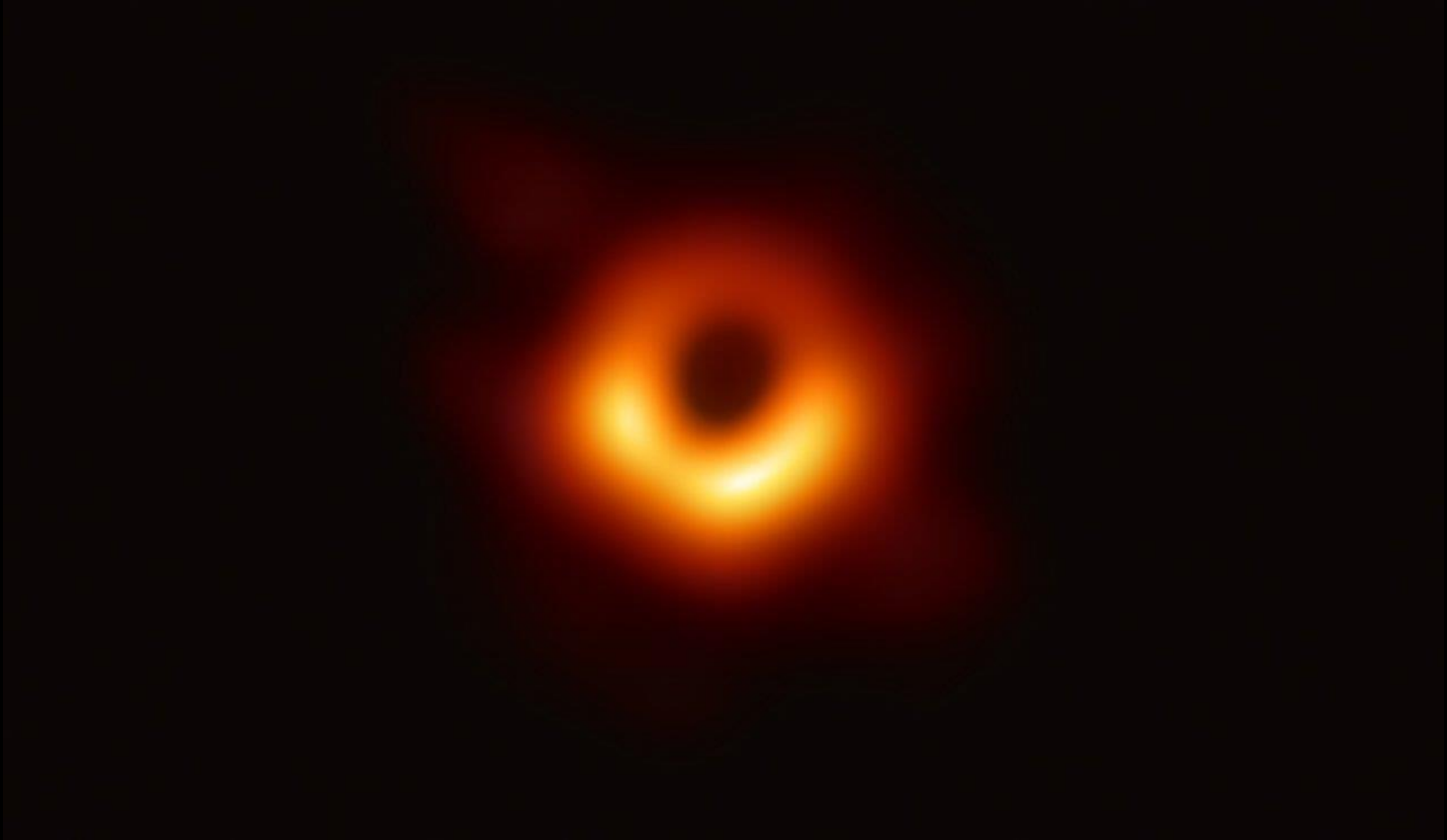
Real Logic
ACCELERATING SOFTWARE

Interaction Protocols

Martin Thompson - @mjpt777

Code

PHILIPS
Azur Performer Plus



How significant are protocols for software development?

protocol

noun \ ˈprō-tə-ˌkȯl \

Source: http://www.merriam-webster.com/

protocol

noun \ ˈprō-tə-ˌkȯl \

: a code prescribing strict adherence to correct etiquette and precedence

Source: http://www.merriam-webster.com/

protocol

noun \ ˈprō-tə-ˌkȯl \

: a code prescribing strict adherence to correct etiquette and precedence

: a set of conventions governing the treatment and formatting of data in an electronic communications system

Source: http://www.merriam-webster.com/

Evolutionary Biology & Communication

Facial Expressions

Manners & Etiquette

Evolution of Communities

Hygiene <=> Disease

Evolution of Communities

Hygiene <=> Disease

Courtesy <=> Society

Evolution of Communities

Hygiene	<=>	Disease
Courtesy	<=>	Society
Norms	<=>	Trust

Formal Protocols

Rules of Engagement

Rules of Engagement

“What is acceptable”

Rules of Engagement

“What is acceptable”
“Good conditions to succeed”

Rules of Engagement

“What is acceptable”

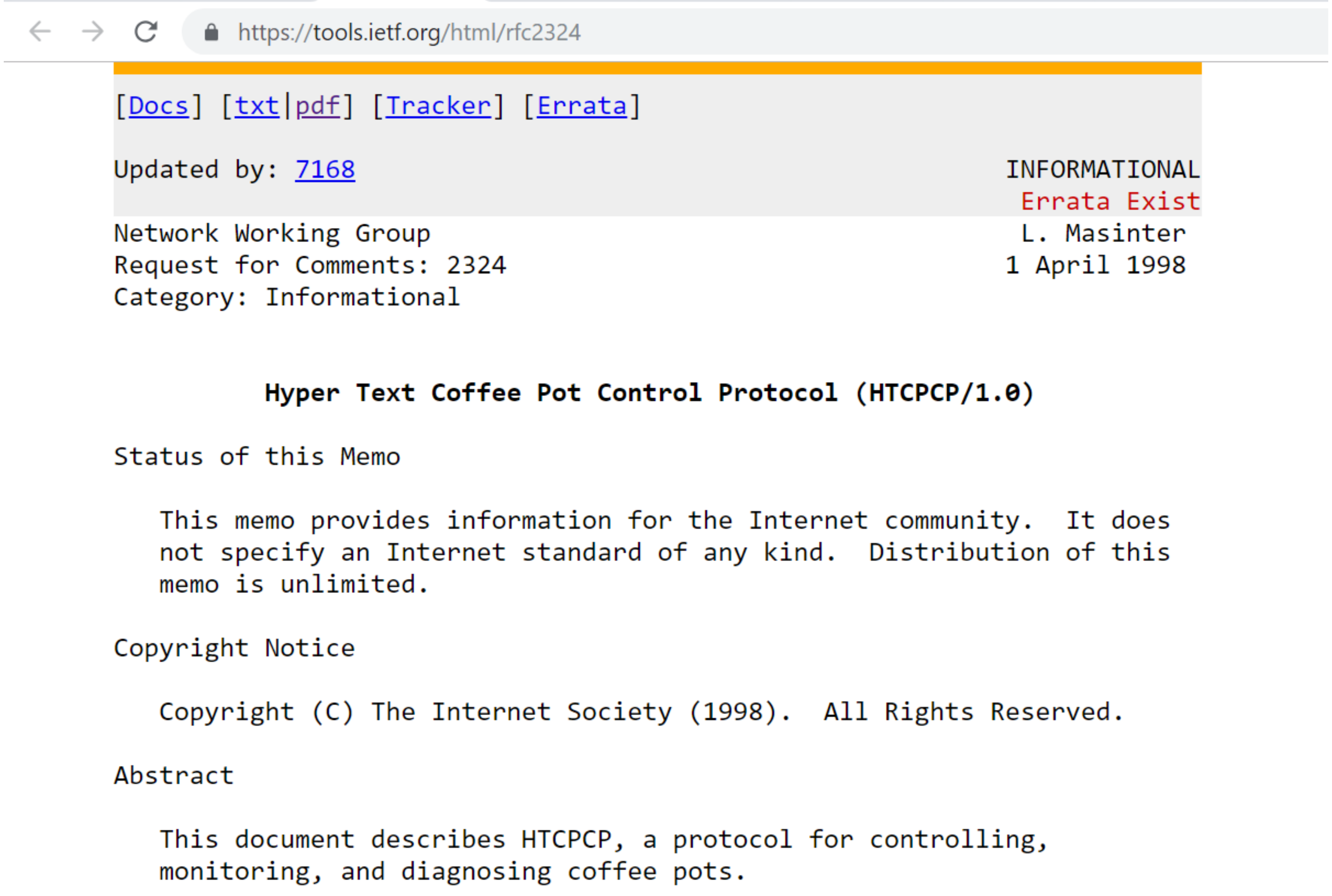
“Good conditions to succeed”

“*Jus ad bellum, Jus ad bello*”

Concurrent & Distributed Systems

How should components Interact?

IETF
(Internet Engineering Task Force)



https://tools.ietf.org/html/rfc2324

[Docs] [txt|pdf] [Tracker] [Errata]

```
Updated by: 7168                                           INFORMATIONAL
                                                           Errata Exist
Network Working Group                                       L. Masinter
Request for Comments: 2324                                 1 April 1998
Category: Informational
```

Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0)

Status of this Memo

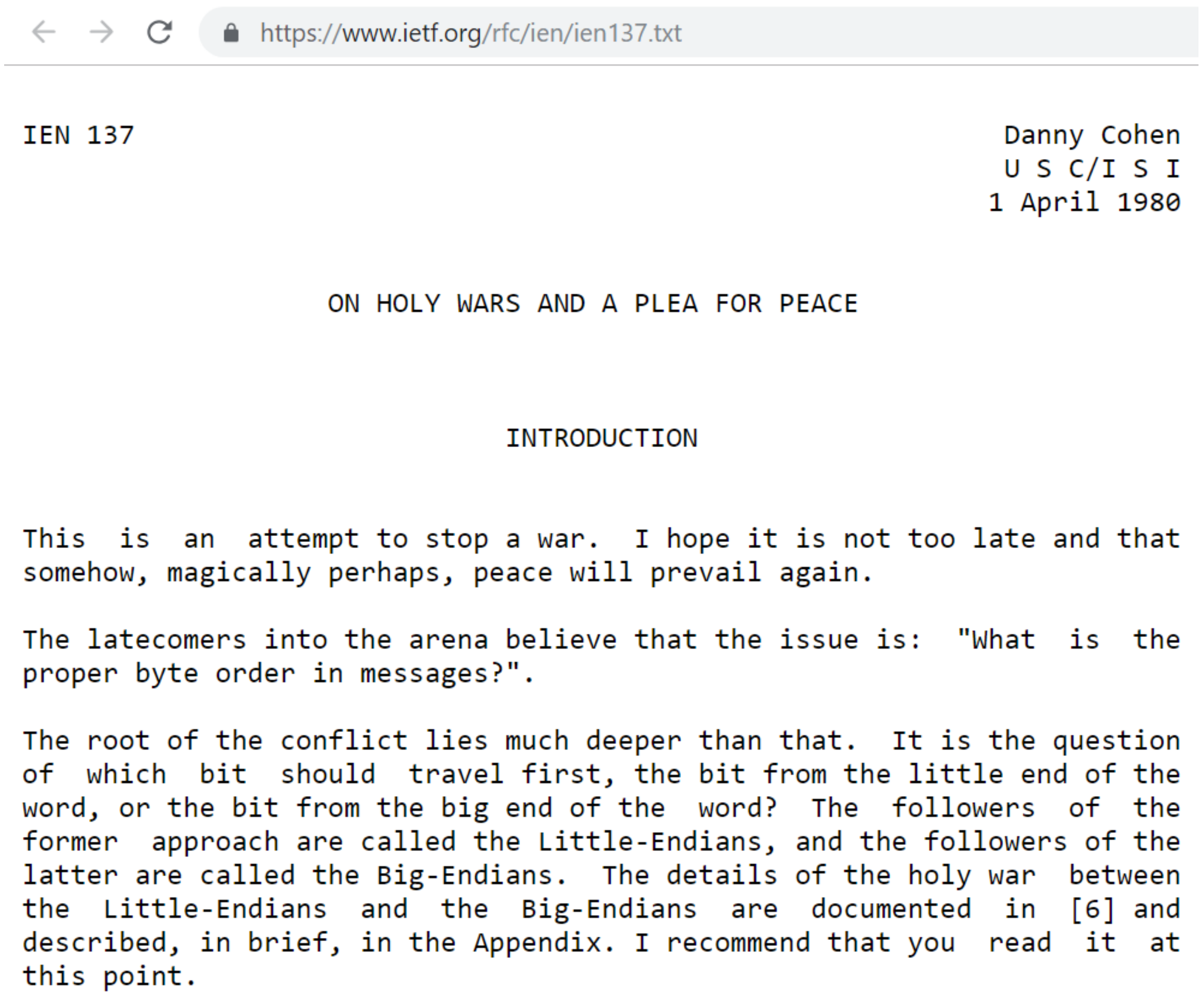
This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document describes HTCPCP, a protocol for controlling, monitoring, and diagnosing coffee pots.

418
“I'm a teapot”



https://www.ietf.org/rfc/ien/ien137.txt

IEN 137

Danny Cohen
U S C/I S I
1 April 1980

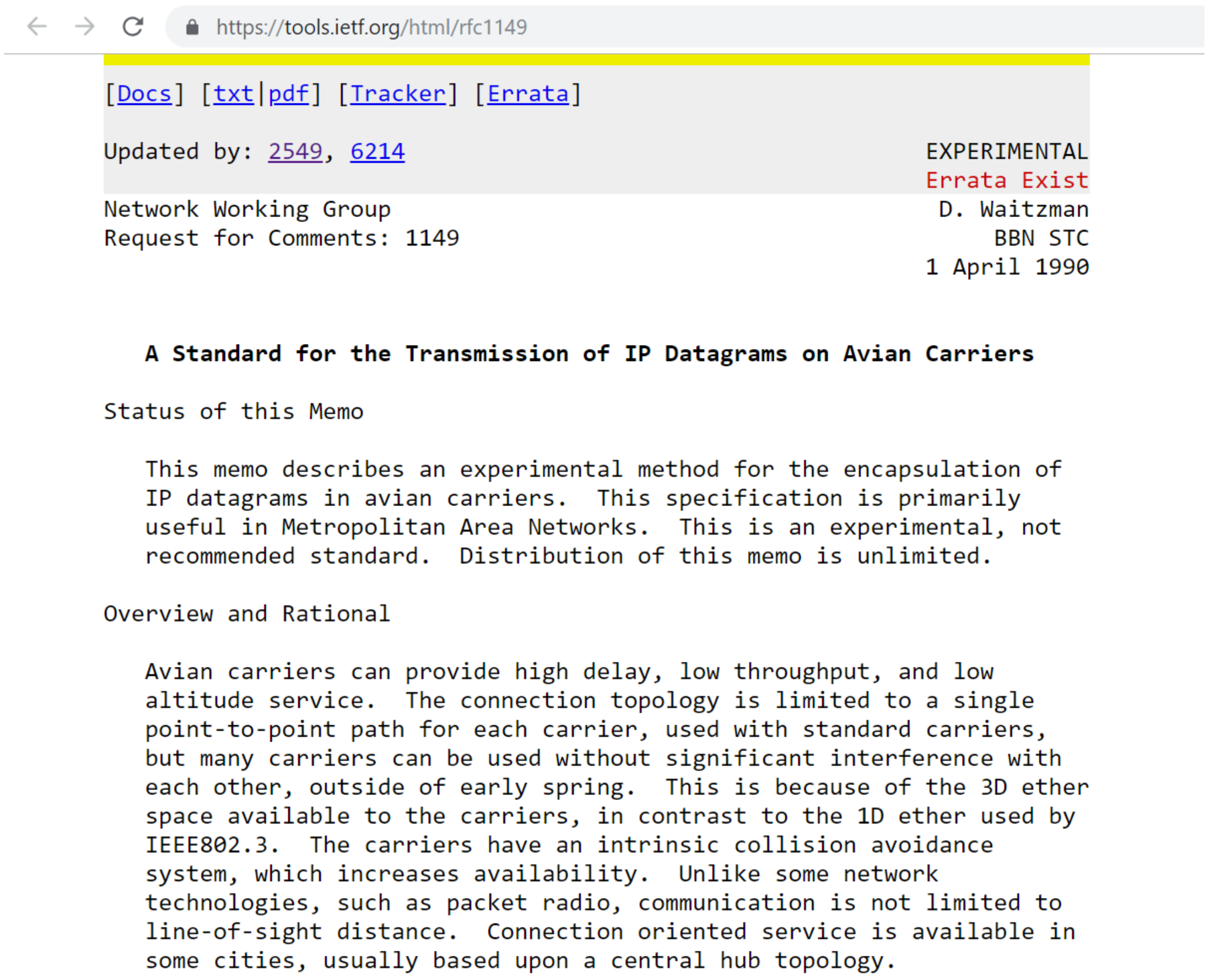
ON HOLY WARS AND A PLEA FOR PEACE

INTRODUCTION

This is an attempt to stop a war. I hope it is not too late and that somehow, magically perhaps, peace will prevail again.

The latecomers into the arena believe that the issue is: "What is the proper byte order in messages?".

The root of the conflict lies much deeper than that. It is the question of which bit should travel first, the bit from the little end of the word, or the bit from the big end of the word? The followers of the former approach are called the Little-Endians, and the followers of the latter are called the Big-Endians. The details of the holy war between the Little-Endians and the Big-Endians are documented in [6] and described, in brief, in the Appendix. I recommend that you read it at this point.



https://tools.ietf.org/html/rfc1149

[Docs] [txt|pdf] [Tracker] [Errata]

```
Updated by: 2549, 6214                                      EXPERIMENTAL
                                                           Errata Exist
Network Working Group                                       D. Waitzman
Request for Comments: 1149                                      BBN STC
                                                           1 April 1990
```

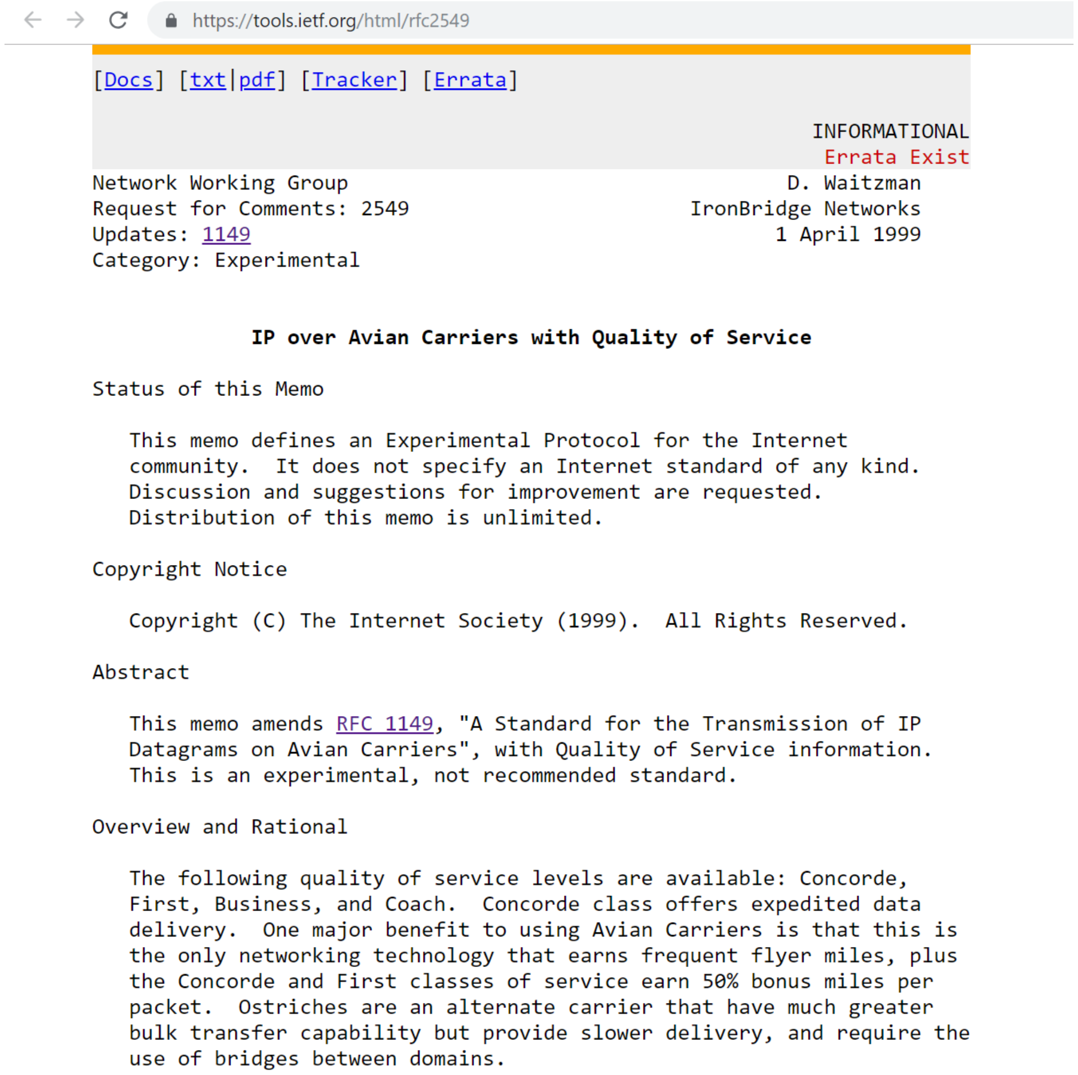
A Standard for the Transmission of IP Datagrams on Avian Carriers

Status of this Memo

This memo describes an experimental method for the encapsulation of IP datagrams in avian carriers. This specification is primarily useful in Metropolitan Area Networks. This is an experimental, not recommended standard. Distribution of this memo is unlimited.

Overview and Rational

Avian carriers can provide high delay, low throughput, and low altitude service. The connection topology is limited to a single point-to-point path for each carrier, used with standard carriers, but many carriers can be used without significant interference with each other, outside of early spring. This is because of the 3D ether space available to the carriers, in contrast to the 1D ether used by IEEE802.3. The carriers have an intrinsic collision avoidance system, which increases availability. Unlike some network technologies, such as packet radio, communication is not limited to line-of-sight distance. Connection oriented service is available in some cities, usually based upon a central hub topology.


https://tools.ietf.org/html/rfc2549

[Docs] [txt|pdf] [Tracker] [Errata]

INFORMATIONAL
Errata Exist

```
Network Working Group                                       D. Waitzman
Request for Comments: 2549                           IronBridge Networks
Updates: 1149                                              1 April 1999
Category: Experimental
```

IP over Avian Carriers with Quality of Service

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

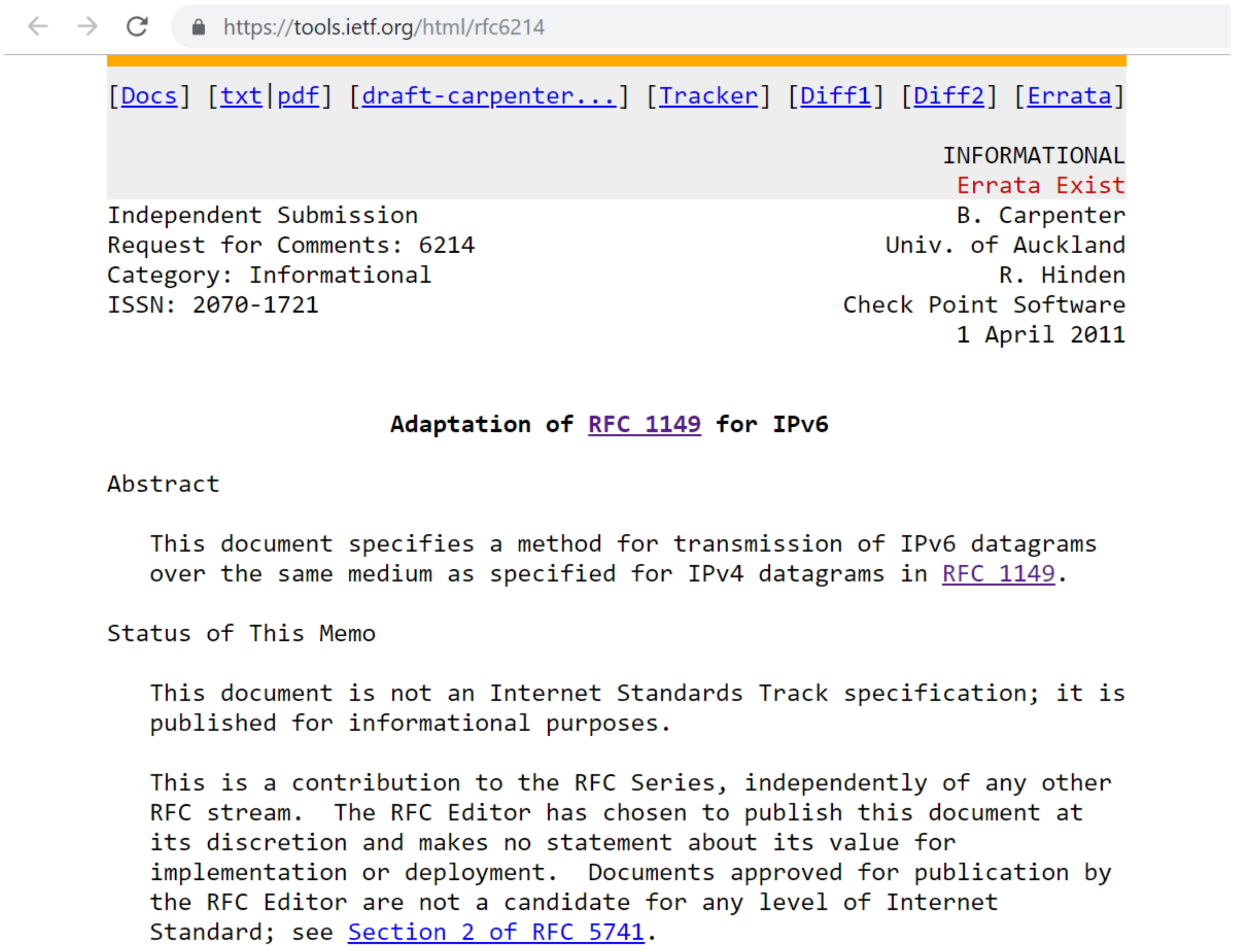
Copyright Notice

Abstract

This memo amends RFC 1149, "A Standard for the Transmission of IP Datagrams on Avian Carriers", with Quality of Service information. This is an experimental, not recommended standard.

Overview and Rational

The following quality of service levels are available: Concorde, First, Business, and Coach. Concorde class offers expedited data delivery. One major benefit to using Avian Carriers is that this is the only networking technology that earns frequent flyer miles, plus the Concorde and First classes of service earn 50% bonus miles per packet. Ostriches are an alternate carrier that have much greater bulk transfer capability but provide slower delivery, and require the use of bridges between domains.



https://tools.ietf.org/html/rfc6214

[Docs] [txt|pdf] [draft-carpenter...] [Tracker] [Diff1] [Diff2] [Errata]

```
                                                        INFORMATIONAL
                                                         Errata Exist
Independent Submission                                   B. Carpenter
Request for Comments: 6214                          Univ. of Auckland
Category: Informational                                     R. Hinden
ISSN: 2070-1721                                  Check Point Software
                                                         1 April 2011
```

Adaptation of RFC 1149 for IPv6

Abstract

This document specifies a method for transmission of IPv6 datagrams over the same medium as specified for IPv4 datagrams in RFC 1149.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

How should we document our protocols?

API vs Protocol

```
Open, *[Read | Write], Close
```

```
Open, *[Read | Write], Close
```

1. Open: ...
2. Read: ...
3. Write: ...
4. Close: ...

```
Connect
=======

    Sender: +SETUP                          +SETUP
                  \                        /      \
    Receiver:      +[STATUS | STATUS-SETUP]        +STATUS

Flow Control
============

    Sender:           *DATA
                     /
    Receiver: +STATUS

Loss Recovery
=============

    Sender: +[DATA | DATA-PAD | HEARTBEAT]    +[DATA | DATA-PAD]
                                          \   /
    Receiver:                              NAK

RTT Measurement
===============

    Sender: RTTM-R              RTTM
                  \            /
    Receiver:      RTTM | RTTM-R
```

“What could possibly go wrong?”

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

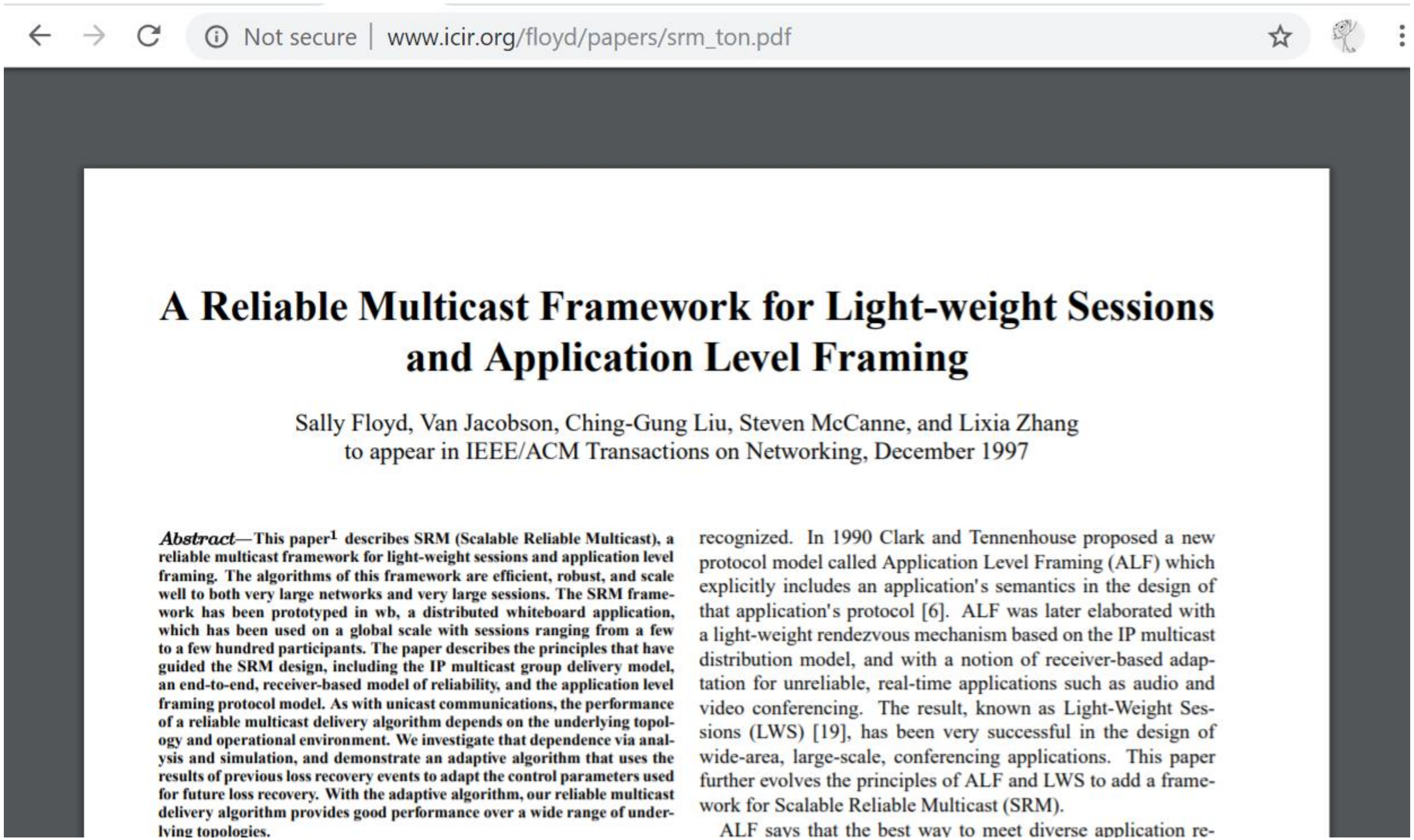
Simple Testing Can Prevent Most Critical Failures: An Analysis of Production Failures in Distributed Data-Intensive Systems

Ding Yuan, Yu Luo, Xin Zhuang, Guilherme Renna Rodrigues, Xu Zhao,
Yongle Zhang, Pranay U. Jain, and Michael Stumm, *University of Toronto*

https://www.usenix.org/conference/osdi14/technical-sessions/presentation/yuan

Multicast Example

ACK / NAK Implosion



Not secure | www.icir.org/floyd/papers/srm_ton.pdf

A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing

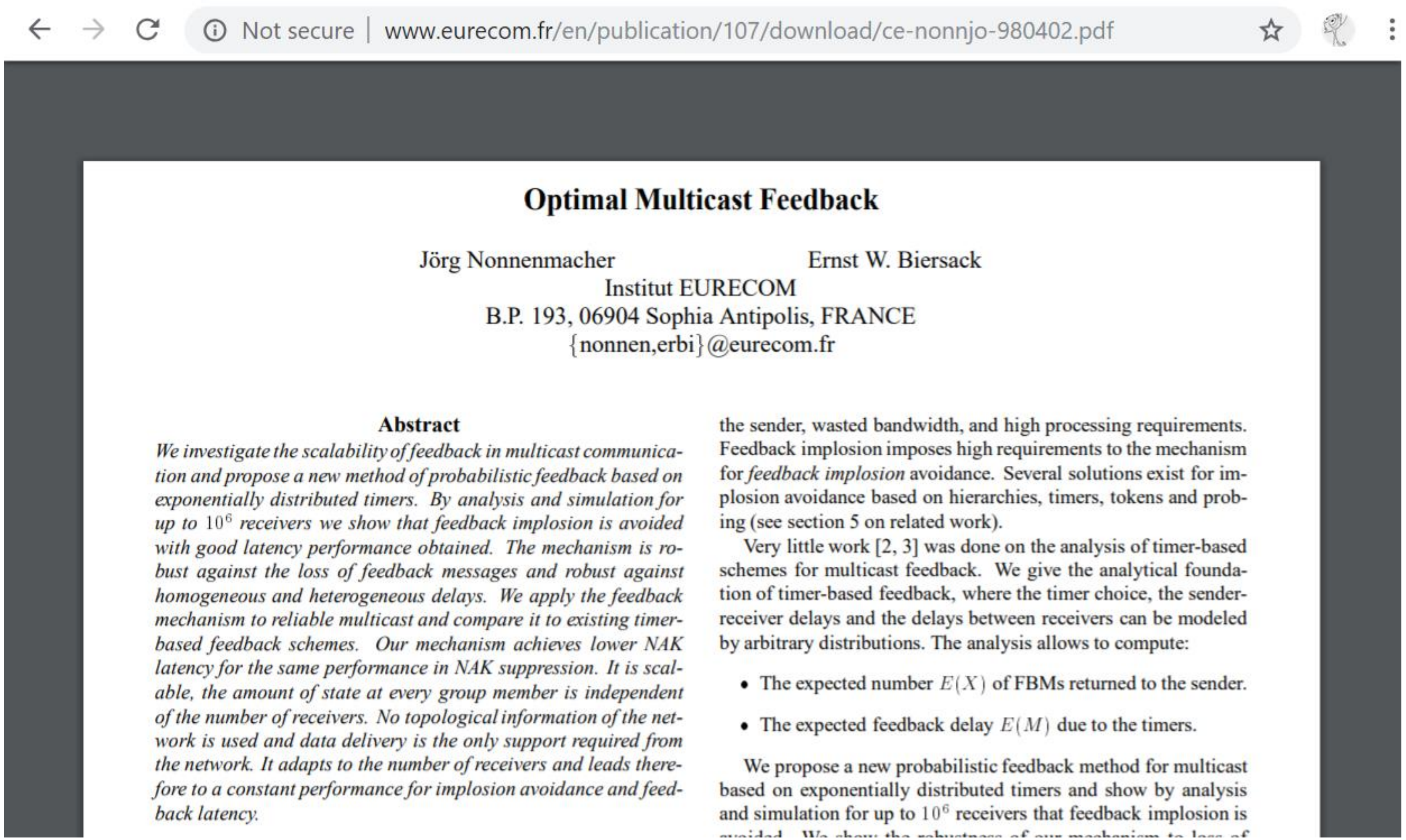
Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang

to appear in IEEE/ACM Transactions on Networking, December 1997

***Abstract*—This paper[1] describes SRM (Scalable Reliable Multicast), a reliable multicast framework for light-weight sessions and application level framing. The algorithms of this framework are efficient, robust, and scale well to both very large networks and very large sessions. The SRM framework has been prototyped in wb, a distributed whiteboard application, which has been used on a global scale with sessions ranging from a few to a few hundred participants. The paper describes the principles that have guided the SRM design, including the IP multicast group delivery model, an end-to-end, receiver-based model of reliability, and the application level framing protocol model. As with unicast communications, the performance of a reliable multicast delivery algorithm depends on the underlying topology and operational environment. We investigate that dependence via analysis and simulation, and demonstrate an adaptive algorithm that uses the results of previous loss recovery events to adapt the control parameters used for future loss recovery. With the adaptive algorithm, our reliable multicast delivery algorithm provides good performance over a wide range of underlying topologies.**

recognized. In 1990 Clark and Tennenhouse proposed a new protocol model called Application Level Framing (ALF) which explicitly includes an application's semantics in the design of that application's protocol [6]. ALF was later elaborated with a light-weight rendezvous mechanism based on the IP multicast distribution model, and with a notion of receiver-based adaptation for unreliable, real-time applications such as audio and video conferencing. The result, known as Light-Weight Sessions (LWS) [19], has been very successful in the design of wide-area, large-scale, conferencing applications. This paper further evolves the principles of ALF and LWS to add a framework for Scalable Reliable Multicast (SRM).

ALF says that the best way to meet diverse application re-


Not secure
www.eurecom.fr/en/publication/107/download/ce-nonnjo-980402.pdf

Optimal Multicast Feedback

Jörg Nonnenmacher Ernst W. Biersack
Institut EURECOM
B.P. 193, 06904 Sophia Antipolis, FRANCE
{nonnen,erbi}@eurecom.fr

Abstract

We investigate the scalability of feedback in multicast communication and propose a new method of probabilistic feedback based on exponentially distributed timers. By analysis and simulation for up to 10^6 receivers we show that feedback implosion is avoided with good latency performance obtained. The mechanism is robust against the loss of feedback messages and robust against homogeneous and heterogeneous delays. We apply the feedback mechanism to reliable multicast and compare it to existing timer-based feedback schemes. Our mechanism achieves lower NAK latency for the same performance in NAK suppression. It is scalable, the amount of state at every group member is independent of the number of receivers. No topological information of the network is used and data delivery is the only support required from the network. It adapts to the number of receivers and leads therefore to a constant performance for implosion avoidance and feedback latency.

the sender, wasted bandwidth, and high processing requirements. Feedback implosion imposes high requirements to the mechanism for *feedback implosion* avoidance. Several solutions exist for implosion avoidance based on hierarchies, timers, tokens and probing (see section 5 on related work).

Very little work [2, 3] was done on the analysis of timer-based schemes for multicast feedback. We give the analytical foundation of timer-based feedback, where the timer choice, the sender-receiver delays and the delays between receivers can be modeled by arbitrary distributions. The analysis allows to compute:

- The expected number $E(X)$ of FBMs returned to the sender.
- The expected feedback delay $E(M)$ due to the timers.

We propose a new probabilistic feedback method for multicast based on exponentially distributed timers and show by analysis and simulation for up to 10^6 receivers that feedback implosion is avoided. We show the robustness of our mechanism to loss of

Where should we focus?

Many Aspects to Consider

- Layering
- Versioning
- Encoding
- Addressing
- Error Handling
- Flow Control
- Congestion Control
- Feedback
- Sequencing
- Batching
- Sync/Async
- Validation
- Trust
- Privacy

Who cares about waste?

Encoding

Don’t use text codecs!
Please please use binary codecs

Encoding

“…but it’s human readable…”

```
  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
 +-+-------------------------------------------------------------+
 |R|                       Frame Length                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-------------------------------+
 |    Version     |B|E|   Flags   |             Type              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-------------------------------+
 |R|                       Term Offset                           |
 +-+-------------------------------------------------------------+
 |                          Session ID                           |
 +---------------------------------------------------------------+
 |                           Stream ID                           |
 +---------------------------------------------------------------+
 |                            Term ID                            |
 +---------------------------------------------------------------+
 |                        Encoded Message                       ...
...                                                              |
 +---------------------------------------------------------------+
```

Encoding

SBE, Flat Buffers, Cap’n’Proto, ASN.1, etc.

Versioning

Versioning

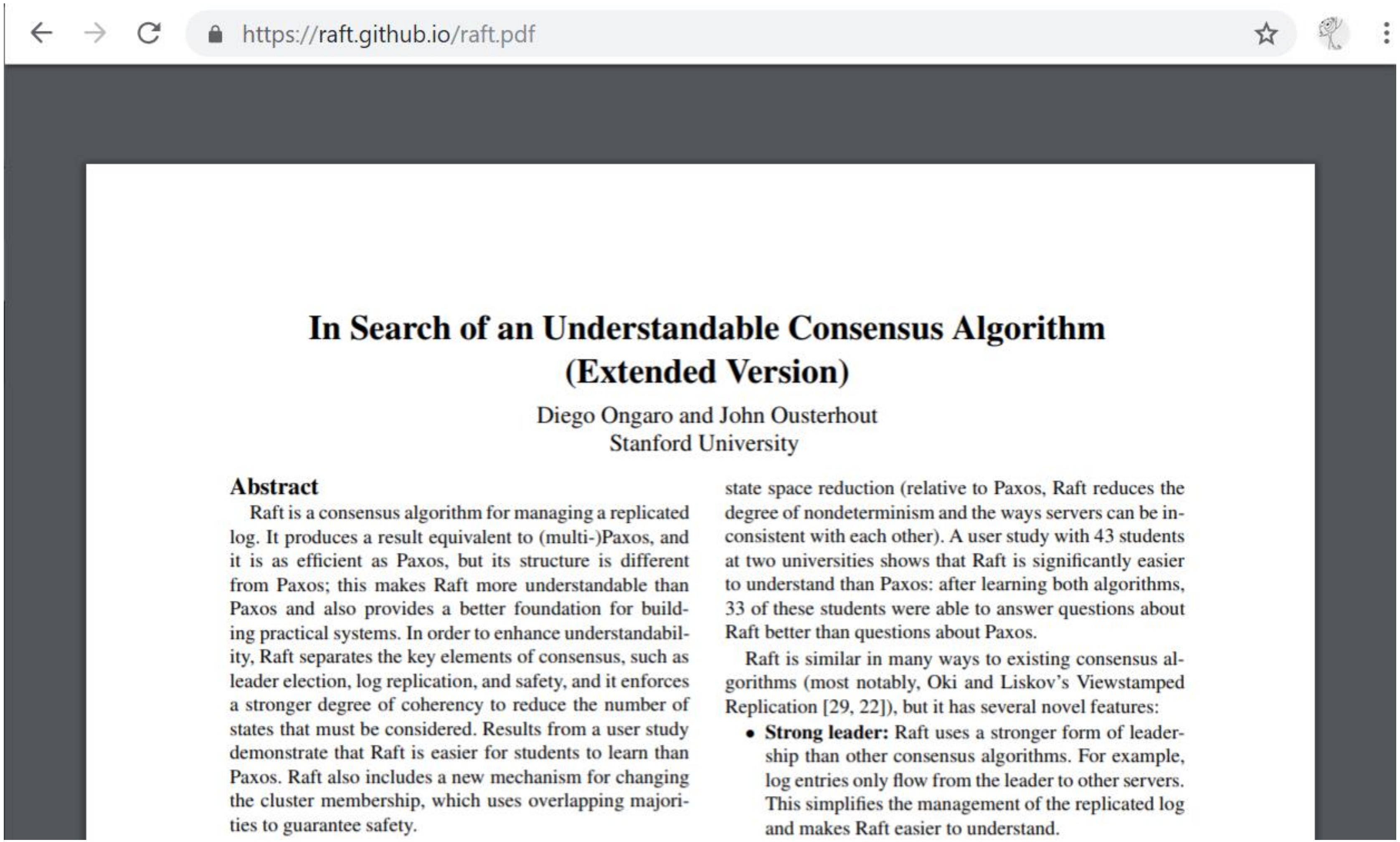
- **Protocols**: What conversation?

Versioning

- Protocols: What conversation?
- Messages: What encoding?

Versioning

- **Protocols**: What conversation?
- **Messages**: What encoding?
- **State**: What instance?



https://raft.github.io/raft.pdf

In Search of an Understandable Consensus Algorithm (Extended Version)

Diego Ongaro and John Ousterhout
Stanford University

Abstract

Raft is a consensus algorithm for managing a replicated log. It produces a result equivalent to (multi-)Paxos, and it is as efficient as Paxos, but its structure is different from Paxos; this makes Raft more understandable than Paxos and also provides a better foundation for building practical systems. In order to enhance understandability, Raft separates the key elements of consensus, such as leader election, log replication, and safety, and it enforces a stronger degree of coherency to reduce the number of states that must be considered. Results from a user study demonstrate that Raft is easier for students to learn than Paxos. Raft also includes a new mechanism for changing the cluster membership, which uses overlapping majorities to guarantee safety.

state space reduction (relative to Paxos, Raft reduces the degree of nondeterminism and the ways servers can be inconsistent with each other). A user study with 43 students at two universities shows that Raft is significantly easier to understand than Paxos: after learning both algorithms, 33 of these students were able to answer questions about Raft better than questions about Paxos.

Raft is similar in many ways to existing consensus algorithms (most notably, Oki and Liskov's Viewstamped Replication [29, 22]), but it has several novel features:

- **Strong leader:** Raft uses a stronger form of leadership than other consensus algorithms. For example, log entries only flow from the leader to other servers. This simplifies the management of the replicated log and makes Raft easier to understand.

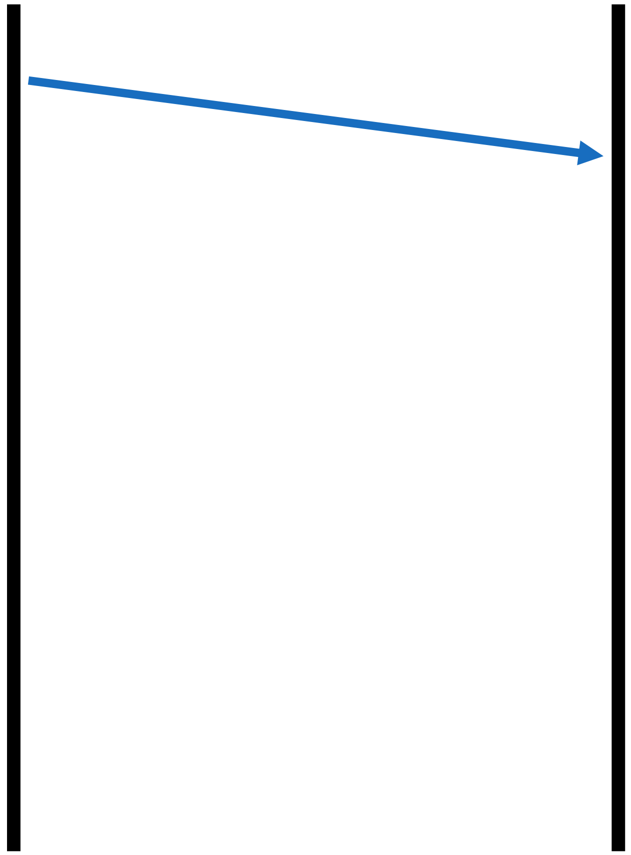
MPMC Queue

http://www.1024cores.net/

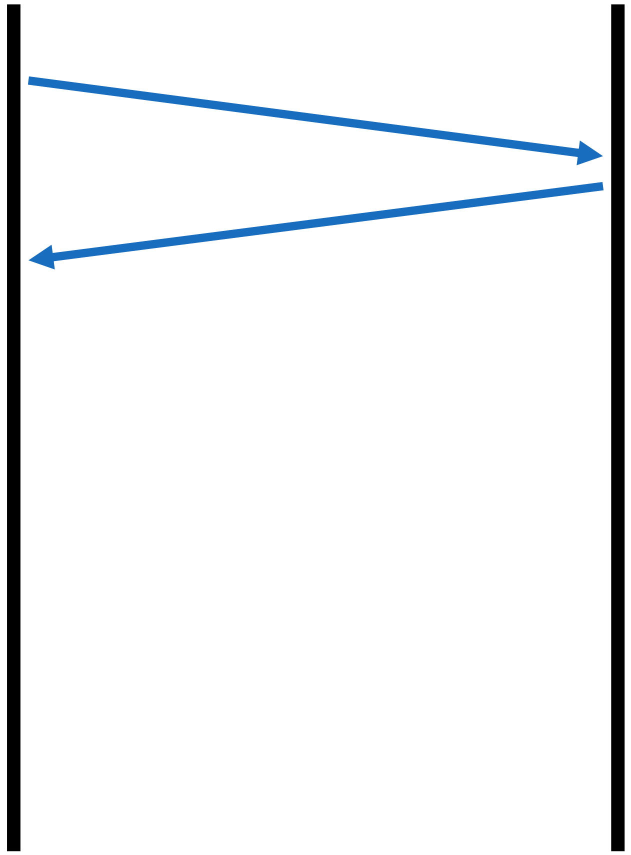
Sync vs Async

Synchronous

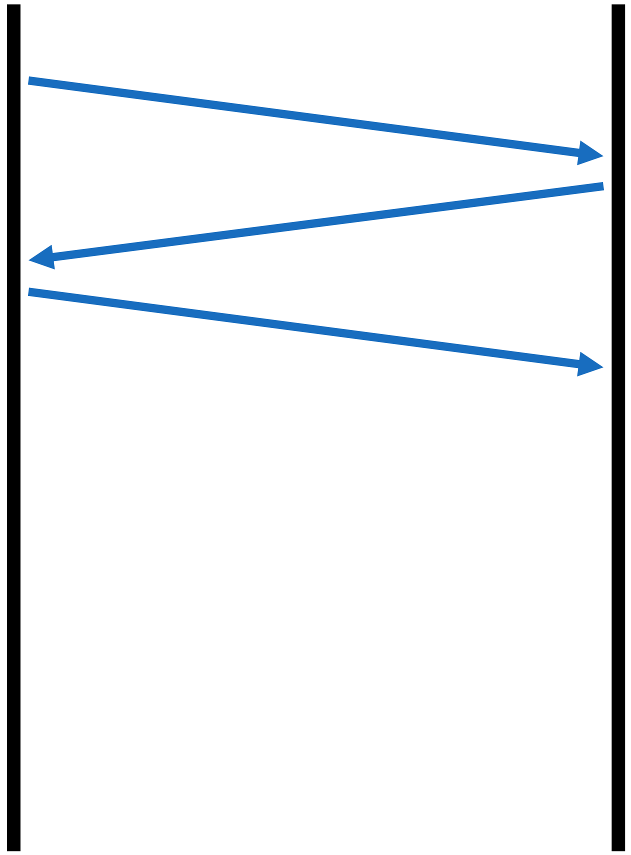
Synchronous



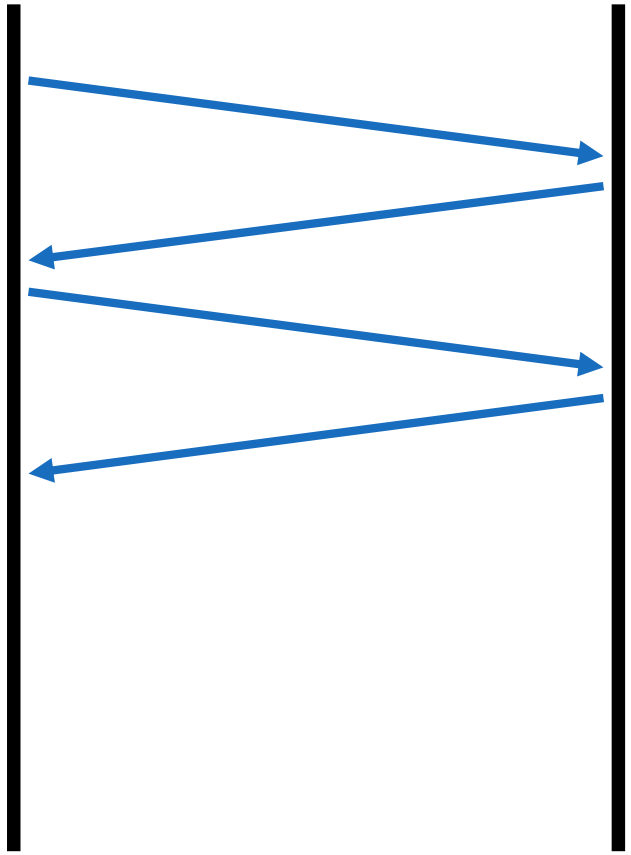
Synchronous



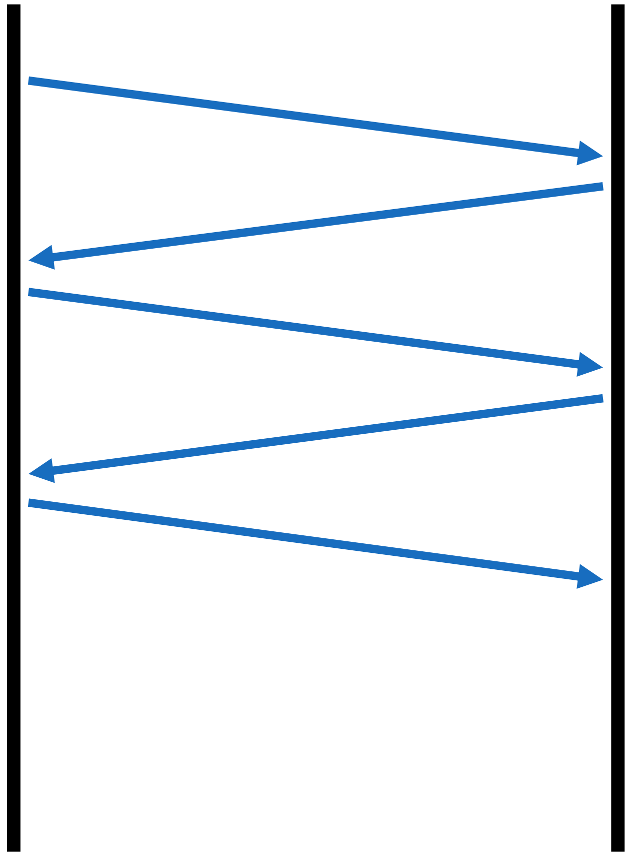
Synchronous



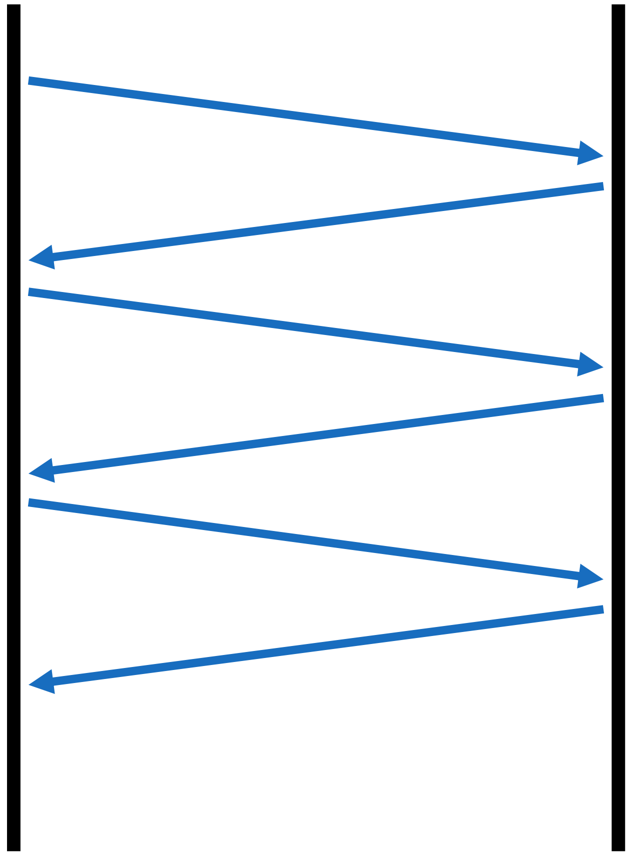
Synchronous



Synchronous

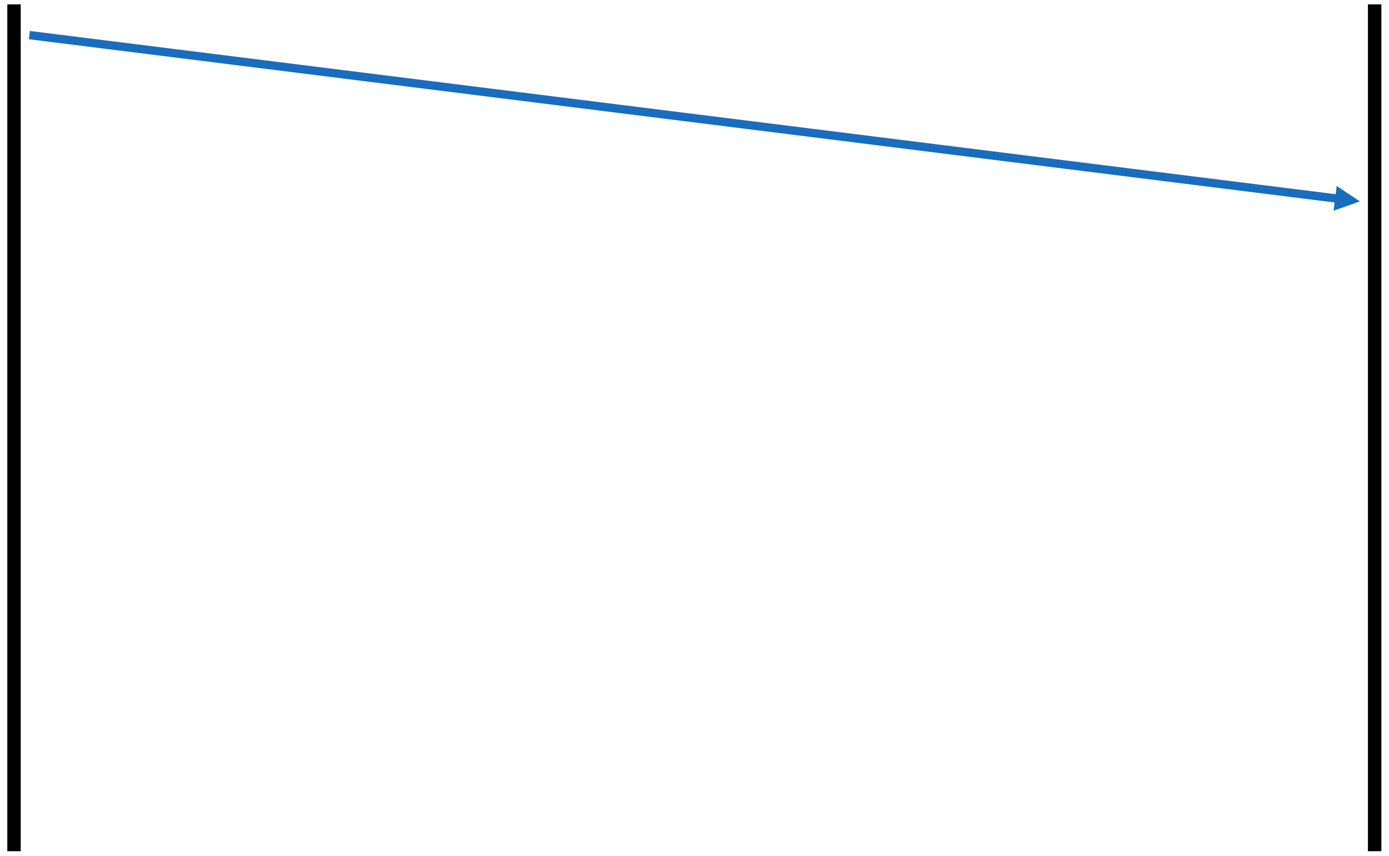


Synchronous

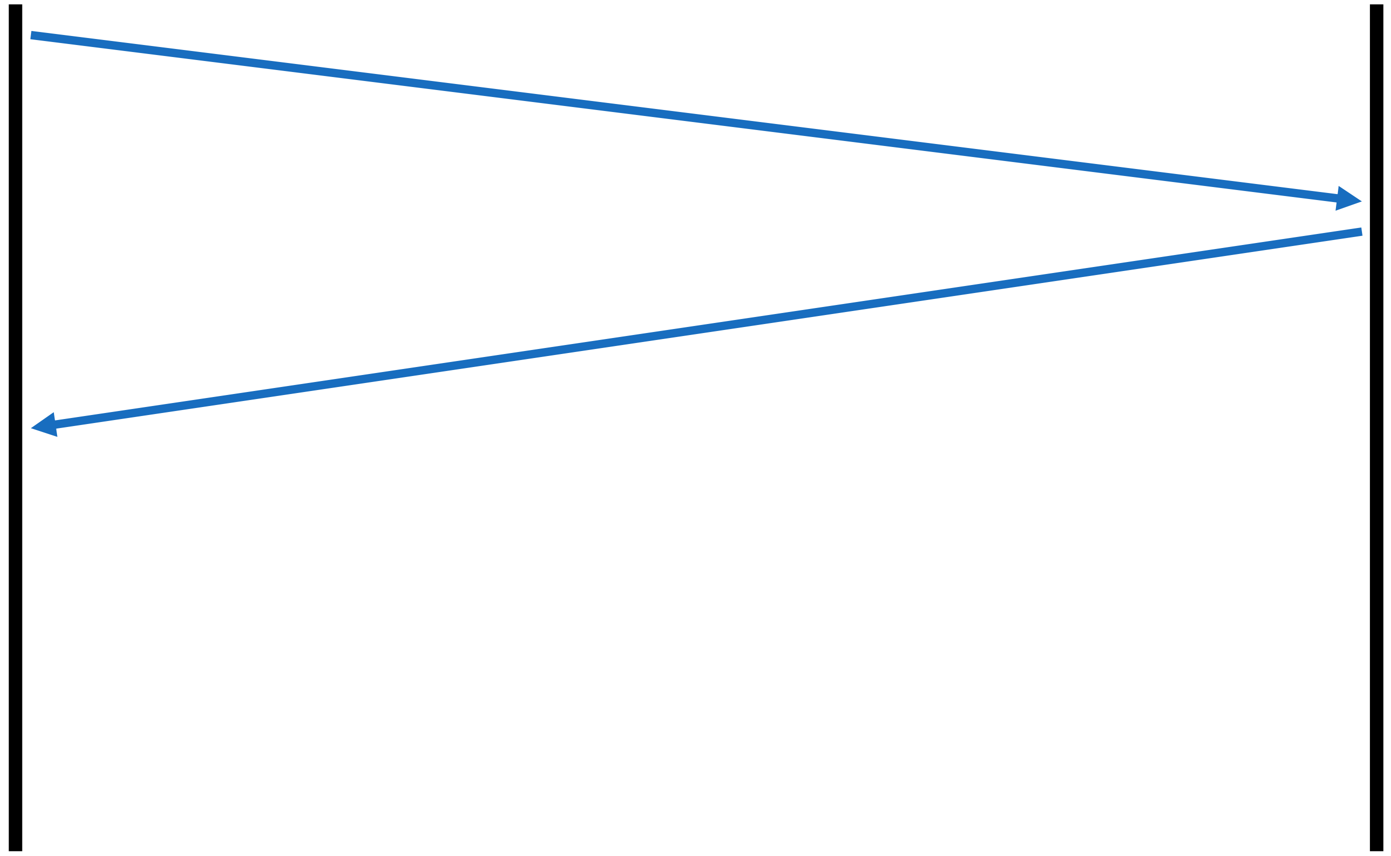


Synchronous

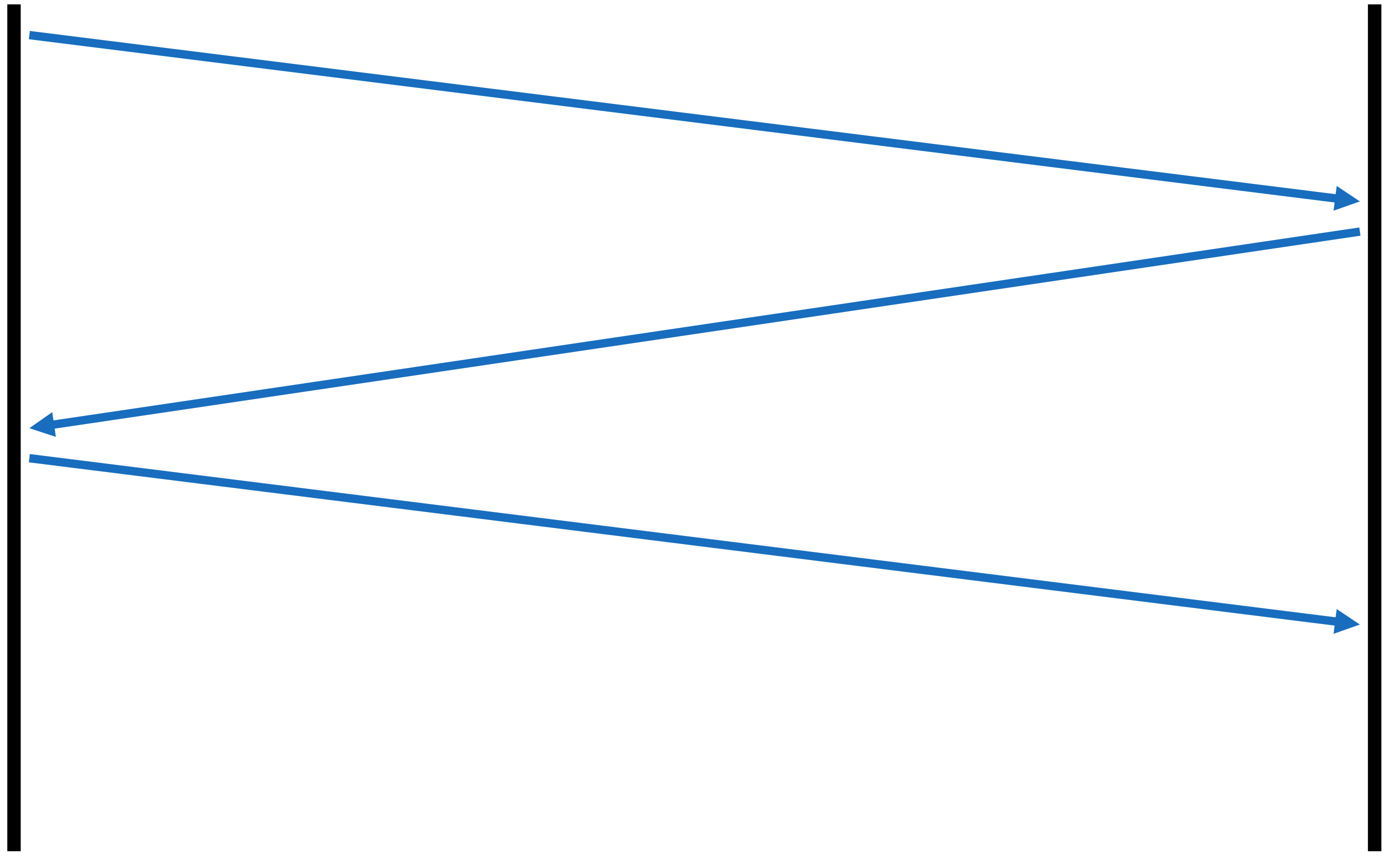
Synchronous



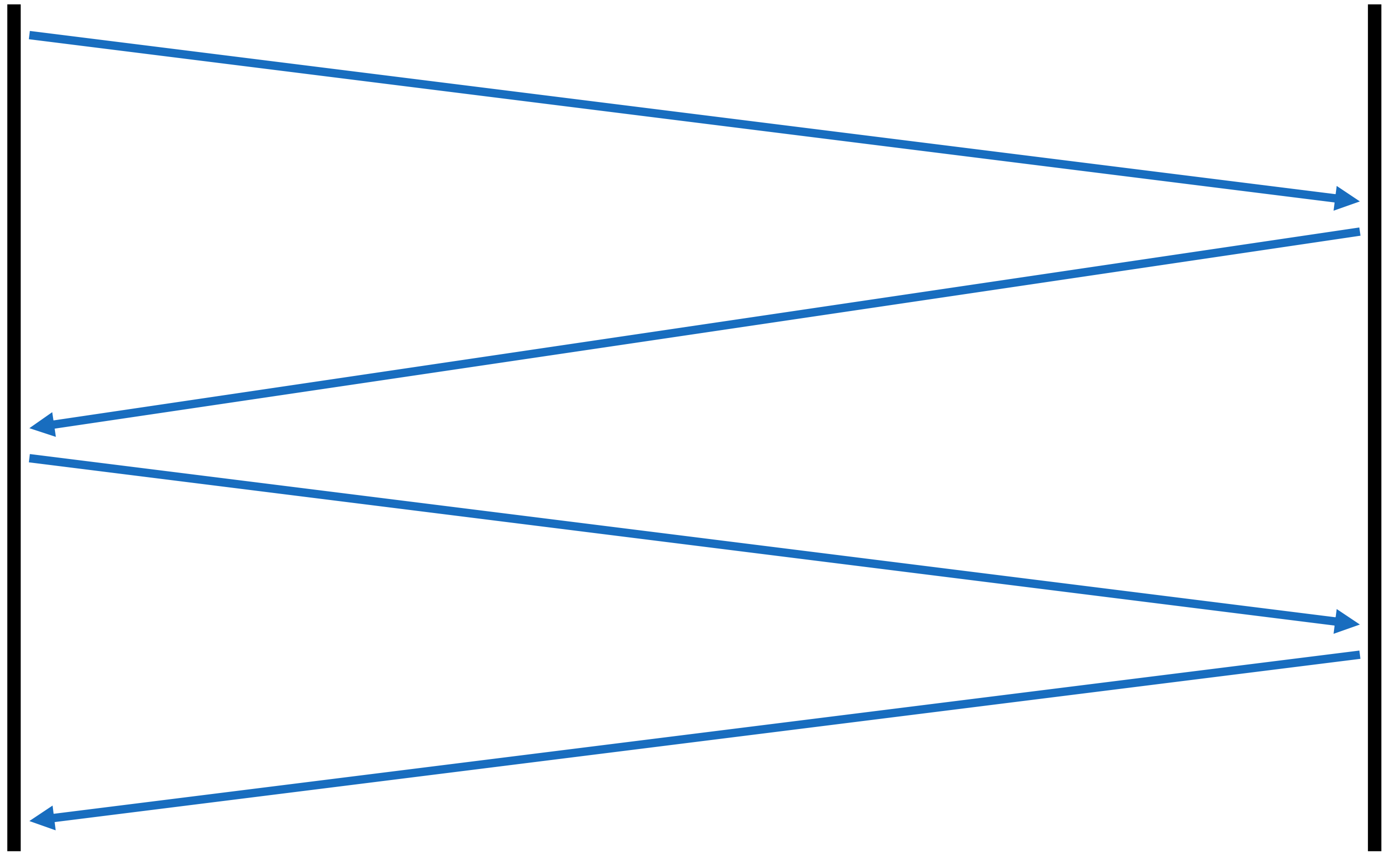
Synchronous



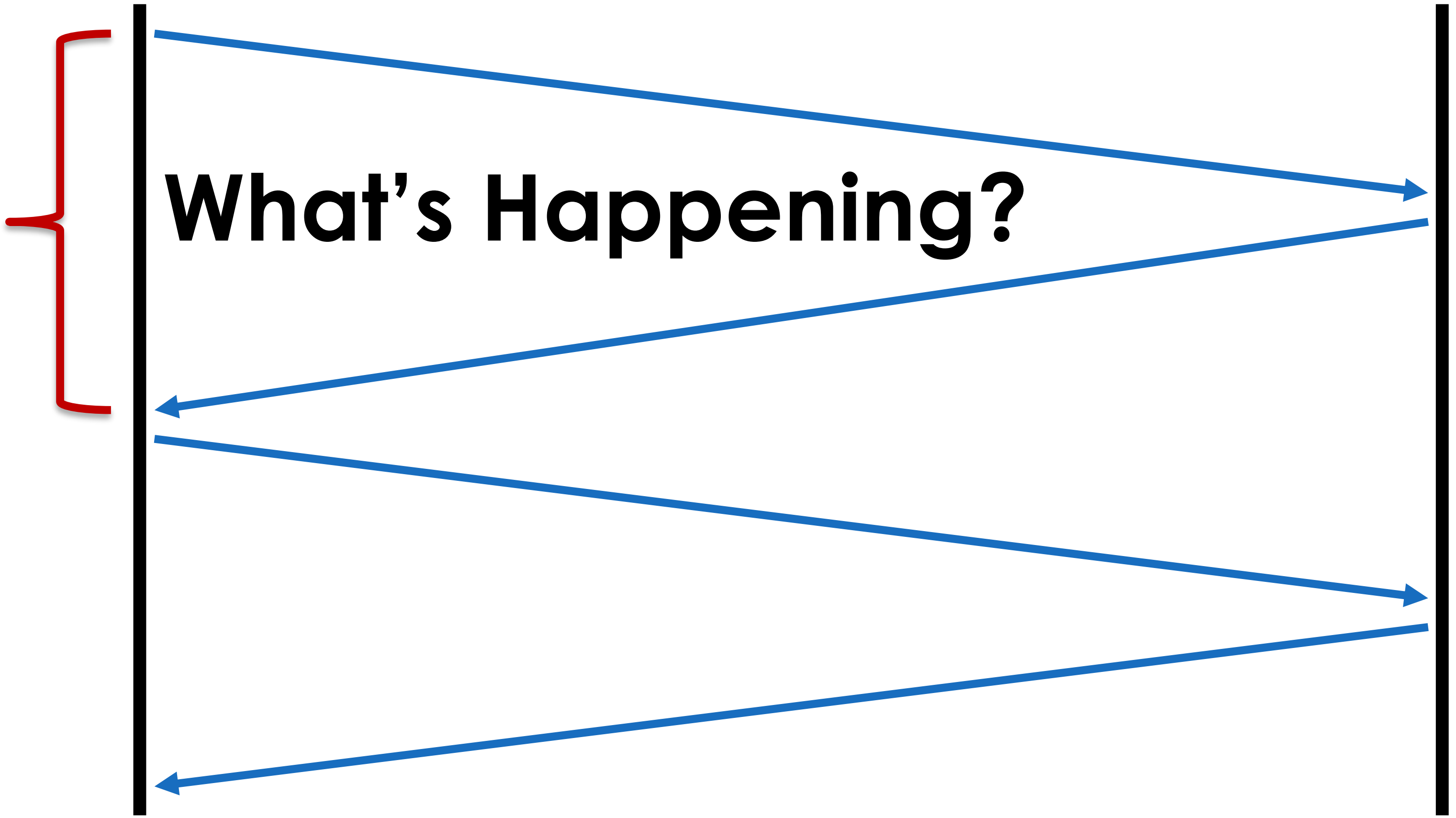
Synchronous



Synchronous



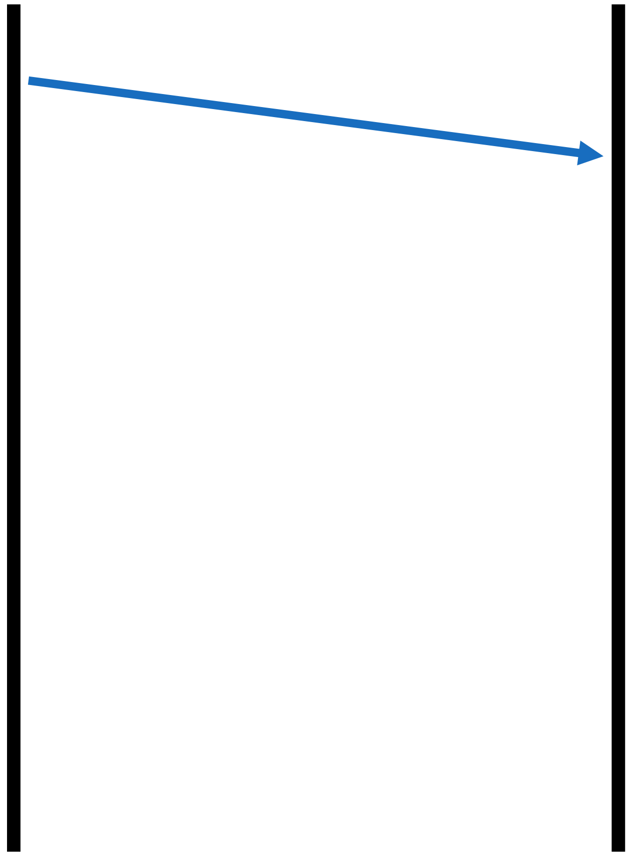
Synchronous



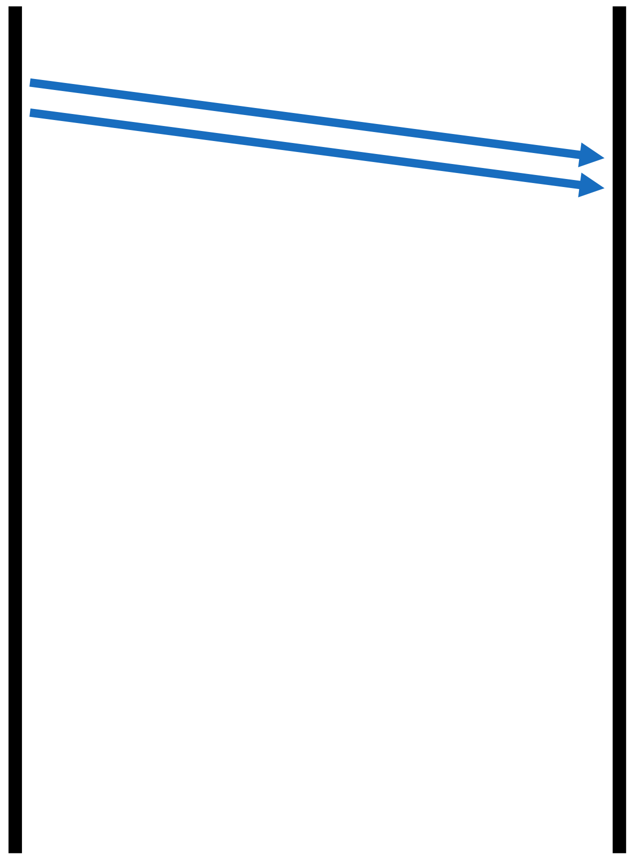
What's Happening?

Asynchronous

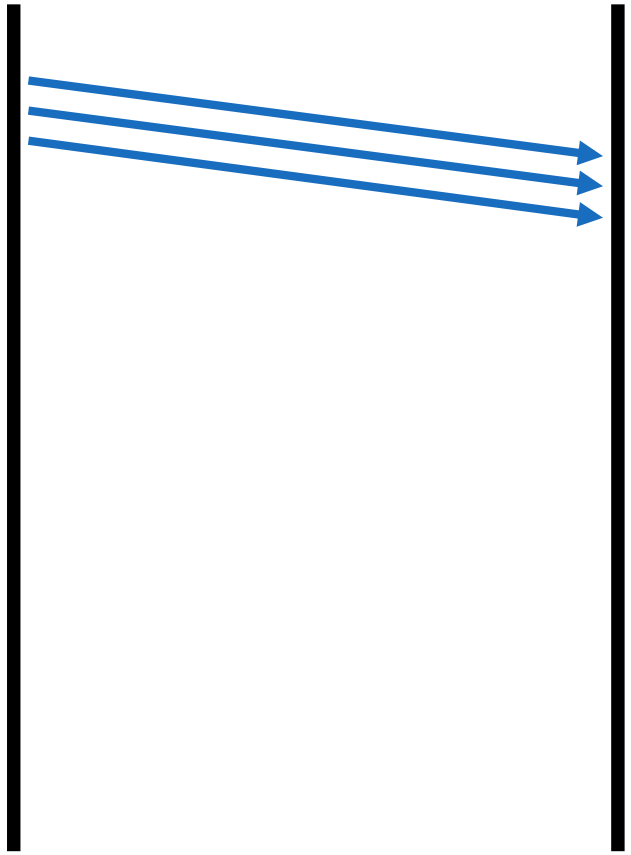
Asynchronous



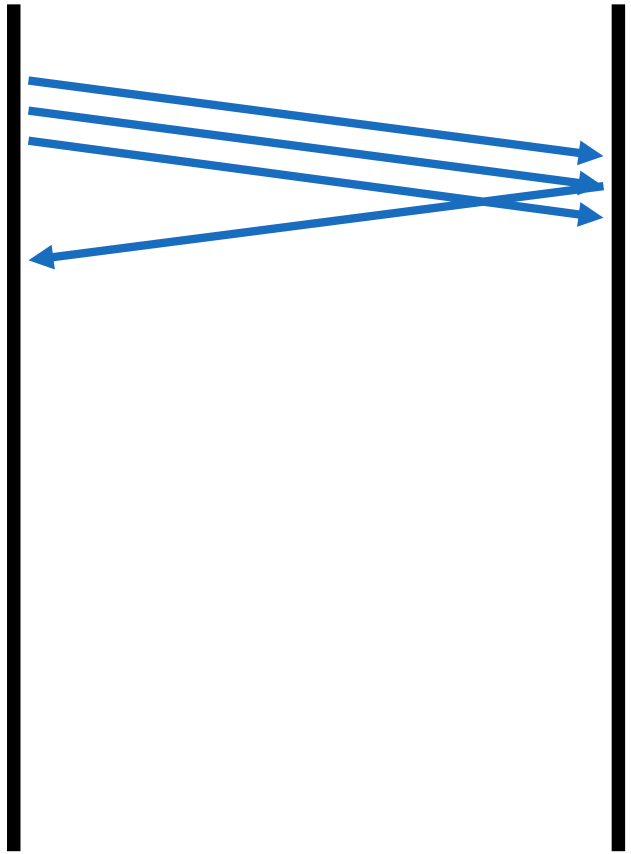
Asynchronous



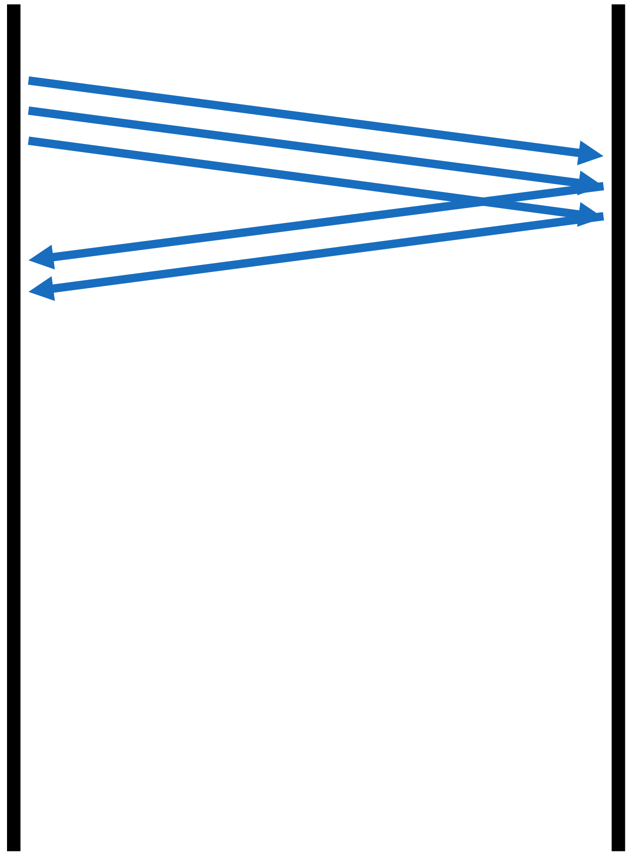
Asynchronous



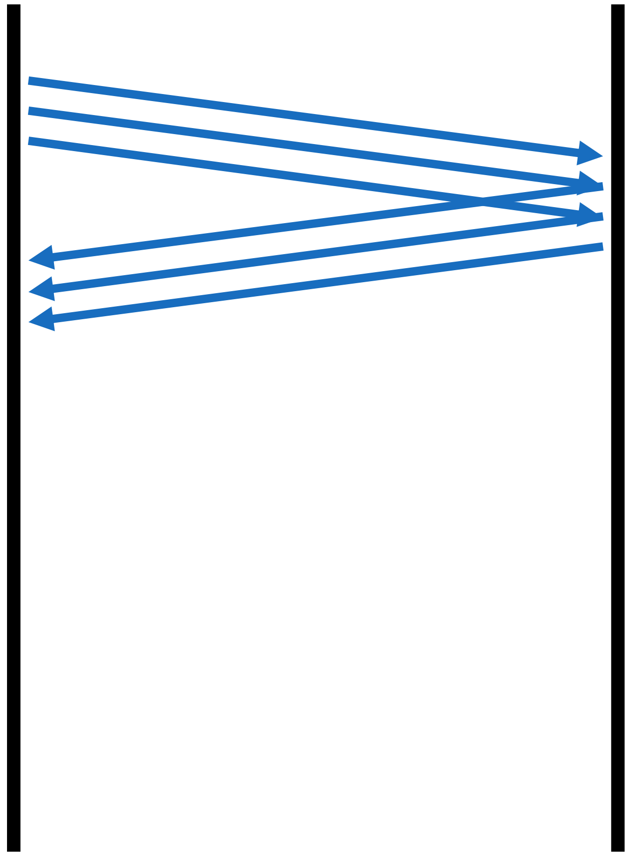
Asynchronous



Asynchronous

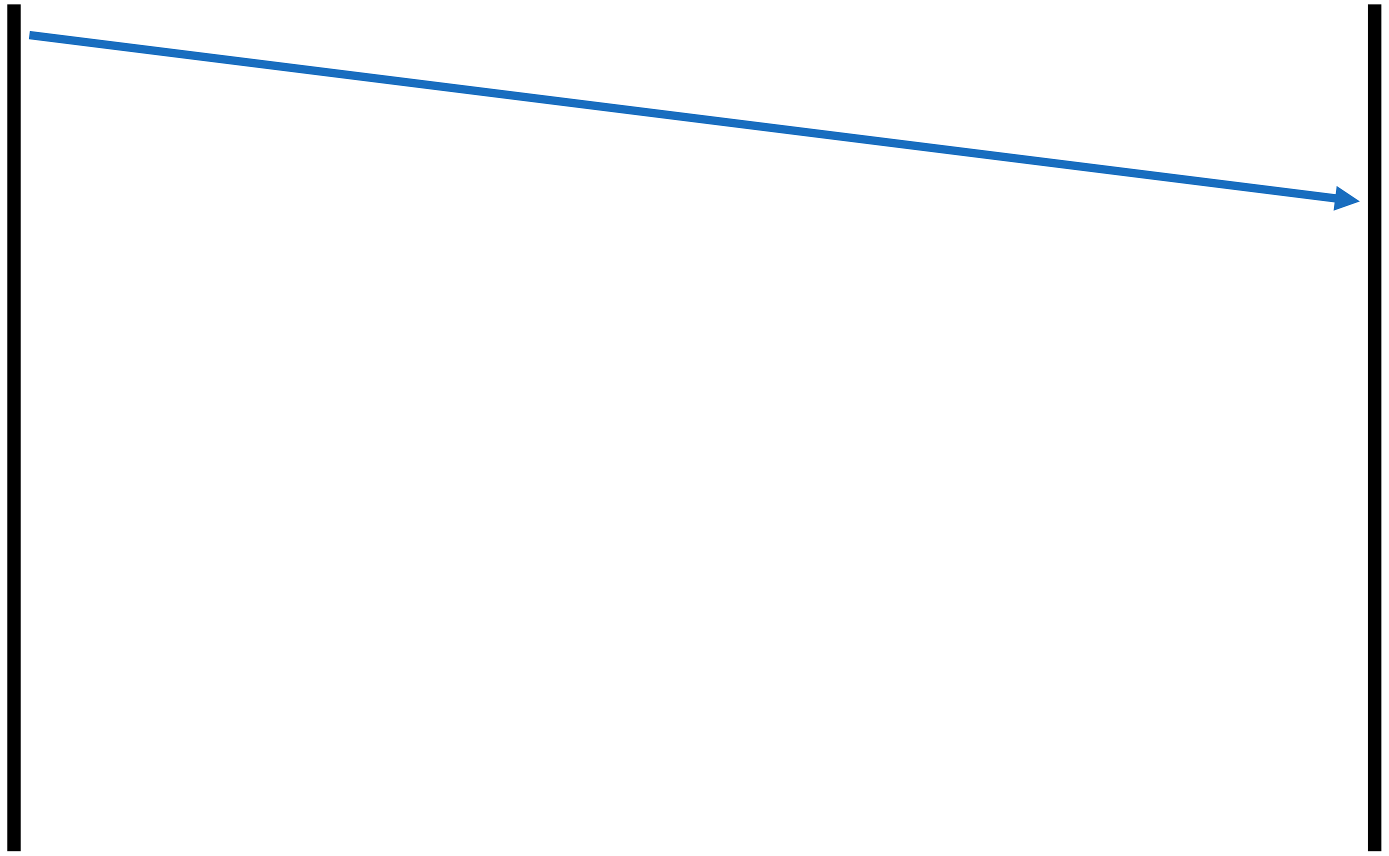


Asynchronous

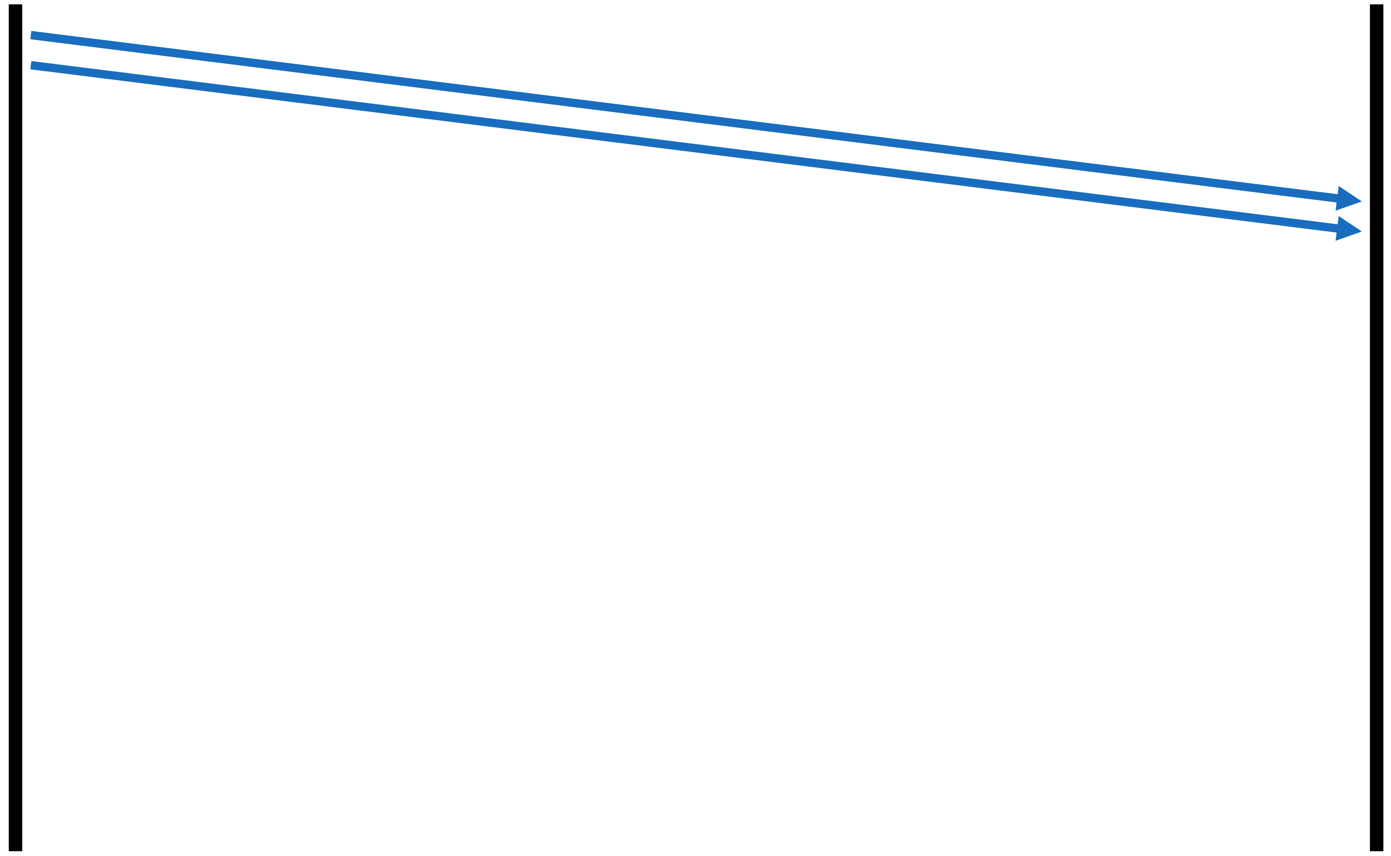


Asynchronous

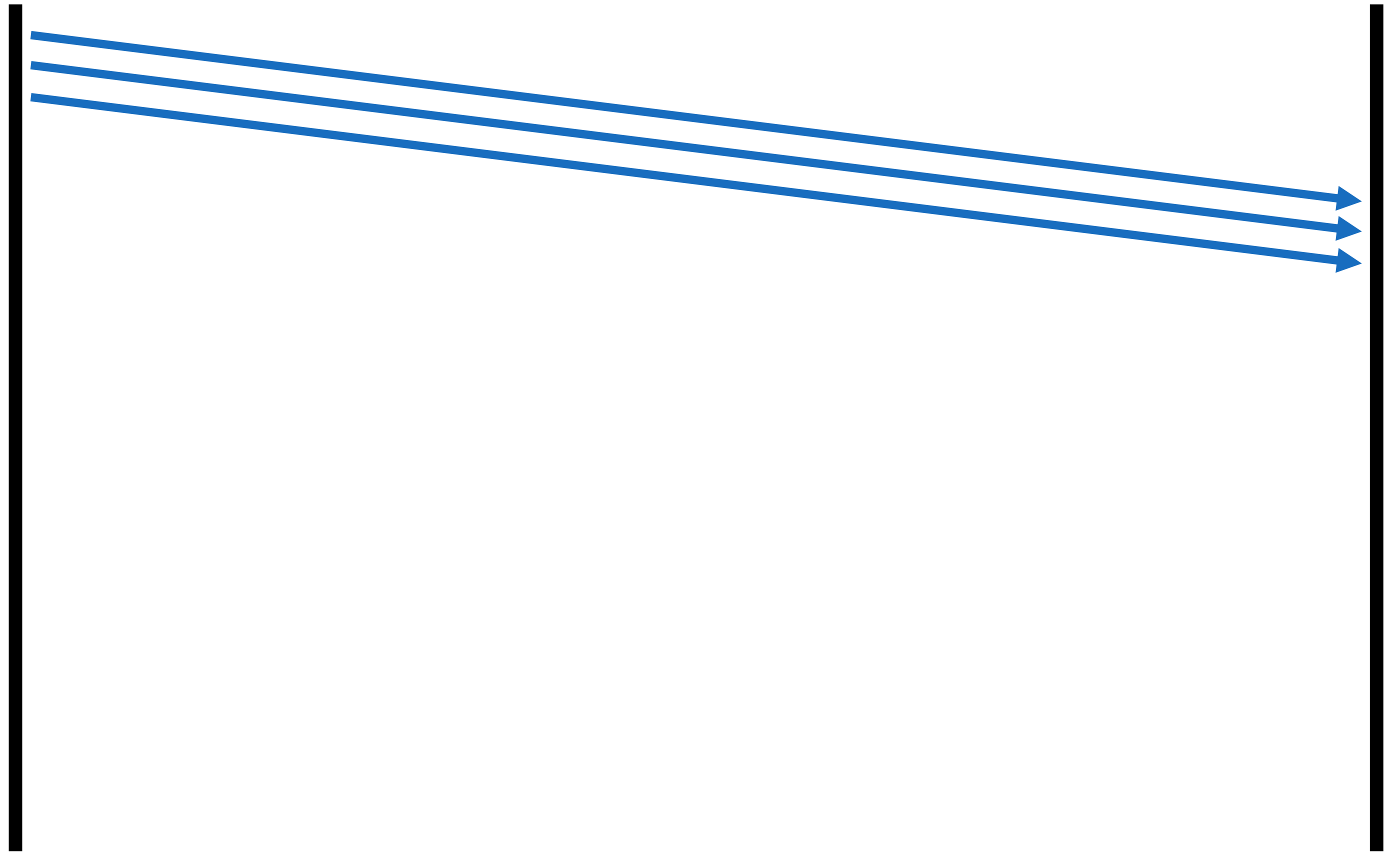
Asynchronous



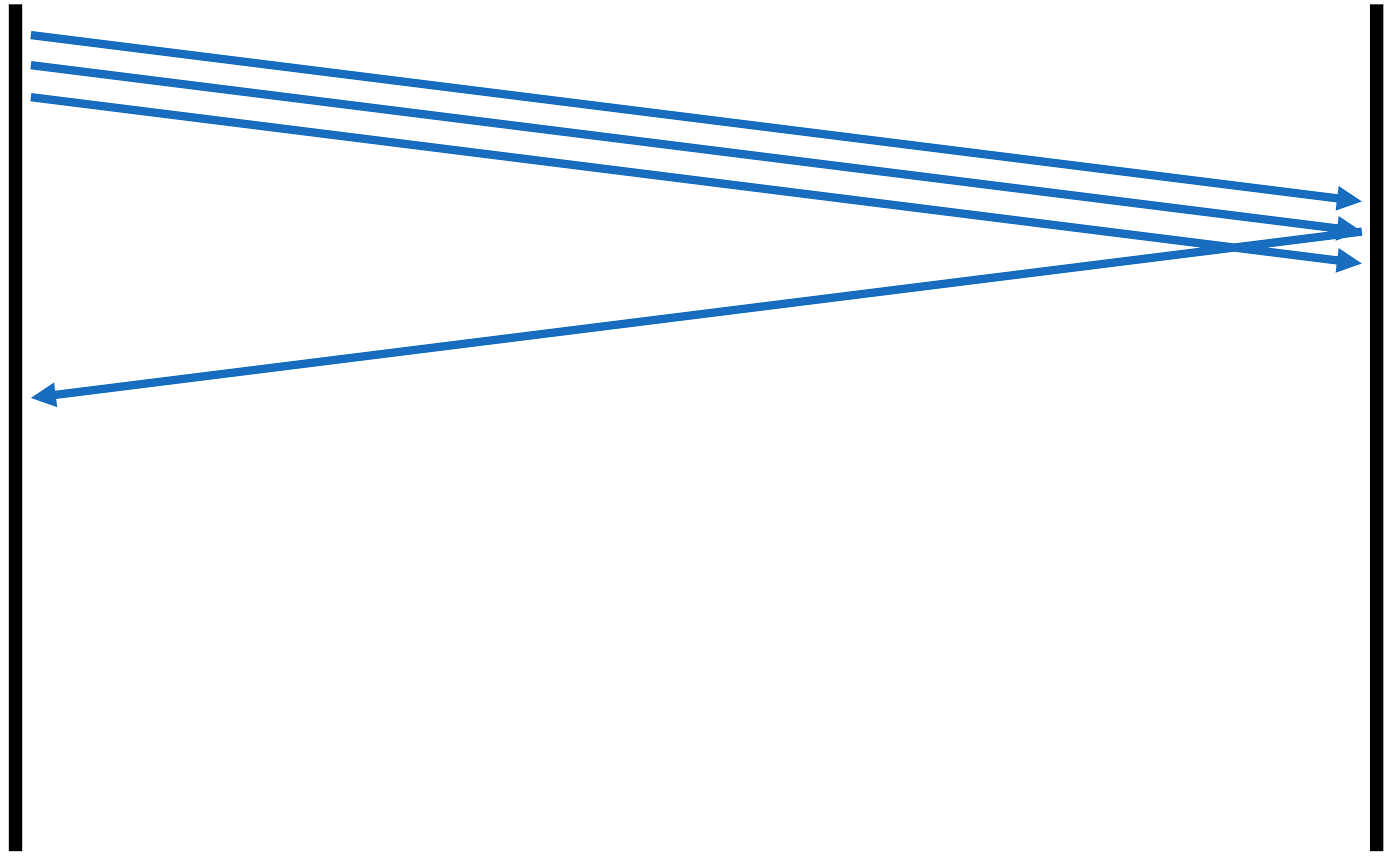
Asynchronous



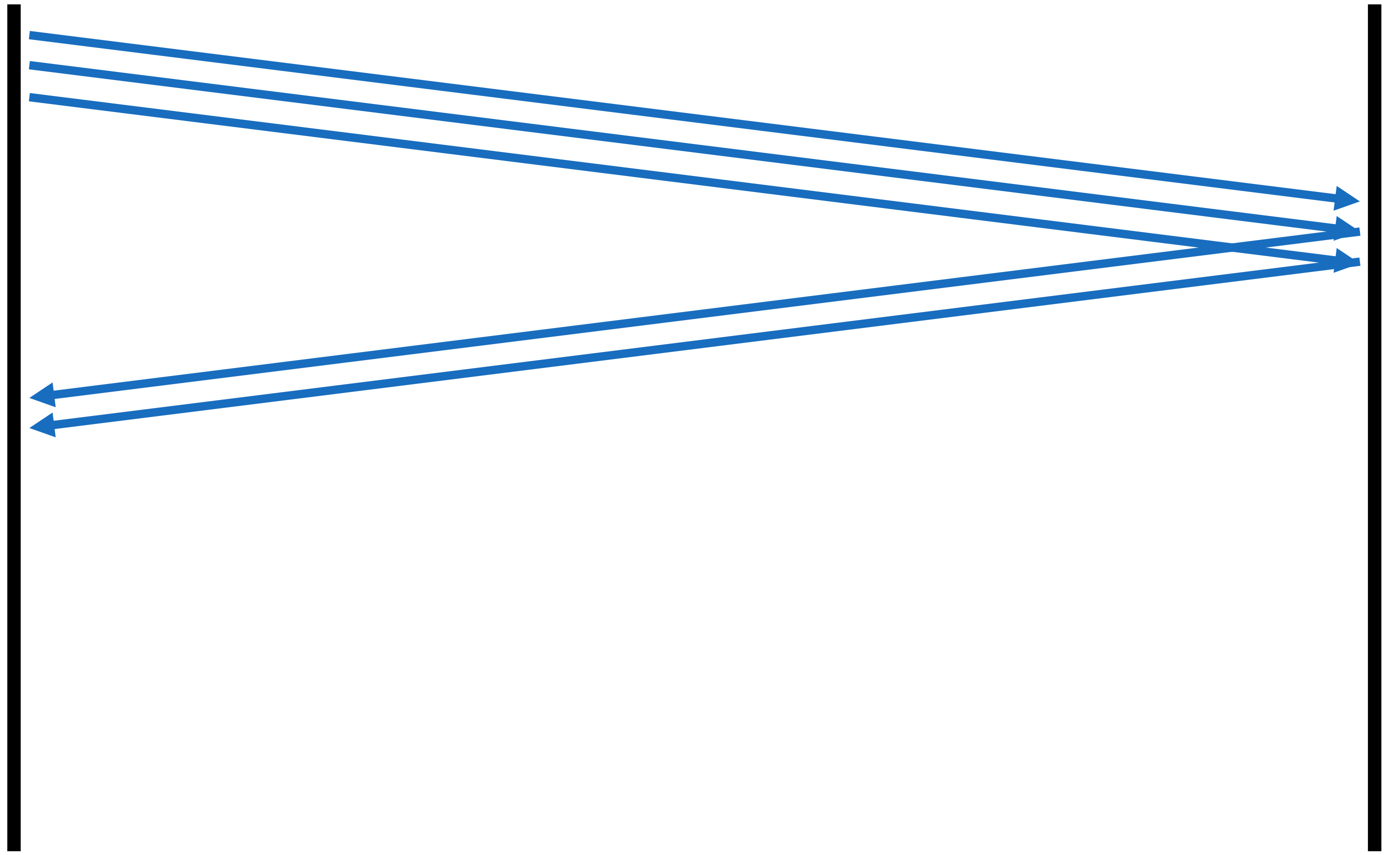
Asynchronous



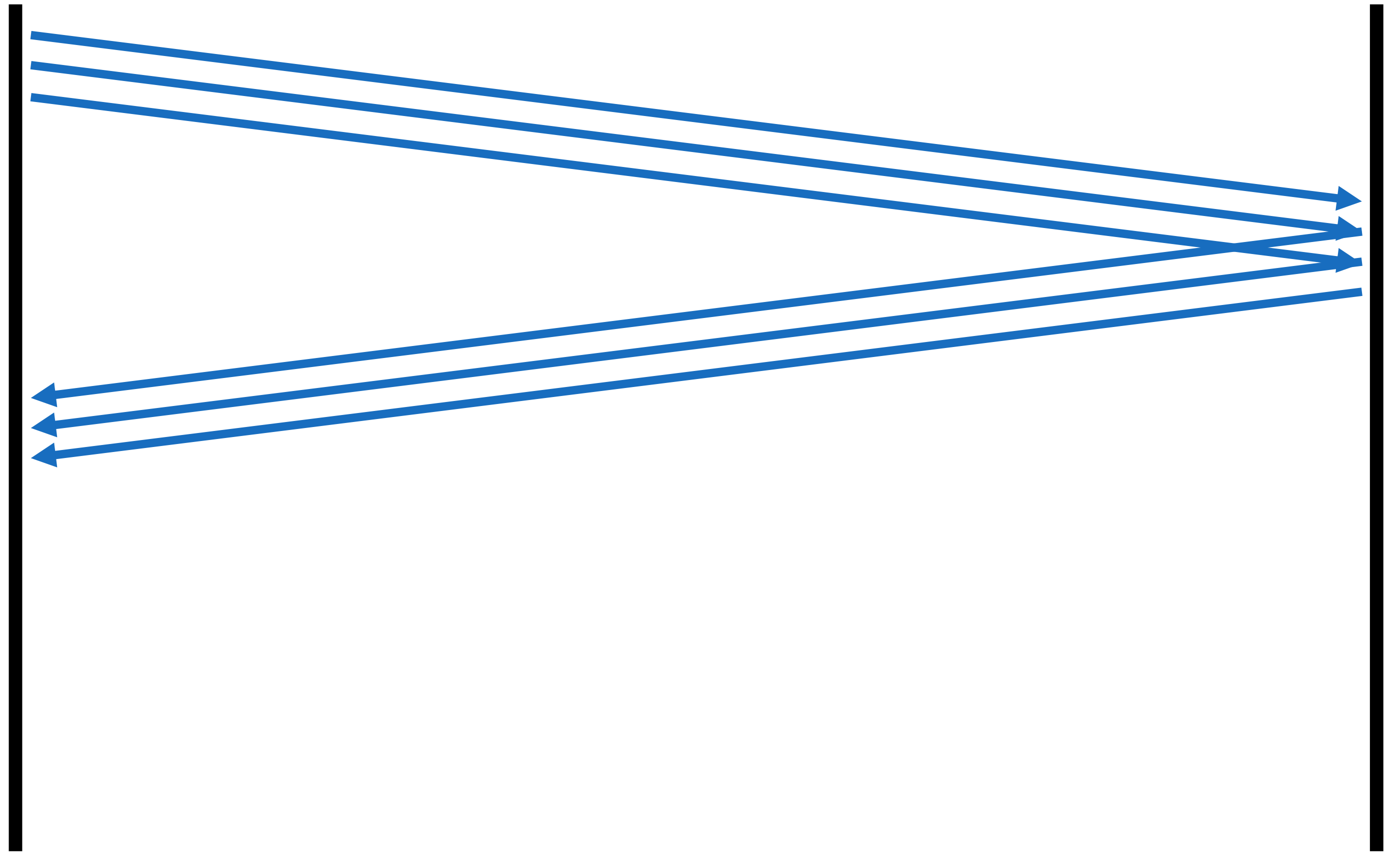
Asynchronous



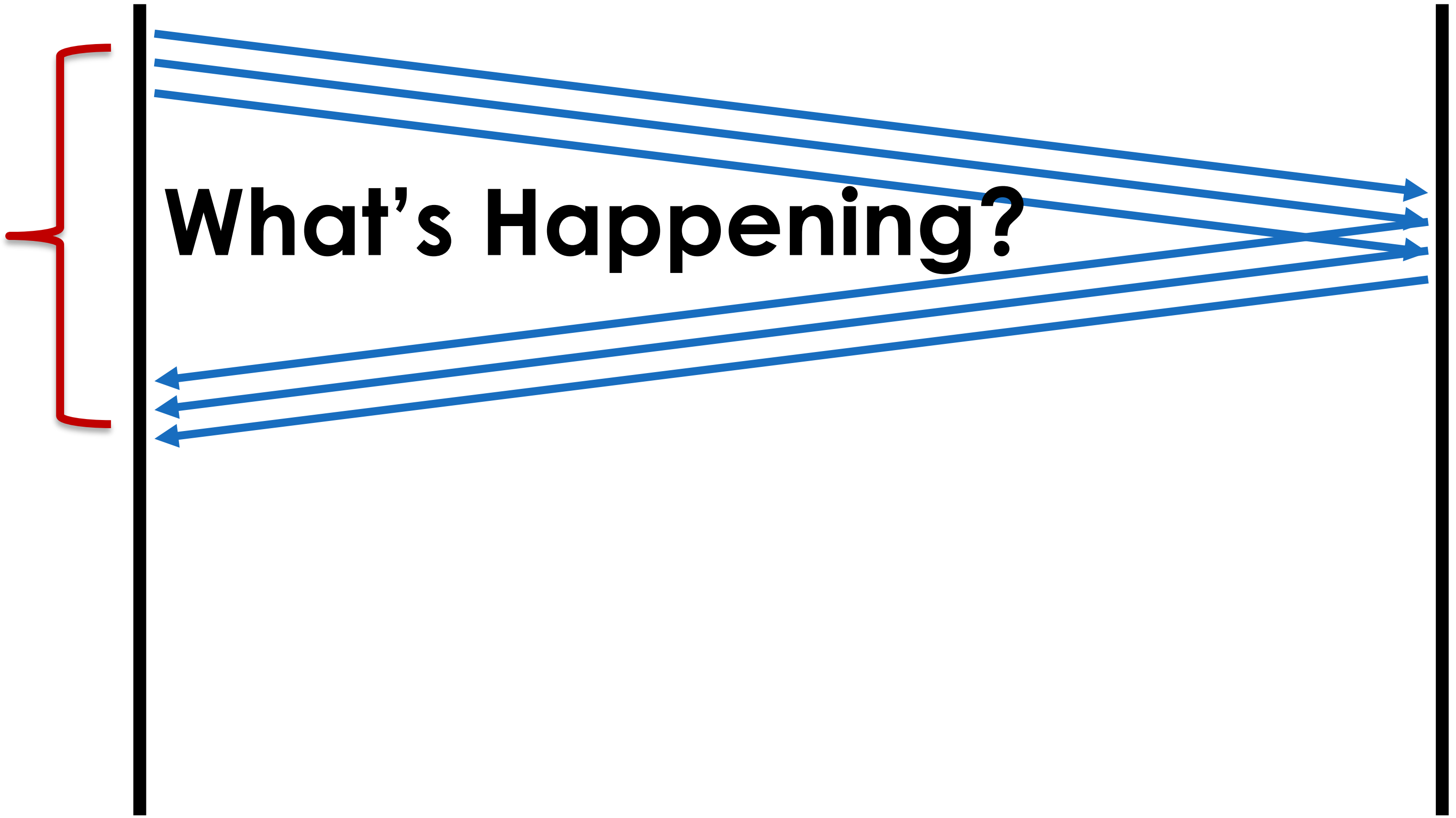
Asynchronous



Asynchronous



Asynchronous



“…but, but, synchronous is so much easier…”

Synchronous => Blocking

…it’s about managing state via feedback

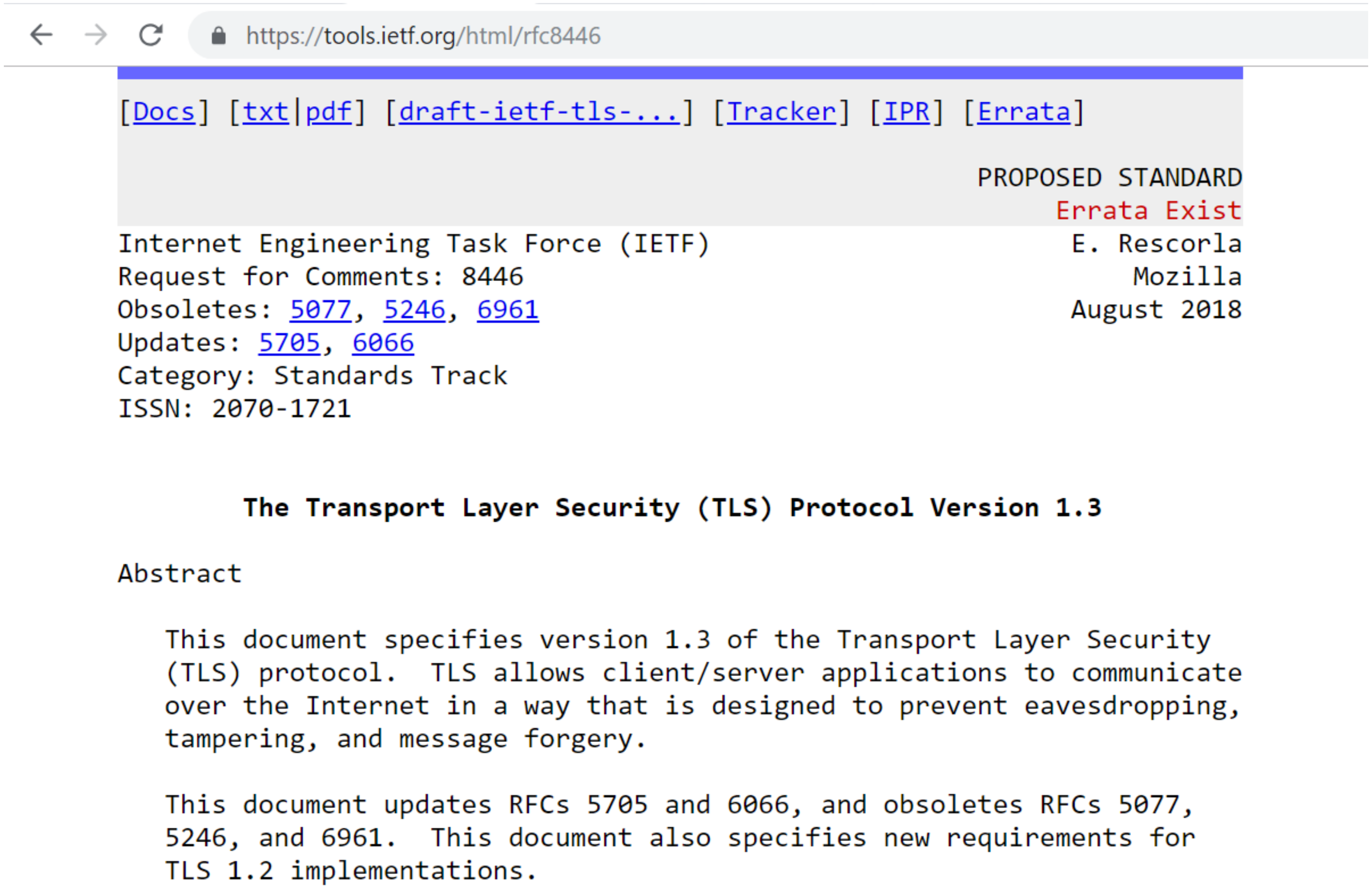
When designing an API,
wrap Async with Sync
if you must ☹

RPC / HTTP / TCP

TCP → TCP Fast Open → QUIC

HTTP/1.1 → SPDY → HTTP/2

TLS 1.2 → TLS 1.3



https://tools.ietf.org/html/rfc8446

[Docs] [txt|pdf] [draft-ietf-tls-...] [Tracker] [IPR] [Errata]

```
                                                       PROPOSED STANDARD
                                                            Errata Exist
Internet Engineering Task Force (IETF)                       E. Rescorla
Request for Comments: 8446                                       Mozilla
Obsoletes: 5077, 5246, 6961                                  August 2018
Updates: 5705, 6066
Category: Standards Track
ISSN: 2070-1721
```

The Transport Layer Security (TLS) Protocol Version 1.3

Abstract

This document specifies version 1.3 of the Transport Layer Security (TLS) protocol. TLS allows client/server applications to communicate over the Internet in a way that is designed to prevent eavesdropping, tampering, and message forgery.

This document updates RFCs 5705 and 6066, and obsoletes RFCs 5077, 5246, and 6961. This document also specifies new requirements for TLS 1.2 implementations.

0-RTT and replay attacks

Batching

Etiquette of a request

100 GigE ?

Future of Syscalls

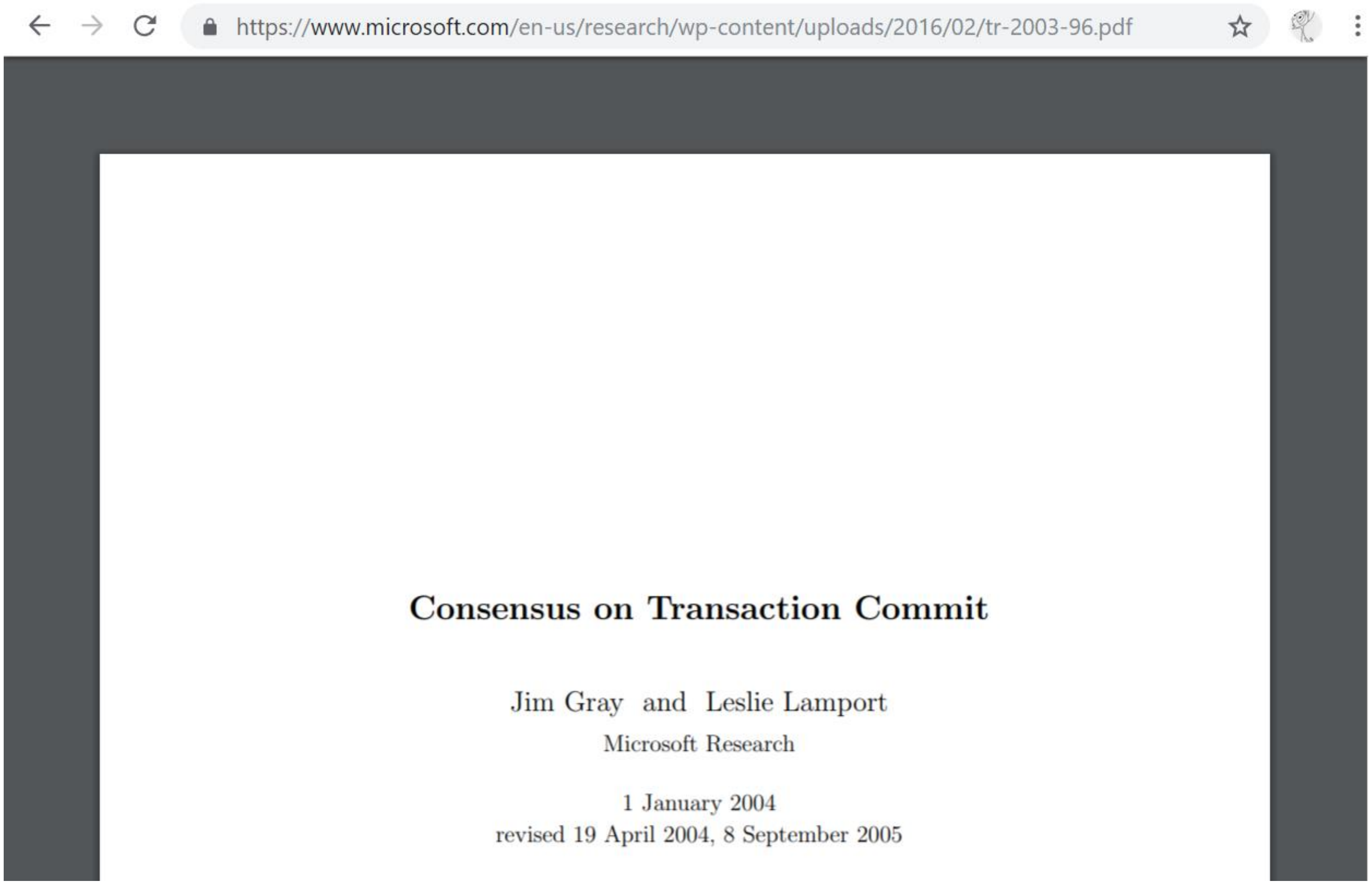
`sndmmsg, rcvmmsg()`

Onload, VMA

DPDK, ef_vi

Snake Oil Protocols

2PC / XA



https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2003-96.pdf

Consensus on Transaction Commit

Jim Gray and Leslie Lamport
Microsoft Research

1 January 2004
revised 19 April 2004, 8 September 2005

“Two-Phase Commit is not fault tolerant because it uses a single coordinator whose failure can cause the protocol to block”

Guaranteed Delivery™

Applications should have feedback & recovery protocols

Protocol Layering

“What can we depend on?”

Wrapping up…

Are protocols significant to software development?

Question,
Hypothesis,
Prediction,
Experiment,
Analysis

Falsifiability

Conway’s Law

@mjpt777

https://github.com/real-logic/aeron

“A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.”

- Leslie Lamport