Search

Subscribe now

Get the highlights in your inbox every week.

Your email...

Your location...

Subscribe

Privacy Statement

3 facts every Enterprise Architect w

becoming commonplace communication protocol. Learn how it
1 and HTTP/2 to provide faster web services.

nan (Red Hat, Lead Contributor)

Photo by NASA on Unsplash

HTTP/3 is the next-generation communication protocol for applications running on the internet. The protocol is not yet being used for general purposes in the way HTTP/1.1 and, to some extent, HTTP/2 are, but nonetheless many companies are preparing for its arrival. At some point, it will gain wide adoption. Thus, getting a general understanding of HTTP/3 is useful for Enterprise Architects interested in developing next-generation architectures.

The following is a list of five things that we think every Enterprise Architect needs to know about HTTP/3.

1. HTTP/3 is driven by the need for speed
2. HTTP/3 evolved from QUIC
3. HTTP/3 uses UDP
4. HTTP/3 addresses the head-of-line blocking problem
5. HTTP/3 injects QUIC into the OSI and TCP/IP Models

Let's look at the details of each.

HTTP/3 is driven by the need for speed

event messaging, and other types of streaming data proliferate to a broader set of applications. HTTP/3 is intended to meet this demand.

HTTP/2 is fast, but it has design limitations that HTTP/3 addresses. You'll read about these limitations and how

ıoment. The bottom line is that HTTP/3 is designed for greater efficiency in web
t how HTTP3 evolved.

from QUIC

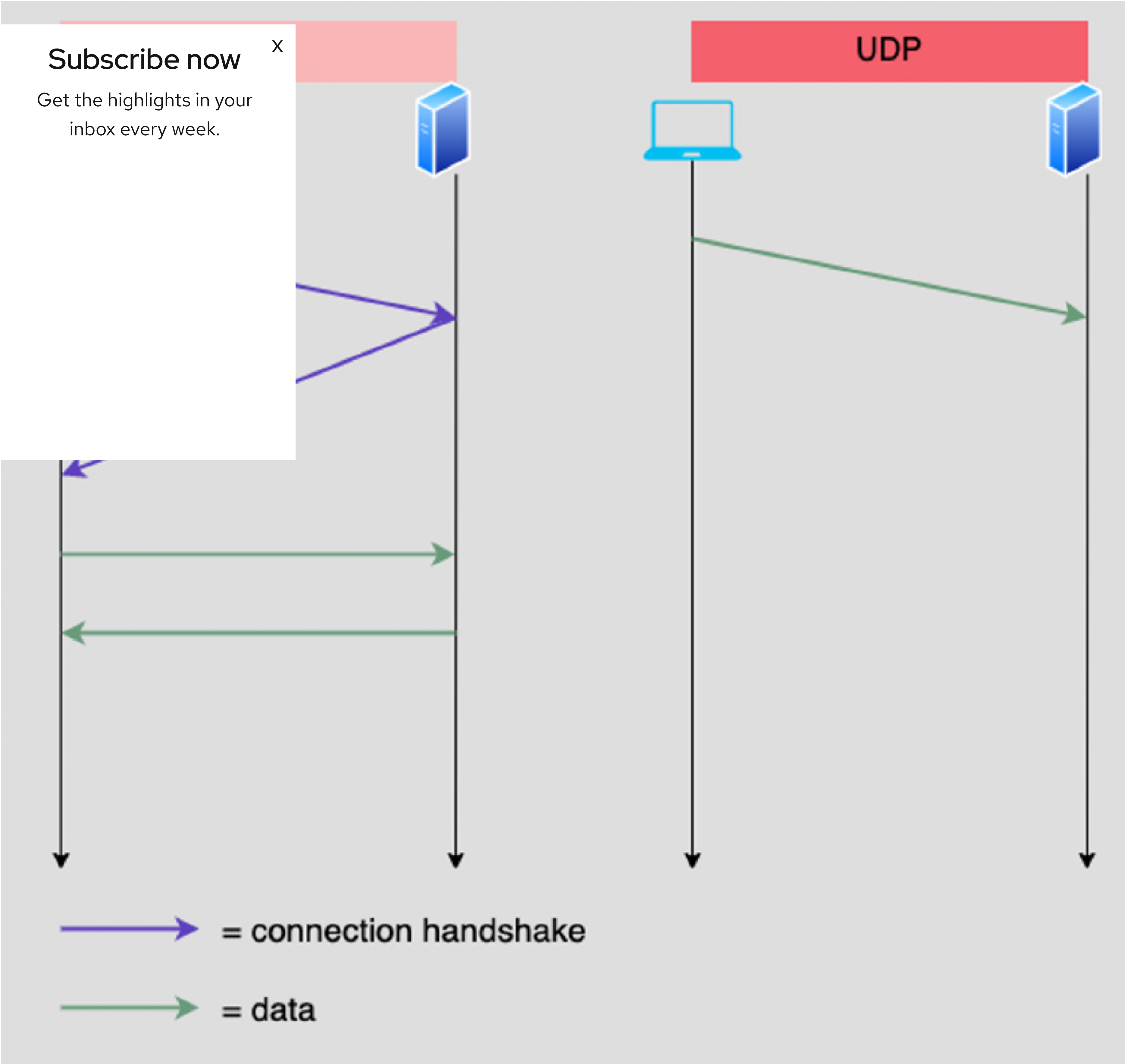
UDP Internet Connections. QUIC was created at Google by Jim Roskind, who released
pecification in 2012. In 2018 QUIC was combined with the HTTP working group of the
e (IETF) to create the HTTP/3 specification. The goal is to have HTTP/3 become a
e order of HTTP/1.1 or HTTP/2.

use the Transmission Control Protocol (TCP) as their data transport protocol, QUIC
ng to work with User Datagram Protocol (UDP). Using UDP incurs some tradeoffs,
ɔut one of the main benefits that UDP offers is that it's a faster, more efficient way to
unterpart.

HTTP/3 uses UDP

As mentioned above, HTTP/1.1 and HTTP/2 use TCP as their transmission protocol. HTTP/3 and QUIC, the protocol from which HTTP/3 was derived, use UDP. The difference matters. Here's why:

Fundamentally UDP is a fire and forget protocol. You send a data packet to a target, and that's that! TCP requires a lot of back and forth data exchange between sender and target to establish a network connection. Then, even after the connection is established, there's more back and forth because TCP requires that the sender receive an acknowledgment from the target every time a data packet is sent. This back and forth eats up time. UDP has no back and forth connection *handshake*, nor does it require acknowledgment from a target when data packets are received. It just sends packets of data to the destination and hopes for the best. (See Figure 1, below.)



Subscribe now

Get the highlights in your inbox every week.

Figure 1: TCP requires a handshake to establish a network connection between sender and target; UDP does not.

Eliminating all the back and forth activity makes UDP faster than TCP. But, there is a tradeoff. Under UDP, packets can get lost. Because there's no acknowledgment from the target, the sender never knows about the packet loss, and thus there's no way to recover. TCP, on the other hand, is designed to recover from packet loss. TCP notices that no acknowledgment has been received and resends the packet. TCP is slower, but it is fault-tolerant. Thus the tradeoff. TCP gives you fault tolerance. UDP gives you speed. The trick is to get both fault tolerance and speed. HTTP/3 provides both. Also, HTTP/3 addresses another problem: Head-of-line blocking.

HTTP/3 addresses the head-of-line blocking problem

One of the challenges that HTTP/3 addresses is the network latency bottleneck created by the head-of-line block. To understand the nature of head-of-line blocking, it's useful to review how data travels over the internet.

Essentially, head-of-line blocking is when one request blocks another from completing. The problem exists under HTTP/1.1 and HTTP/2 because these use TCP as the transport protocol. Take a look at Figure 2 below. It shows an illustration of the head-of-line blocking problem. An explanation follows.

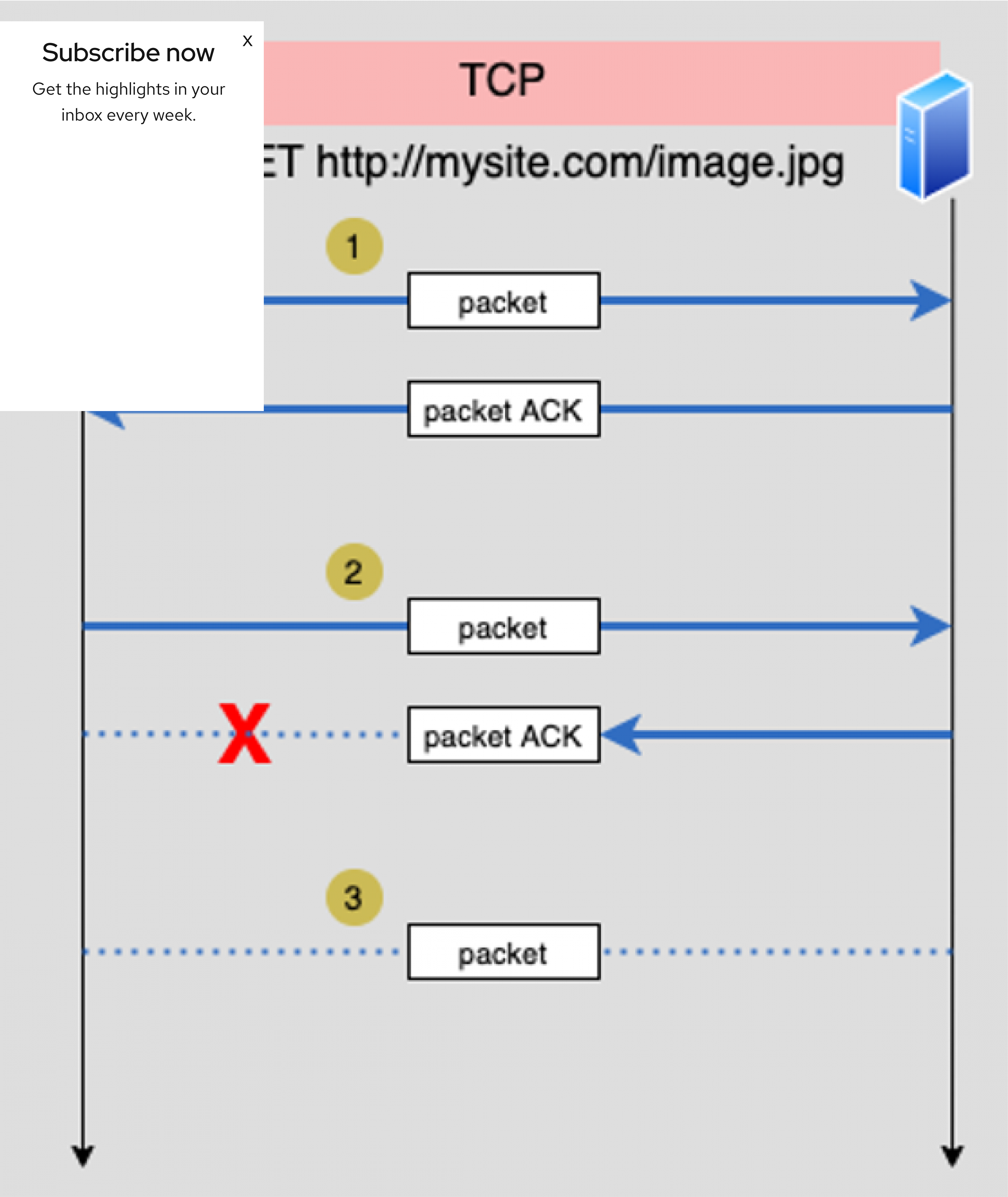


Figure 2: To facilitate the continuous flow of data packets, the TCP protocol requires an acknowledgment to be sent to the sender once the target receives a packet.

Whenever a client makes an HTTP request for data, in this case, the .jpg file described above in Figure 2, that data is broken up into packets sent across the network to the target and then reassembled upon arrival. As mentioned above, TCP uses a send-acknowledge pattern. Once the target receives a data packet, it replies to the sender with an acknowledgment. After the sender receives the acknowledgment, it sends another data packet. If no acknowledgment is received within a certain timespan, for example, half a second, the sender will try to send the packet again or just time out.

the acknowledgment for Packet 2 was never received. Packet 3 is blocked by Packet 2's exchange failure. This is what

head-of-line blocking is about: A data packet can't be sent because a previous one has not been acknowledged. As
nce implications can be considerable.

Subscribe now X

Get the highlights in your inbox every week.

e blocking problem because, as you read above, UDP just sends packets; there is no
data exchange performance improves. However, because UDP doesn't expect an
; can get lost. This is a problem. HTTP/3 addresses the issue by modifying the OSI
ology into the stack, the details of which we describe next.

UIC into the OSI and TCP/IP Models

ıetwork is a complex undertaking. Doing something as simple as viewing a picture of a
es a lot of technology and process. Without a standard way of doing things, the world
ction of proprietary factions fighting it out for market dominance. It's hard to build
internet under such conditions.

Fortunately, the days of the Wild West of network computing have long passed. Today we live in a world of generally accepted models and standards to conduct networking on a worldwide basis. The two prevalent models for internet networking are the [OSI Model] and the [TCP/IP Model].

Both models describe the layers of technology required to construct and operate a non-proprietary worldwide network. The OSI Model has seven layers, while the TCP/IP Model has four layers. At the conceptual level, they are similar. They both describe how to connect two computers on a network to pass data back and forth.

Figure 3 below illustrates the various layers of the OSI and the TCP/IP models. Also, it shows how HTTP/3 maps against the models.

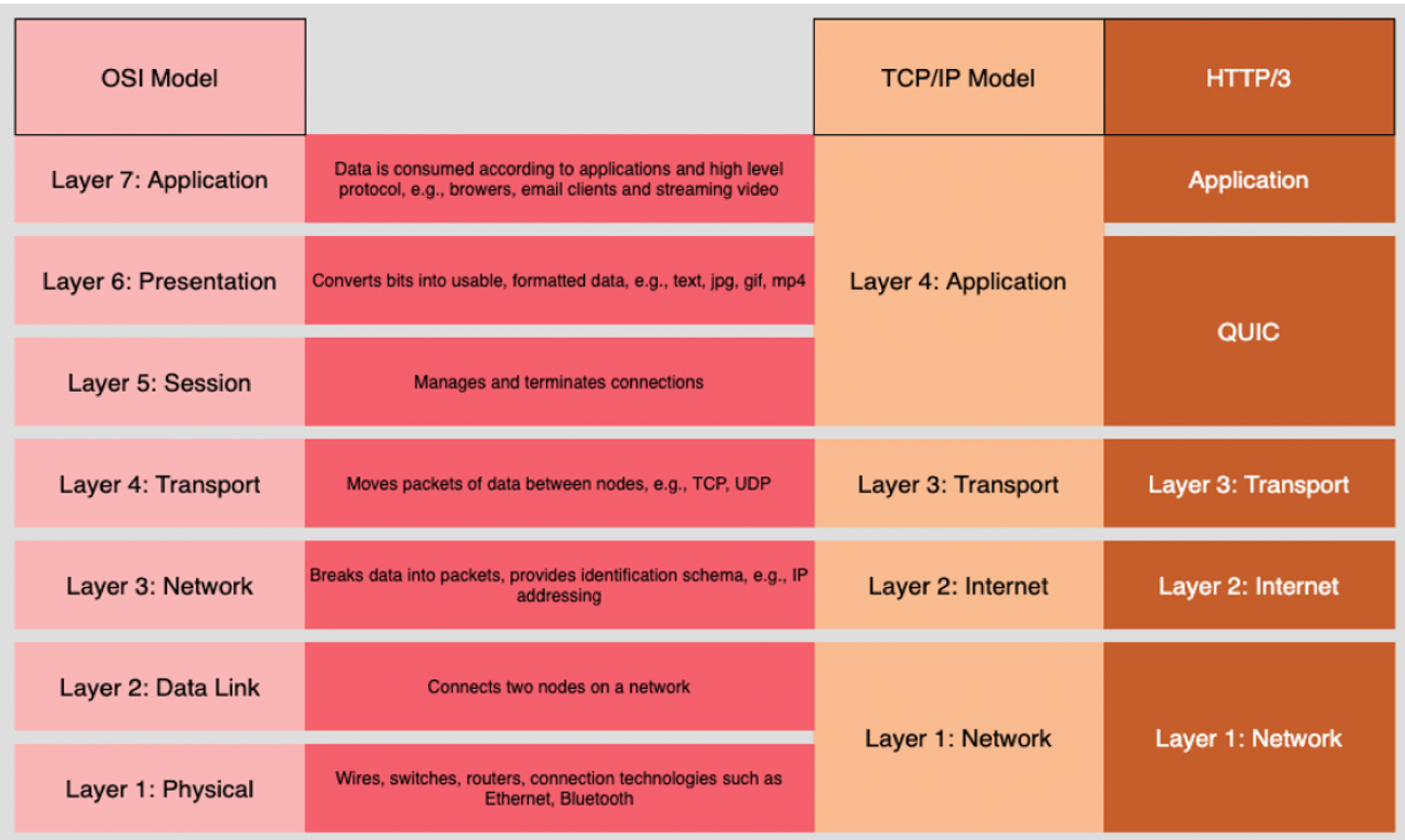


Figure 3: Mapping HTTP/3 and QUIC against the OSI and TCP/IP models.

The important thing to understand about HTTP/3 with regard to the OSI and TCP/IP models is that HTTP/3 injects QUIC into the network layering. The reason for the injection is that QUIC provides fault tolerance for data packet transmission under UDP. QUIC lives between transport and the application layers of the OSI and TCP/IP models. (See acceptable for a few use cases, but most applications depend on reliable packet transmission. While TCP ensures reliable packet transmission, it's slower than UDP. However, when you combine UDP with QUIC, you get the best of both worlds: speed with reliable packet transmission. The result is HTTP/3.

ther

driving force in network architecture for more than a half-century, back to a time ed to mainframe computers miles away. The rise of the internet and the proliferation of d demand.

ake the internet go faster, but it still has a ways to go to become a generally adopted rowsers support it, but it only has a 19% adoption rate among websites as of May 2021. me is adoption on the server-side. Facebook and Google have HTTP/3 enabled on the this writing, Twitter, Uber, Apple, and LinkedIn do not.

How to check if a server supports HTTP/3

You can use tools such as http3check.net or https://geekflare.com/http3-test/ to see if a website supports HTTP/3.

Still, if history has shown us anything, it's that when a protocol exists that will make the Internet go faster, it will be adopted. It's only a matter of time. HTTP/3 is on the path to wide acceptance that will make it an essential part of any Enterprise Architect's technology landscape. Given the impending adoption rate on the horizon, HTTP/3 is a protocol that's worth learning about sooner rather than later.

What to read next

HTTP/2: 5 things every Enterprise Architect needs to know

The second iteration of the HTTP protocol is full of important features for your enterprise architecture. Here are some of the highs and lows of that process, plus 3 facts about the next-generation protocol.

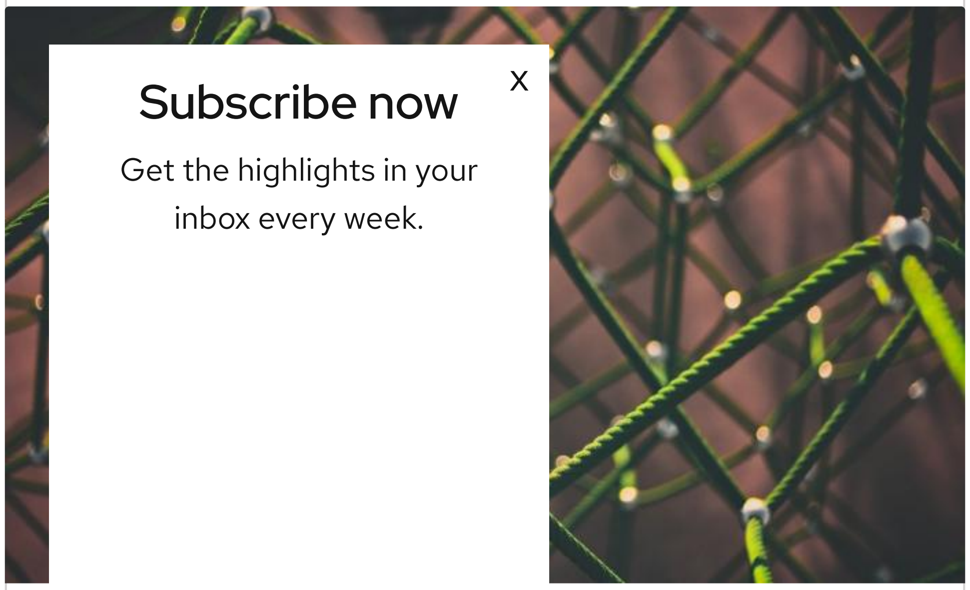
Posted: April 27, 2021
Author: Bob Reselman (Red Hat, Lead Contributor)

Topics: Telecom Infrastructure Messaging patterns

Bob Reselman

Bob Reselman is a nationally known software developer, system architect, industry analyst, and technical writer/journalist. More about me

Related Content



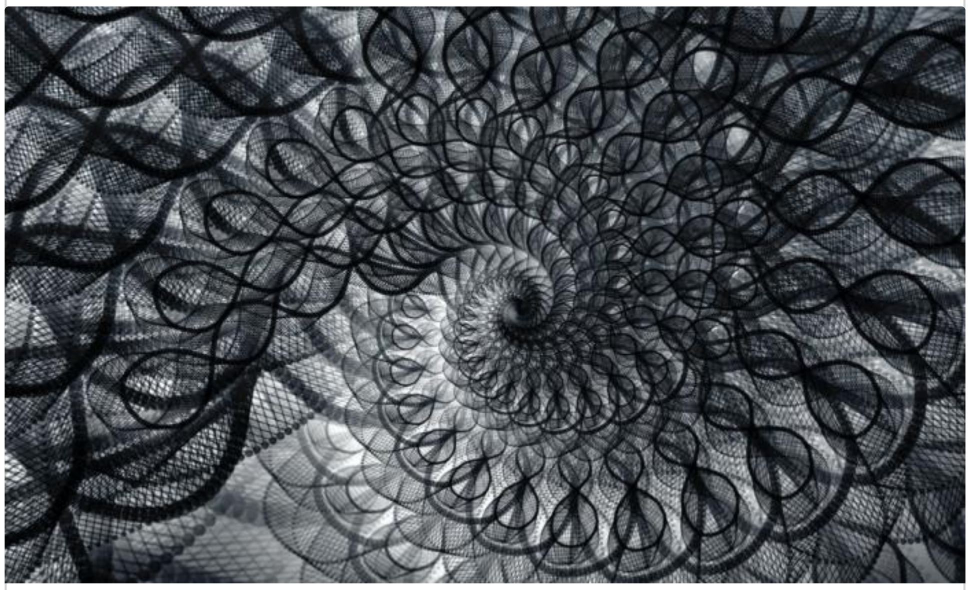
Subscribe now

Get the highlights in your inbox every week.

tects are using

Take inspiration from how Lyft and other companies use gRPC before planning your next implementation.

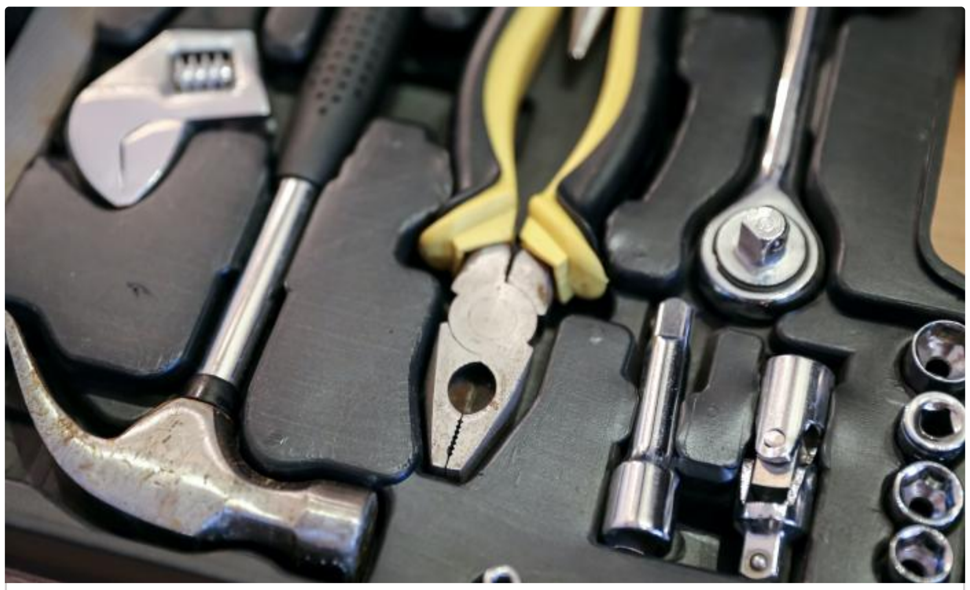
Posted: November 18, 2021
Author: Bob Reselman (Red Hat, Lead Contributor)



Expanding 5G with the 5G Open HyperCore architecture

By using common solutions, open source technologies, and provider-agnostic infrastructure, this architecture hopes to expand 5G availability and adoption.

Posted: October 28, 2021
Authors: Fatih Nar (Red Hat), Ishu Verma (Red Hat)



5 essential tools for managing hybrid cloud infrastructure

It's not just a matter of having the right tools; you must also use them correctly.

Posted: October 19, 2021
Author: Chidambaram Muthupalaniappan (Red Hat)

OUR BEST CONTENT, DELIVERED TO YOUR INBOX

Subscribe now

X

Get the highlights in your inbox every week.

Enter your email address...

Select your country or region

Subscribe

Privacy Statement