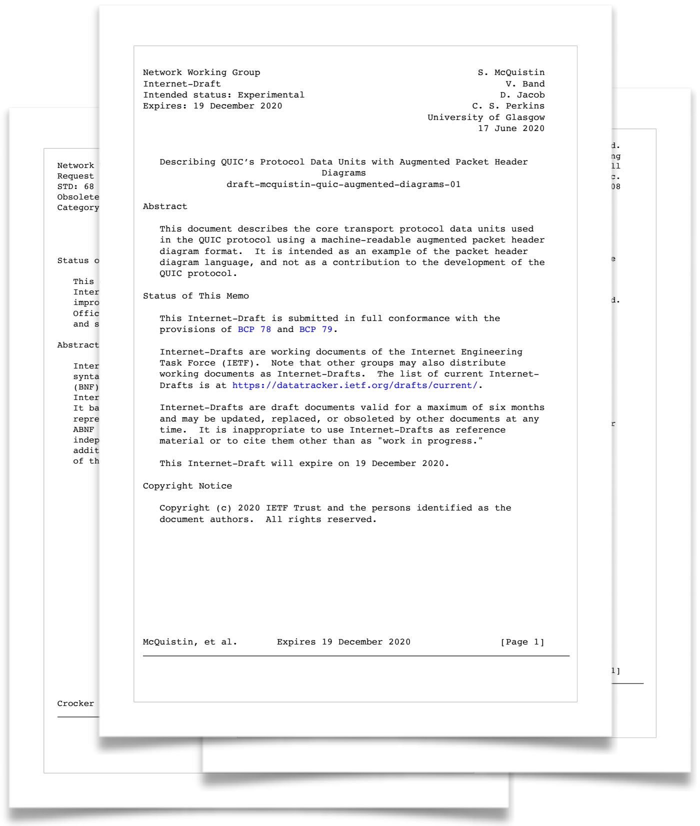
Acknowledgment Number: 32 bits

If the ACK control bit is set, this field c[…] next sequence number the sender of the segme[…] receive. Once a connection is established, this is always sent.

Data Offset: 4 bits

Implementation, debugging, and deployment take time too

Parseable Protocol Standards



Network Working Group S. McQuistin
Internet-Draft V. Band
Intended status: Experimental D. Jacob
Expires: 19 December 2020 C. S. Perkins
University of Glasgow
17 June 2020

Describing QUIC's Protocol Data Units with Augmented Packet Header Diagrams
draft-mcquistin-quic-augmented-diagrams-01

Abstract

This document describes the core transport protocol data units used in the QUIC protocol using a machine-readable augmented packet header diagram format. It is intended as an example of the packet header diagram language, and not as a contribution to the development of the QUIC protocol.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 19 December 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

McQuistin, et al. Expires 19 December 2020 [Page 1]

NLP

Structured language parsers

ABNF

YANG

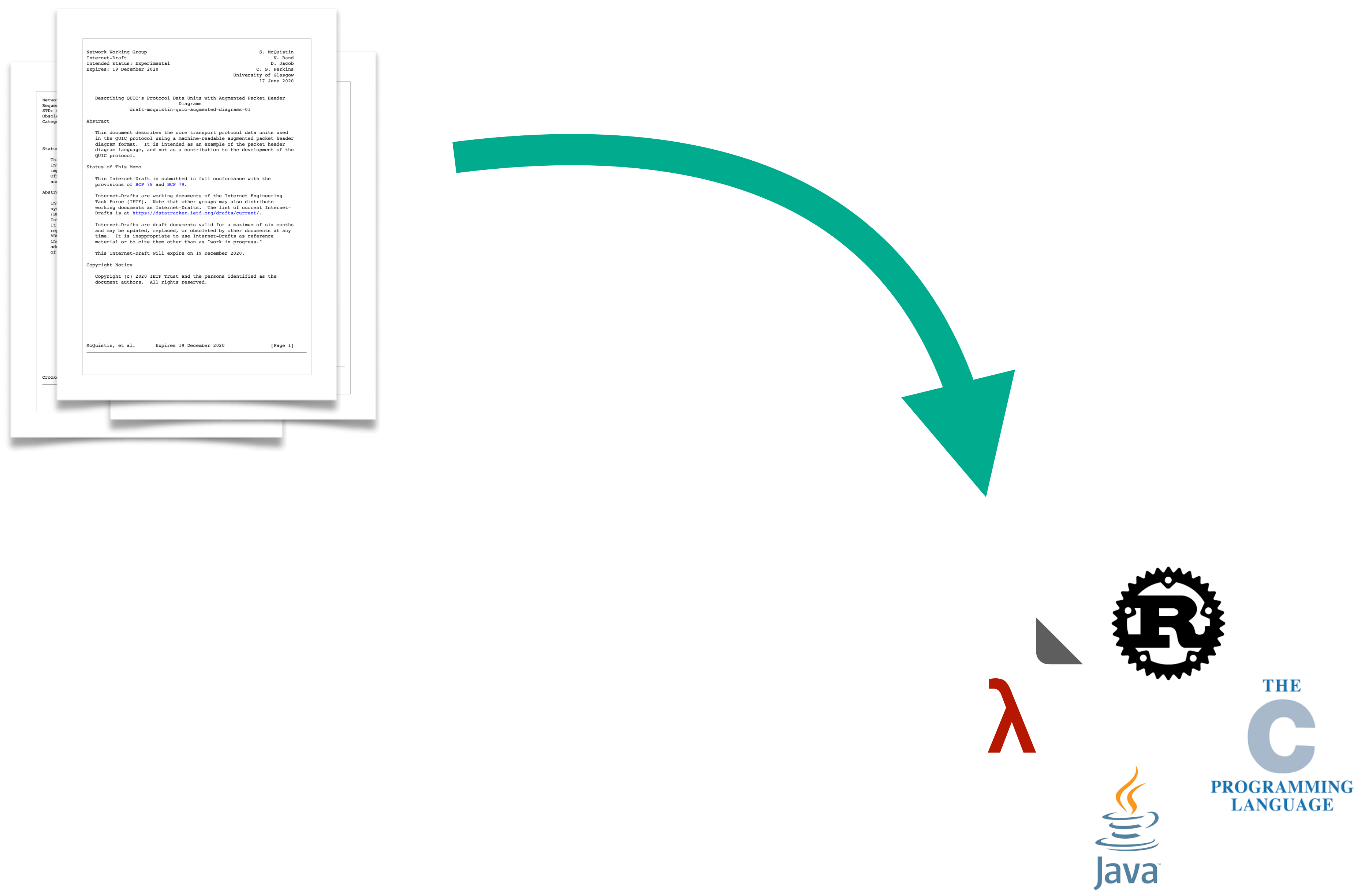
TLS presentation language

Augmented Packet Header Diagrams

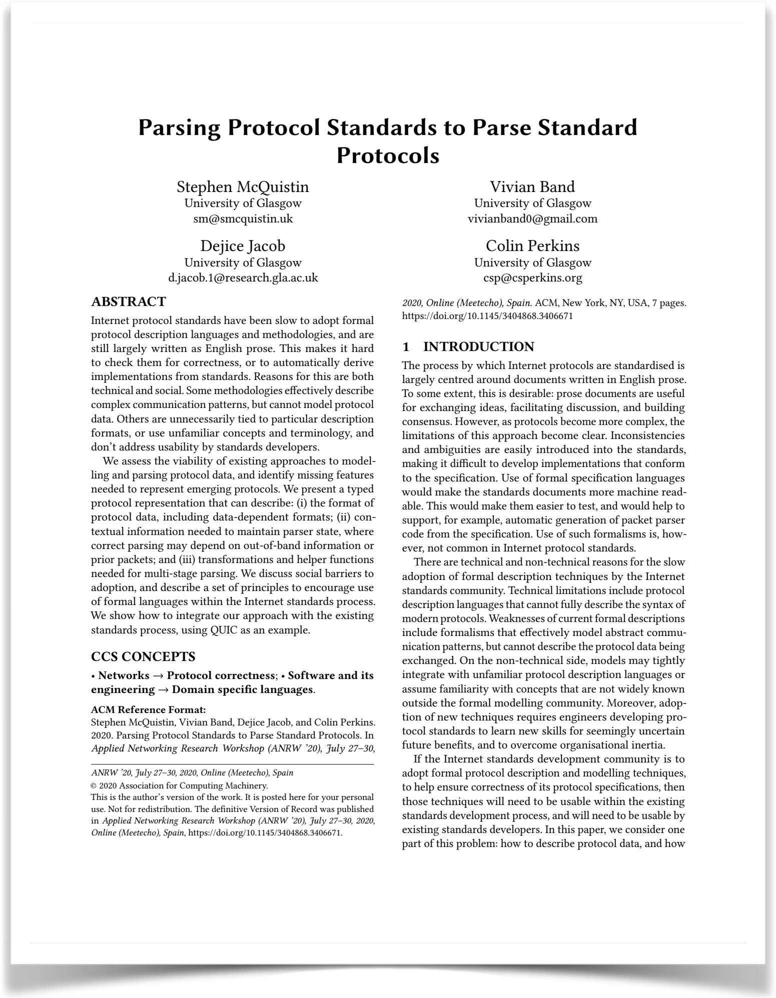
QUIC packet notation

Parseable Protocol Standards

- Enables automatic code generation, allowing for rapid testing and deployment
- Provides rich metadata about the protocol: parameters and fields that can be tweaked, for example
- This in turn enables evolution, adaptation, and experimentation in the network



Publications



Parsing Protocol Standards to Parse Standard Protocols

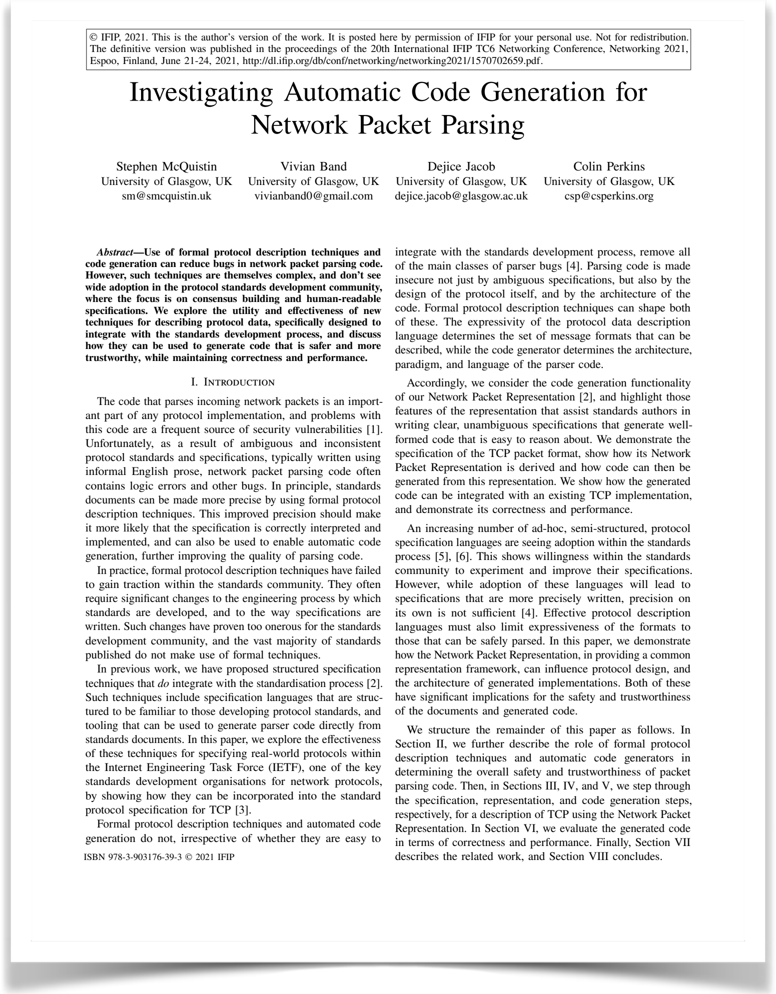
Stephen McQuistin, University of Glasgow, sm@smcquistin.uk
Vivian Band, University of Glasgow, vivianband0@gmail.com
Dejice Jacob, University of Glasgow, d.jacob.1@research.gla.ac.uk
Colin Perkins, University of Glasgow, csp@csperkins.org

Parsing Protocol Standards to Parse Standard Protocols

Stephen McQuistin, Vivian Band, Dejice Jacob, and Colin Perkins

ACM/IRTF Applied Networking Research Workshop, July 2020.

https://doi.org/10.1145/3404868.3406671



Investigating Automatic Code Generation for Network Packet Parsing

Stephen McQuistin, University of Glasgow, UK
Vivian Band, University of Glasgow, UK
Dejice Jacob, University of Glasgow, UK
Colin Perkins, University of Glasgow, UK

Investigating Automatic Code Generation for Network Packet Parsing

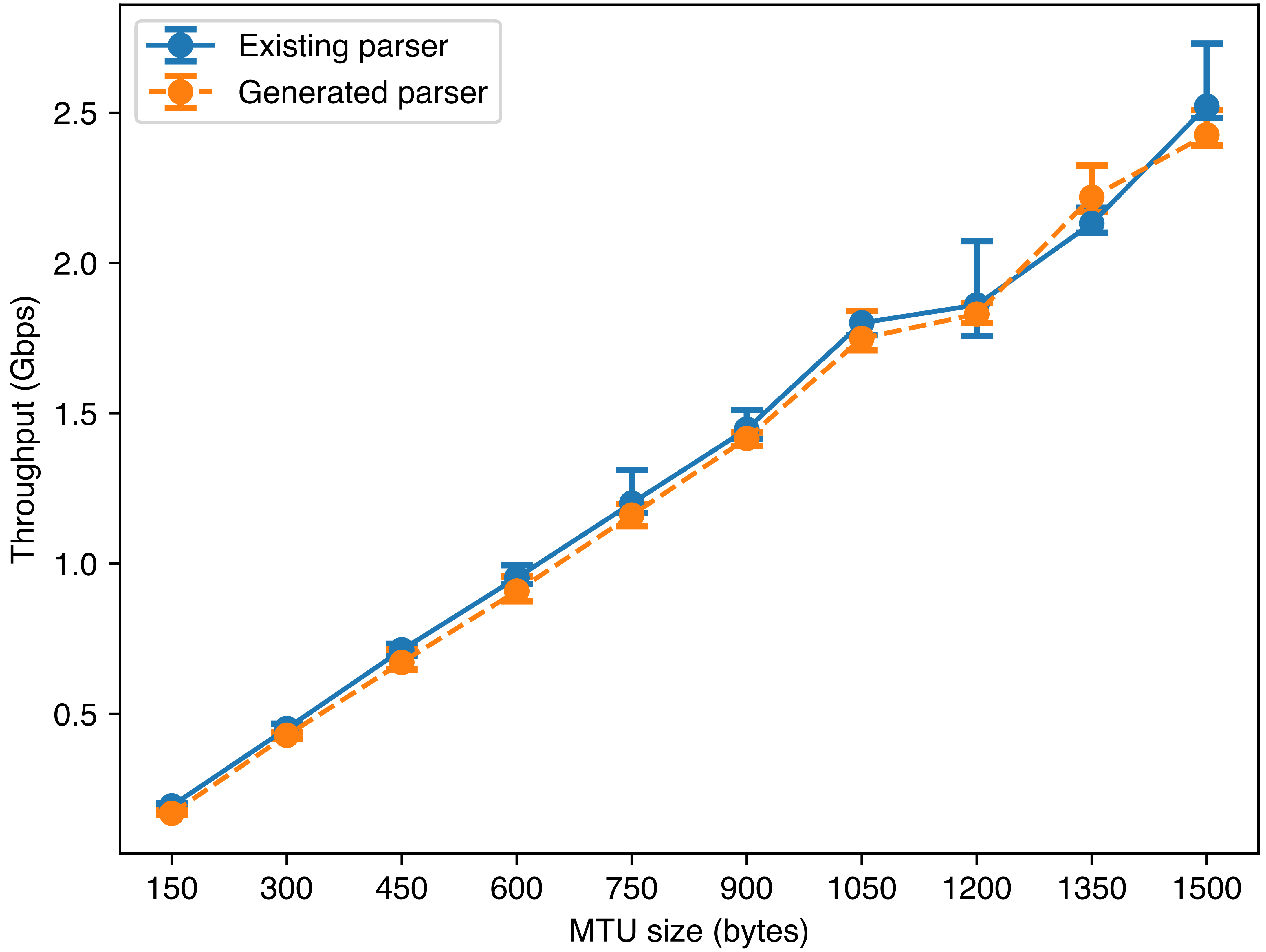
Stephen McQuistin, Vivian Band, Dejice Jacob, and Colin Perkins

IFIP Networking Conference, June 2021.

http://dl.ifip.org/db/conf/networking/networking2021/1570702659.pdf

Augmented Packet Header Diagrams

- Regularised the format of packet header diagrams with minimal change, easing adoption
- Prototype parser code that takes an RFC and generates Rust code for the protocol that is specified
- Generated code is correct and performant
- Adopted in the recent update to the TCP RFC



Summary

- Autonomous networks require rapid deployment and reconfigurability

- This can seem at odds with the protocol standardisation process, which often takes years, and produces documents that require manual implementation and deployment

- Machine parseable standards documents would enable evolution, adaptation, and experimentation