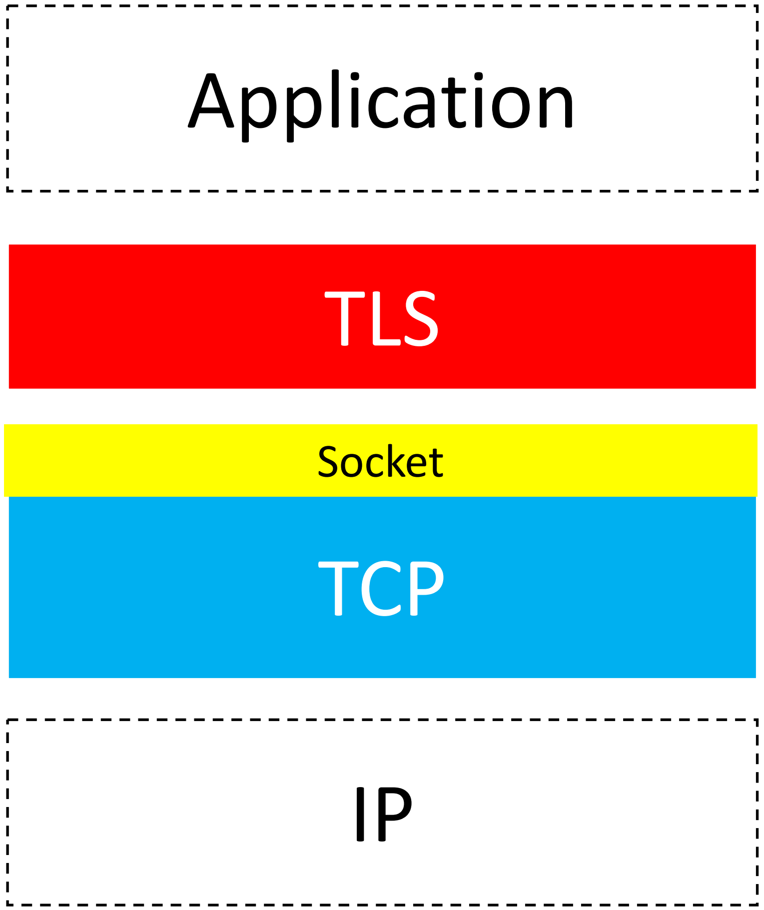
# TCPLS : Closely Integrating TCP and TLS

Florentin Rochet, Emery Assogba, **Olivier Bonaventure**

**UCLouvain**

This work was partially supported by the Walloon government within the MQUIC project

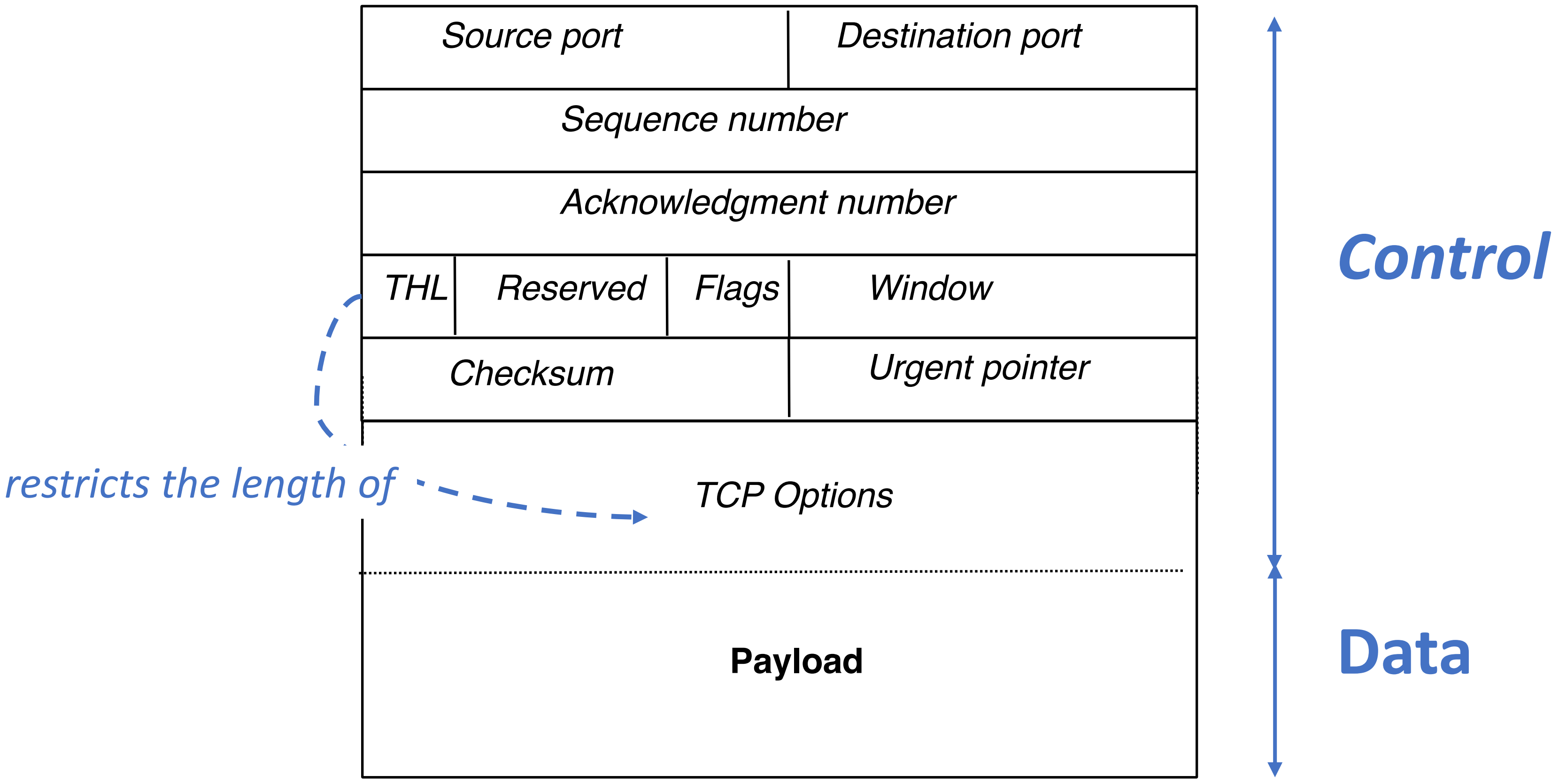
# Our current stack




- TLS 1.3
  - provides security
  - More and more used on WANs and by a variety of applications
- TCP
  - provides connection abstraction, realibility, congestion control
  - Most popular transport protocol
- In the future, TCP could **always** be used with TLS

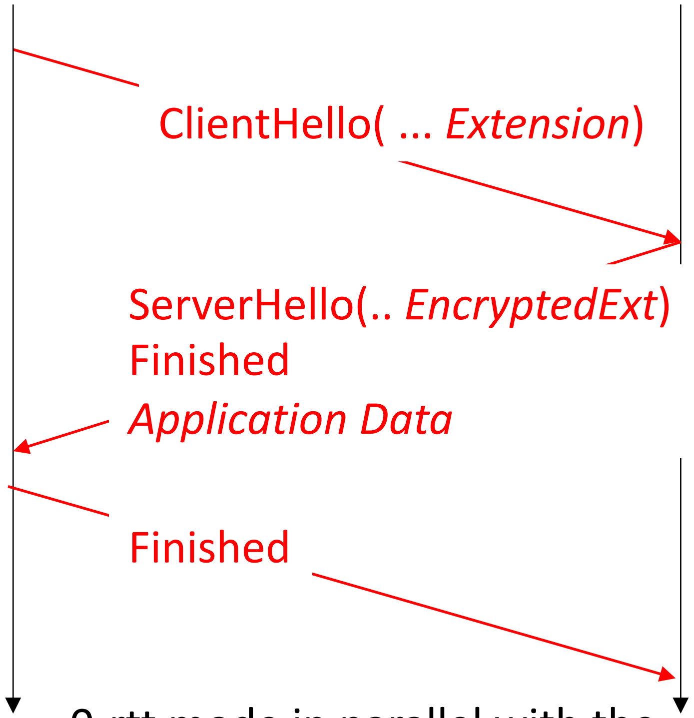
# *Control* and data separation in TCP

- Very simple



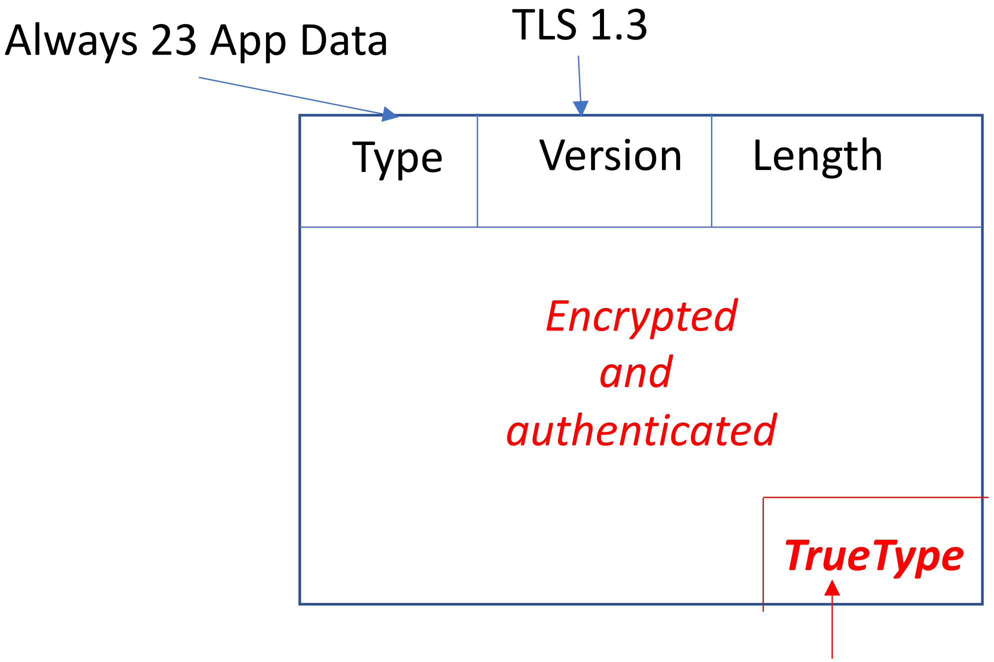
# TLS 1.3 in one slide

- Secure Handshake



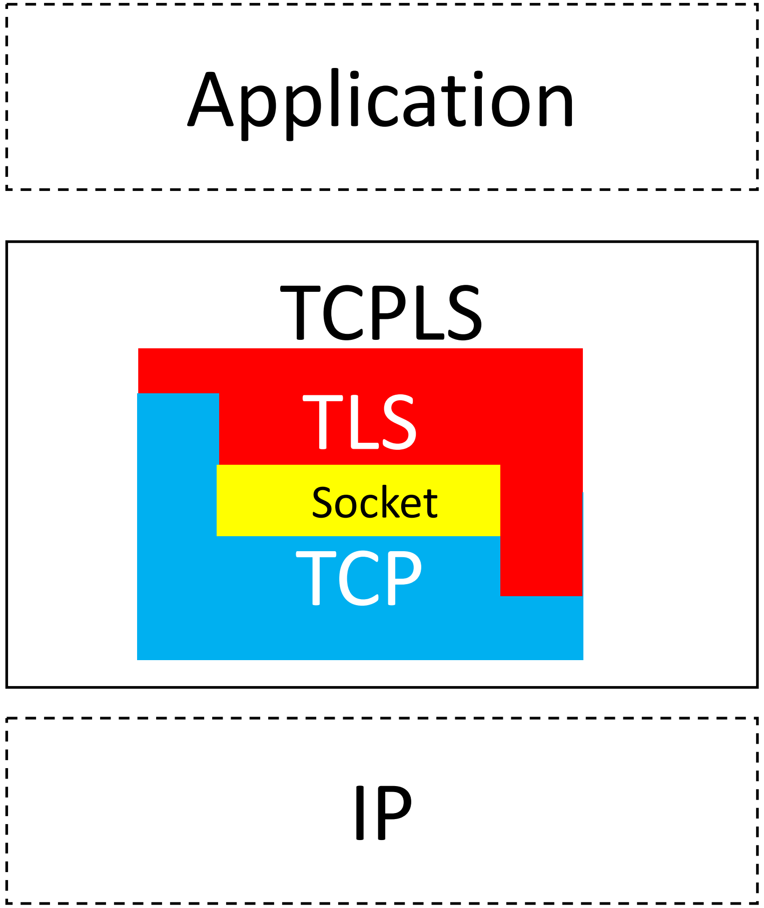

0-rtt mode in parallel with the TCP handshake is possible

- The encrypted TLS records



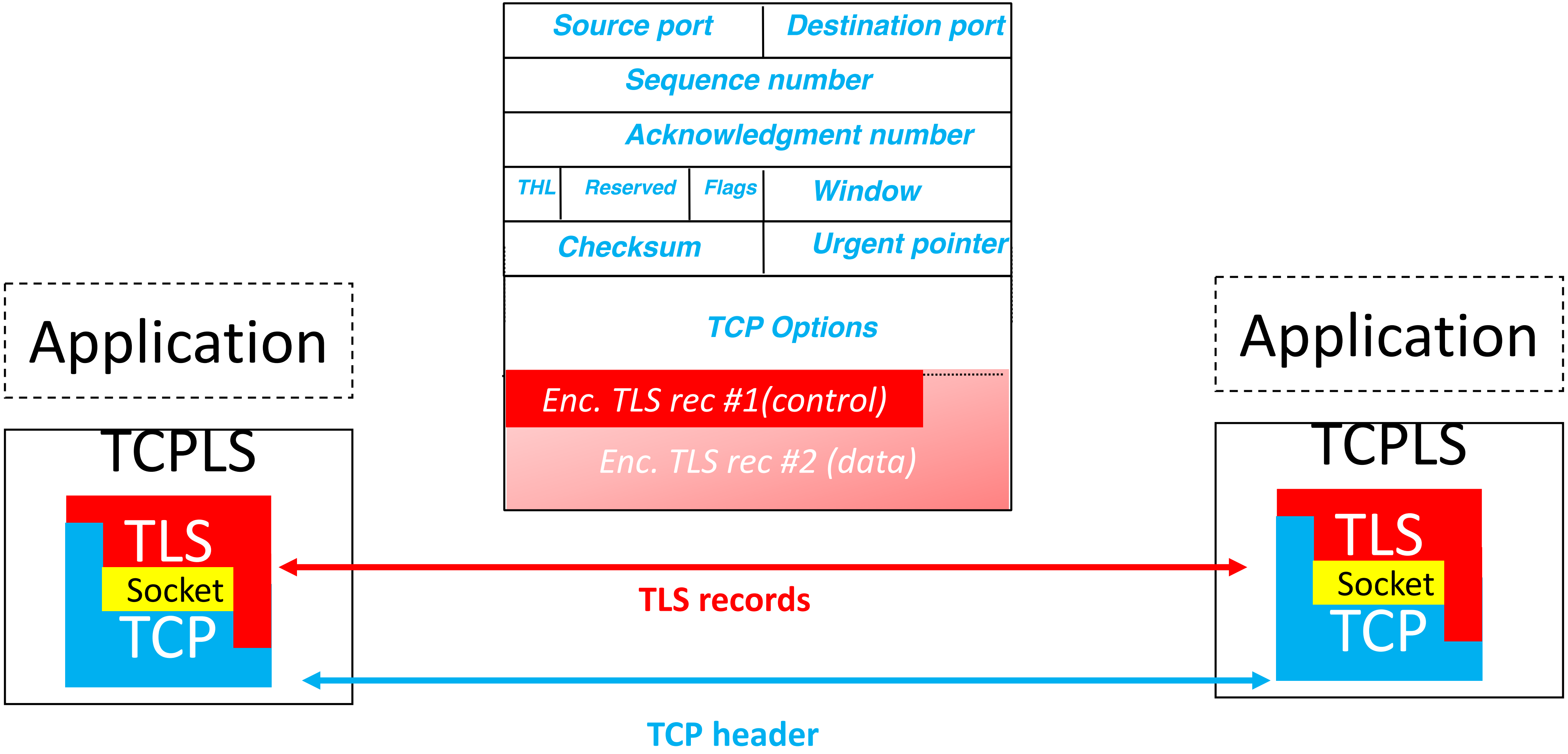

Encrypted and thus invisible to middleboxes

# An integrated stack




- Key idea
- Use new TLS record types to carry TCP control plane information
  - TLS record to carry TCP option
  - TCP option inside ClientHello Extension
  - TCP option inside ServerHello EncrytedExt
- TCPLS has 2 different channels for TCP control
  - regular TCP options
  - Encrypted TLS records

# The TCPLS control channels



# Use case : Securing Multipath TCP

- Security concerns
  - token is exchanged inside SYN/SYN+ACK
  - ADD_ADDR authentication
  - ADD_ADDR not reliable
- With TCPLS
  - Derive token from TLS secrets
  - TCPLS record for ADD_ADDR
    - reliable and authenticated
  - REMOVE_ADDR could still be sent as TCP option

```
Internet Engineering Task Force (IETF)                        M. Bagnulo
Request for Comments: 6181                                          UC3M
Category: Informational                                       March 2011
ISSN: 2070-1721

        Threat Analysis for TCP Extensions for Multipath Operation
                         with Multiple Addresses
```

```
Internet Engineering Task Force (IETF)                        M. Bagnulo
Request for Comments: 7430                                          UC3M
Category: Informational                                        C. Paasch
ISSN: 2070-1721                                               UCLouvain
                                                                 F. Gont
                                                  SI6 Networks / UTN-FRH
                                                         O. Bonaventure
                                                              UCLouvain
                                                               C. Raiciu
                                                                     UPB
                                                               July 2015

           Analysis of Residual Threats and Possible Fixes for
                          Multipath TCP (MPTCP)
```

# Use case : Stronger TFO

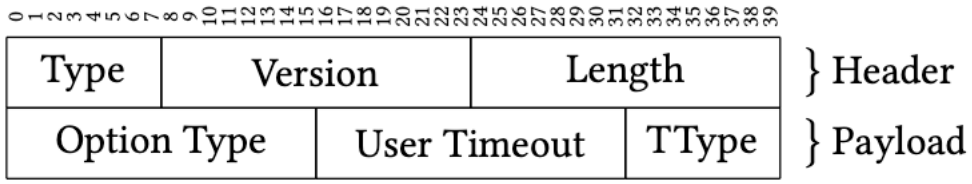
- Concern
  - The security of TFO is limited by the length of the TCP options in the SYN to encode the cookie

- TCPLS approach
  - Use TLS's 0-RTT and
    - send ClientHello inside SYN payload with TCPLS cookie
    - send ServerHello inside SYN+ACK payload with TCPLS cookie

  - Cookies can be longer and more secure by leveraging the existing TLS mechanisms
  - Middlebox interference
    - Apple's measurements do not seem to indicate that the length of the payload in the SYN is a strong factor in middlebox interference

# Use case : More space for TCP options

- TCPLS approach
  - More options during the handshake
    - Leverage the 0-RTT handshake
      - ServerHello inside SYN+ACK and TCP Options as ServerHello EncryptedExt
    - Define TLS record type to carry TCP options




  - Late negotiation of TCP extensions
    - Since TLS records are reliably exchanged, we could also negotiate a TCP extension after the establishment of a connection

# Use case : True keepalives

- Concern
  - Keepalives really part of TCP

**RFC1122**

```
4.2.3.6  TCP Keep-Alives

   Implementors MAY include "keep-alives" in their TCP
   implementations, although this practice is not universally
   accepted.  If keep-alives are included, the application MUST
   be able to turn them on or off for each TCP connection, and
   they MUST default to off.

   Keep-alive packets MUST only be sent when no data or
   acknowledgement packets have been received for the
   connection within an interval.  This interval MUST be
   configurable and MUST default to no less than two hours.
```

- TCPLS approach
  - New ping/pong TCPLS record type
    - Hosts can send ping/pong records including data without interfering with payload
    - TCPLS can negotiate keepalive intervals and other informations

# Use case : Secure session release

- Concern
  - Middleboxes or attackers can force the termination of TCP connections using RST or FIN
- TCPLS approach
  - New authenticated record type indicating end of TCPLS connection
  - If RST or FIN are received before the exchange of this record, then the underlying TCP connection can be automatically reestablished

# Use case : Happy eyeballs

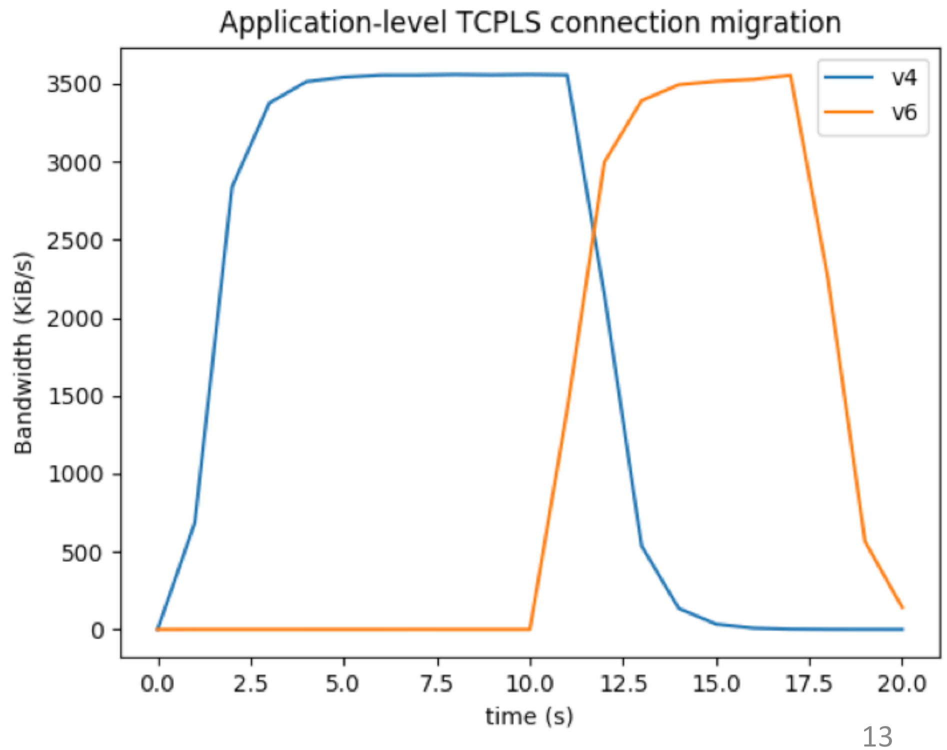
- Server supports IPv4 and IPv6
  - Client learns addresses from DNS and initiates IPv6 and later IPv4 connection

- With TCPLS
  - Server uses EncryptedExt in ServerHello to advertise its alternate address
    - Similar to what QUIC connection migration or MPTCP's ADD_ADDR
  - Client learns alternate server address during handshake
  - Client can create connection to alternate address, test it and migrate the connection

# Use case : Connection migration

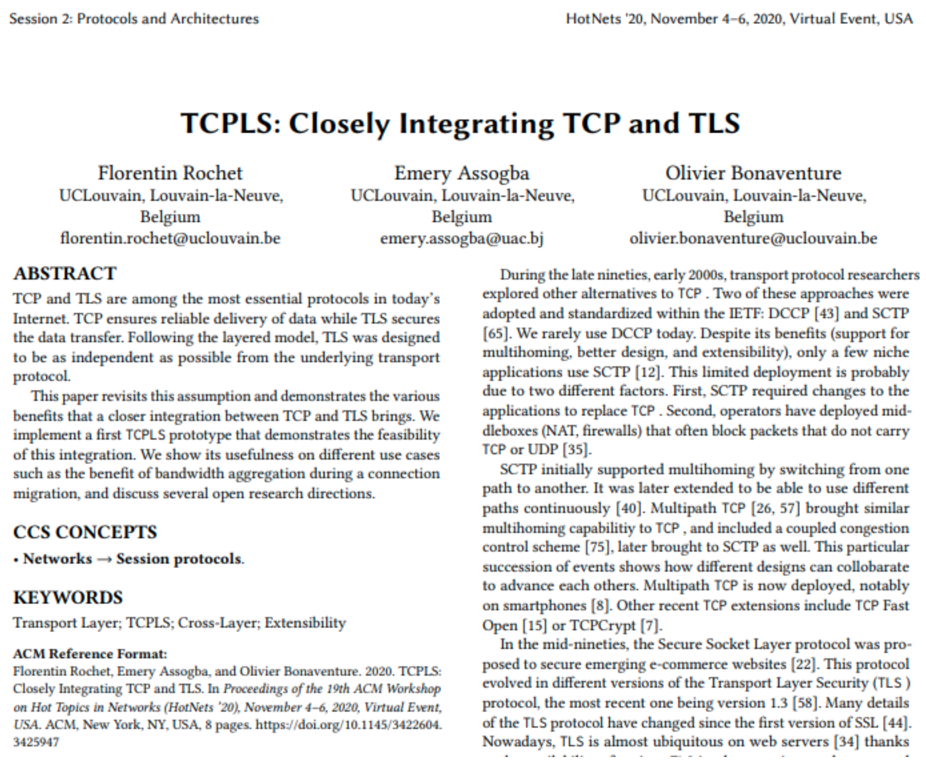
- Concern
  - Smartphone wants to move to cellular while preserving established TCPLS session

- Implemented TCPLS approach
  - Server provides *connection identifier* and *cookie* in ServerHello
  - Client creates second TCPLS *subflow* to server using this information
  - Server and client move data transfer to new TCPLS *subflow*



# Conclusion

- Don't consider TCP and TLS as separate and independent protocols
- TLS 1.3 can be efficiently combined with TCP to improve it
- More details are available in our Hotnets'20 paper



Session 2: Protocols and Architectures HotNets '20, November 4–6, 2020, Virtual Event, USA

## TCPLS: Closely Integrating TCP and TLS

Florentin Rochet, UCLouvain, Louvain-la-Neuve, Belgium, florentin.rochet@uclouvain.be
Emery Assogba, UCLouvain, Louvain-la-Neuve, Belgium, emery.assogba@uac.bj
Olivier Bonaventure, UCLouvain, Louvain-la-Neuve, Belgium, olivier.bonaventure@uclouvain.be

**ABSTRACT**

TCP and TLS are among the most essential protocols in today's Internet. TCP ensures reliable delivery of data while TLS secures the data transfer. Following the layered model, TLS was designed to be as independent as possible from the underlying transport protocol.

This paper revisits this assumption and demonstrates the various benefits that a closer integration between TCP and TLS brings. We implement a first TCPLS prototype that demonstrates the feasibility of this integration. We show its usefulness on different use cases such as the benefit of bandwidth aggregation during a connection migration, and discuss several open research directions.

**CCS CONCEPTS**

• **Networks → Session protocols.**

**KEYWORDS**

Transport Layer; TCPLS; Cross-Layer; Extensibility

**ACM Reference Format:**
Florentin Rochet, Emery Assogba, and Olivier Bonaventure. 2020. TCPLS: Closely Integrating TCP and TLS. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks (HotNets '20), November 4–6, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3422604.3425947

During the late nineties, early 2000s, transport protocol researchers explored other alternatives to TCP . Two of these approaches were adopted and standardized within the IETF: DCCP [43] and SCTP [65]. We rarely use DCCP today. Despite its benefits (support for multihoming, better design, and extensibility), only a few niche applications use SCTP [12]. This limited deployment is probably due to two different factors. First, SCTP required changes to the applications to replace TCP . Second, operators have deployed middleboxes (NAT, firewalls) that often block packets that do not carry TCP or UDP [35].

SCTP initially supported multihoming by switching from one path to another. It was later extended to be able to use different paths continuously [40]. Multipath TCP [26, 57] brought similar multihoming capabilitiy to TCP , and included a coupled congestion control scheme [75], later brought to SCTP as well. This particular succession of events shows how different designs can collobarate to advance each others. Multipath TCP is now deployed, notably on smartphones [8]. Other recent TCP extensions include TCP Fast Open [15] or TCPCrypt [7].

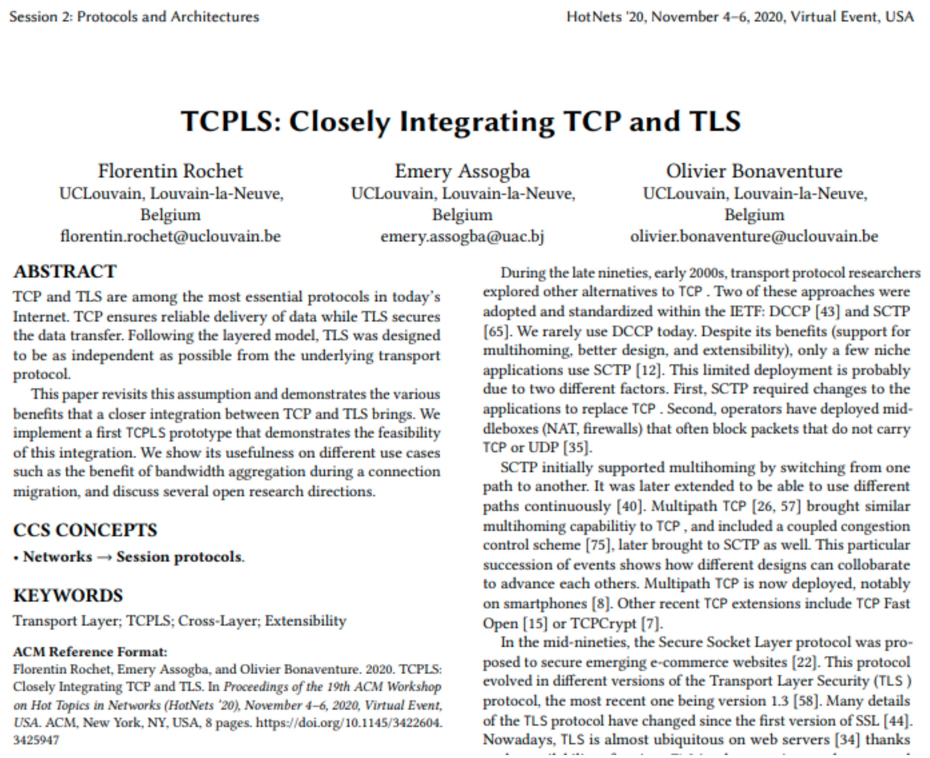
In the mid-nineties, the Secure Socket Layer protocol was proposed to secure emerging e-commerce websites [22]. This protocol evolved in different versions of the Transport Layer Security (TLS ) protocol, the most recent one being version 1.3 [58]. Many details of the TLS protocol have changed since the first version of SSL [44]. Nowadays, TLS is almost ubiquitous on web servers [34] thanks

- There is running code based on picotls at

https://pluginized-protocols.org

# Conclusion

- Don’t consider TCP and TLS as separate and independent protocols
- TLS 1.3 can be efficiently combined with TCP to improve it
- More details are available in our Hotnets’20 paper





## TCPLS: Closely Integrating TCP and TLS

Florentin Rochet
UCLouvain, Louvain-la-Neuve, Belgium
florentin.rochet@uclouvain.be

Emery Assogba
UCLouvain, Louvain-la-Neuve, Belgium
emery.assogba@uac.bj

Olivier Bonaventure
UCLouvain, Louvain-la-Neuve, Belgium
olivier.bonaventure@uclouvain.be

**ABSTRACT**

TCP and TLS are among the most essential protocols in today’s Internet. TCP ensures reliable delivery of data while TLS secures the data transfer. Following the layered model, TLS was designed to be as independent as possible from the underlying transport protocol.

This paper revisits this assumption and demonstrates the various benefits that a closer integration between TCP and TLS brings. We implement a first TCPLS prototype that demonstrates the feasibility of this integration. We show its usefulness on different use cases such as the benefit of bandwidth aggregation during a connection migration, and discuss several open research directions.

**CCS CONCEPTS**

• **Networks → Session protocols**.

**KEYWORDS**

Transport Layer; TCPLS; Cross-Layer; Extensibility

**ACM Reference Format:**
Florentin Rochet, Emery Assogba, and Olivier Bonaventure. 2020. TCPLS: Closely Integrating TCP and TLS. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks (HotNets '20), November 4–6, 2020, Virtual Event, USA*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3422604.3425947

During the late nineties, early 2000s, transport protocol researchers explored other alternatives to TCP . Two of these approaches were adopted and standardized within the IETF: DCCP [43] and SCTP [65]. We rarely use DCCP today. Despite its benefits (support for multihoming, better design, and extensibility), only a few niche applications use SCTP [12]. This limited deployment is probably due to two different factors. First, SCTP required changes to the applications to replace TCP . Second, operators have deployed middleboxes (NAT, firewalls) that often block packets that do not carry TCP or UDP [35].

SCTP initially supported multihoming by switching from one path to another. It was later extended to be able to use different paths continuously [40]. Multipath TCP [26, 57] brought similar multihoming capabilitiy to TCP , and included a coupled congestion control scheme [75], later brought to SCTP as well. This particular succession of events shows how different designs can collobarate to advance each others. Multipath TCP is now deployed, notably on smartphones [8]. Other recent TCP extensions include TCP Fast Open [15] or TCPCrypt [7].

In the mid-nineties, the Secure Socket Layer protocol was proposed to secure emerging e-commerce websites [22]. This protocol evolved in different versions of the Transport Layer Security (TLS ) protocol, the most recent one being version 1.3 [58]. Many details of the TLS protocol have changed since the first version of SSL [44]. Nowadays, TLS is almost ubiquitous on web servers [34] thanks

- There is running code based on picotls at

https://pluginized-protocols.org