Another week of being a Human in IT Infrastructure. Are you still alive to talk about it?

View this email in your browser

IN THIS WEEK'S ISSUE: An SDN Bedtime Story, The Limits Of Packet Capture, & Tech Billionaires Making The World A Weirder Place. Please remember to enable the images; the magazine looks a lot better that way!

PACKETPUSHERS
Human Infrastructure Magazine
A Newsletter About a Life in Networking

Table of Contents
(aka The Project Plan)

Issue Number 37

08/04/2016

The "Thinking about the future" issue.

Thought For The Week:
Is it really August already?

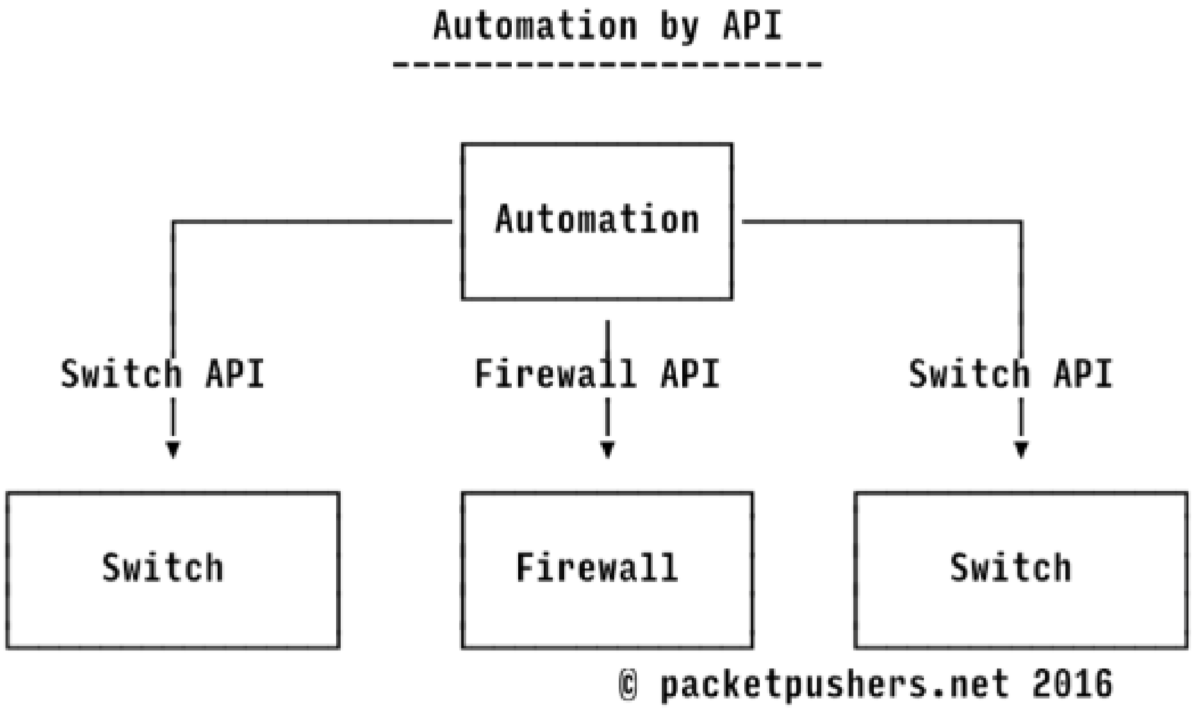
1. An SDN Bedtime Story

by Greg Ferro

So you're starting out on your automation path. The first step you take is to write a script that uses the APIs on your existing devices, because everyone says to begin with automating your existing day-to-day changes. You start with some Python, NetMiko and an IDE to replace the everyday command line tasks to configure VLANs, IP addresses, and firewall rules.

You get your scripts running well. Then the device software is updated and the CLI changes and your scripts fail. Or maybe your script fails because the device CLI is limited to a few commands per second and the error handling is whacked.

So you look into the APIs that latest software versions have implemented, learn about YANG and NetCONF, and you re-write your scripts. This improves things enormously.



As the automation is successful, more and more task are scripted--more devices, more NETCONF, a wider range of tasks.

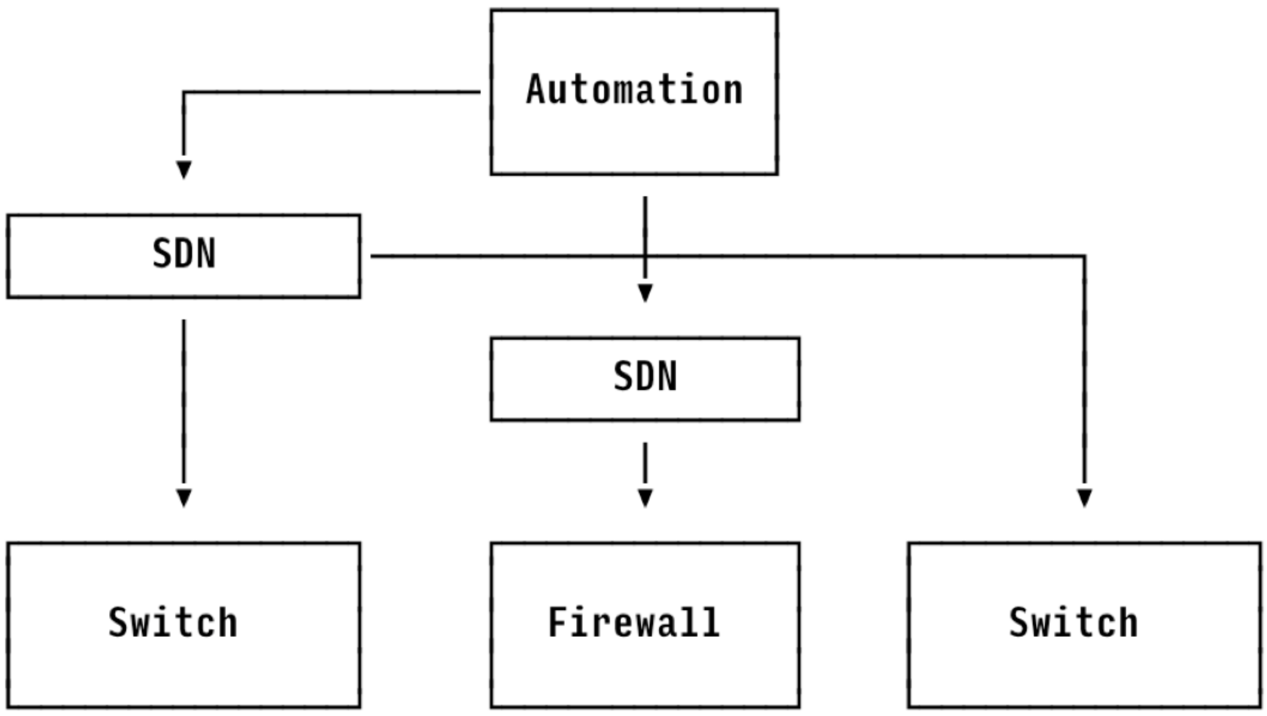
But at some point you realize that you have scaling problem. Managing each device individually isn't working as you spend more time on script interfaces to the devices.

Meanwhile, useful work such as configuring workflows, streamlining log analysis, and improving the Web interface for the help desk falls by the wayside. Frankly, futzing with device configuration scripts isn't going to get you a pay rise:

Boss: "What did you do this quarter?
You: "It was a tough time, I spent 40 days rewriting device scripts."
Boss: "Did you roll out the new features for the help desk?"
You: "Wasn't enough time after solving the problems with the vendors' NETCONF."
Boss: "So nothing got done?"

Enter The SDN Controller

After the pain of missing your bonus has subsided, you start thinking about SDN controllers. You realize that the SDN controller means you don't have to write the device interfaces in your code, you can just write the workflows that will meet the business needs.



Automation by SDN Controller

© packetpushers.net 2016

Just the thing! You deploy OpenDaylight (ODL), install the vendor modules, and start reworking the scripts to use OpenDaylight's NETCONF interface. You know NETCONF already from working with devices, so it doesn’t seem too bad. Then the problems start.

Your firewall vendor (let's call them VendorF) doesn’t have good support for ODL. When the sales grunt turns up to tell you about their SDN controller and mentions a purchase order, you get that sinking feeling. You cynically wonder if VendorF stopped supporting ODL just to sell you their own product (you will be mostly right).

Sure enough, someone on the team falls for the pitch. Maybe it is the visibility/analytics that most SDN controllers have. It might be the improved GUI to replace the one from 1990’s. Or it might be that someone just likes the sales grunt, or wants to bet their career on VendorF.

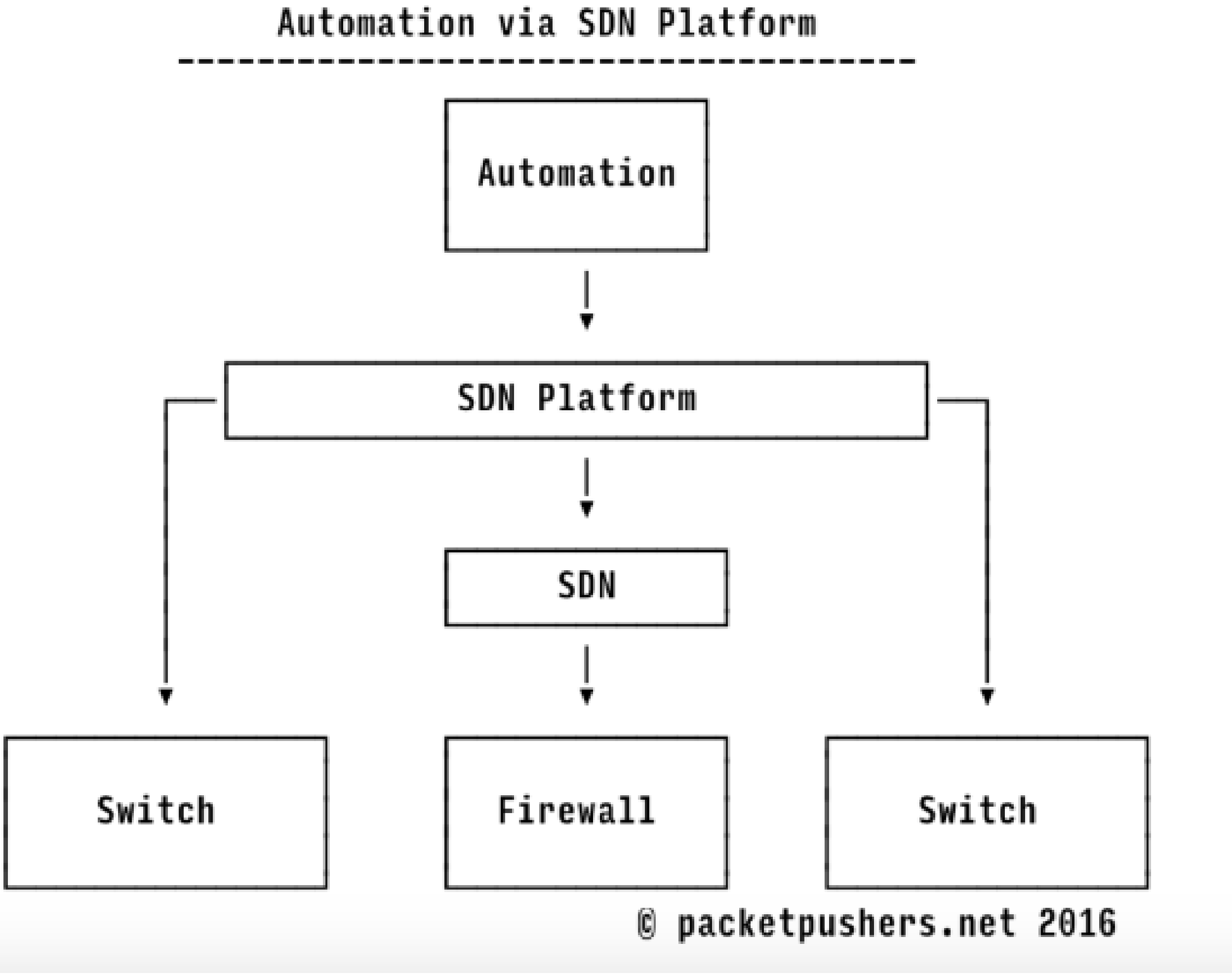
Now you have two SDN Controllers in your architecture.

The SDN Platform

Around about this time, other IT vendors have sniffed out that you have an SDN deployment. “OMG!!!” they are saying, “a customer who is doing new stuff! They must have loads of money!”

The sales siege starts when the big vendors knock at your door with their own SDN platforms. These are SDN controllers that subsume other vendors' devices, controllers, and automation inside of a controllers of controllers.

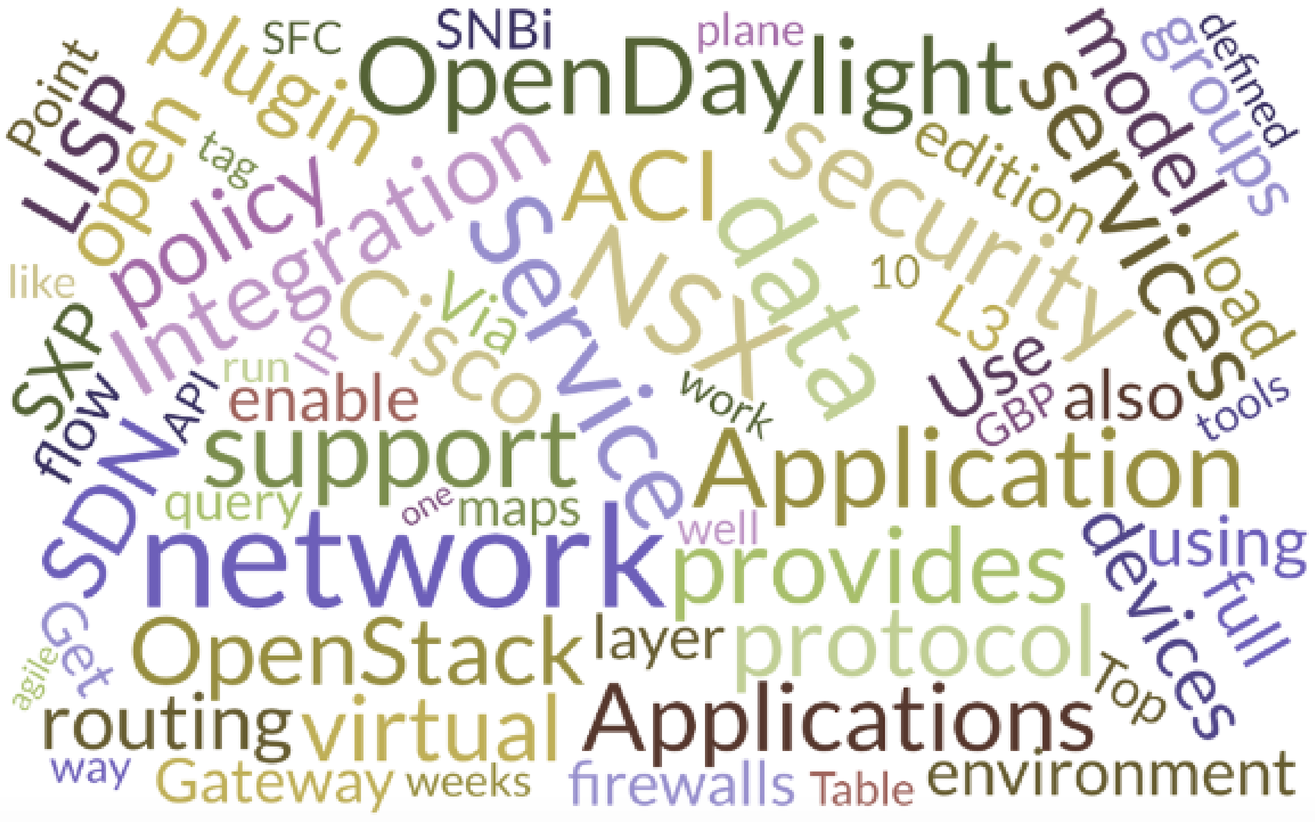
You notice an unusual intensity to these sales pitches. It's because these products are BIG MONEY, and STRATEGIC. More importantly, they have double commissions.



The Final Phase

Now you are in a meeting with your CIO and architecture leads, debating a long list of features. You have a familiar sinking feeling in your gut as they talk about automated application deployment, microservices, end-to-end networking, service-oriented networking, fine-grained traffic control, microsegmentation, visibility, control, performance monitoring, analytics...

OMG...MAKE IT STOP!



And Then

You wake up and realize it was a daymare. You look at your Python code, shrug your shoulders, and go back to figuring out why the CLI on that 15-year-old Ethernet switch isn't working properly.

2. Packet Capture: More Learning, Less Troubleshooting

by Ethan Banks

For networkers, there's no way to get closer to your work than with a packet capture. Yep, if you want to take a look at what's going on down deep in the wires -- to stare right into the network's unblinking eye -- you need a large caffeinated beverage and a big huge monitor to show you all the packet shinies.

The problem with grabbing packets off the wire is that there's too much going on too quickly, and it's hard to distinguish what's important from what's inconsequential.

In a packet capture decoder, all packets look about same. They all have a time stamp. They all have length. They all have fields. They are each likely a part of some end-to-end conversation that you probably caught somewhere in the middle, sort of like dipping a bucket into a fast-flowing river and then analyzing one drop at a time, to see if you can describe the whole river when you're done.

You can make progress if you're diligent at bending the packet decoding tool to your will. But overall, packet captures are a bit over the top.

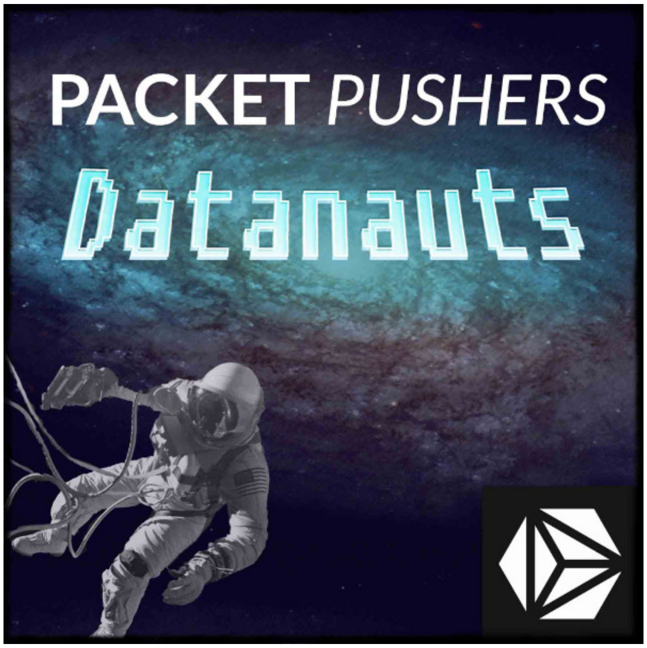
What, then, are packet captures good for? Well, many things...

1. **To see what's really happening on your network.** Everyone who calls themselves a networker should, every now and again, run a random packet capture of a thousand packets or so from a random point on the network, and then walk through the trace. You should be able to explain, at least broadly, what you see. And for the unexplainable packets, you should do some homework. You will learn things you didn't know, and probably didn't even suspect.
2. **To study a specific protocol.** Want to know how TCP really works? Packet captures don't lie. You'll see how TCP three-way handshakes function. The various ways session teardowns happen. How sequence numbers and acknowledgements work together. And how different IP stacks do the same things with subtle changes. The info is all there.
3. **To prove that a packet actually passed a certain point.** A favorite use of mine for packet captures is to prove that a packet got through a suspected drop point. Did the packet make it through the edge router? The perimeter firewall? The DMZ load-balancer? The upper layer of the firewall sandwich? The IPS? The ToR switch? The core switch? The hypervisor host NIC? An unbiased packet capture capable of taking in all

traffic on the wire can prove these things.

4. **To prove that a packet passed through a device changed...or unchanged.** Load balancers, proxy servers, firewalls, and NAT devices all have good reasons for potentially changing a packet. Sometimes those changes affect a network conversation in an unexpected way -- anything from causing a confused recipient to an unhappy firewall.
5. **As a last resort in troubleshooting.** While parsing packets can be informative, in most cases the actual problem is almost always solved at a higher level. I tend to not capture packets until I've had a detailed conversation with the person troubleshooting an application and gone through the likely issues with them. If the answer is in the packet capture, it's usually a very complex problem. Most problems just aren't that ornery.

Packet captures are wonderful things--packets don't lie. But packet captures tend to be more effective as a learning tool than a troubleshooting tool. Which isn't to say you can't troubleshoot with packet captures. Of course, you can. But if Wireshark is the first tool you break out when you get a help desk ticket, you've missed out on engaging the upper, meatier part of the stack.



convergence, data centers, and more.

3. Tech Billionaires Building A Weirder Future

by Drew Conry-Murray

If you asked me as a kid what the world would look like by the time I was middle-aged, I would've guessed hand-held laser weapons, personal flying machines, and human colonies on the moon and Mars, for a start.

Well, here I am at middle age, and so far all we've gotten is the Segway. Seriously?

This is not the future I was looking for.

OK, that's not entirely true. The smartphone is pretty damn cool—and it's essentially the book described in *The Hitchhiker's Guide to the Galaxy*, so a tip of the hat to Douglas Adams. But the smartphone doesn't scream "THIS IS THE FUTURE!" quite like a light saber or a flying car.

While I can only grouse about my disappointment in unrealized progress, a pack of middle-aged billionaires are actually doing something about it.

Jeff Bezos and Elon Musk are pouring billions of dollars into rocket ships and space travel, via Blue Origin and Space X. These guys are tapping old-school sci-fi dreams, the stuff of cheap paperbacks and B movies. Chasing space travel almost seems a little square, like building model train sets, but for really rich dudes.

Let's Get Weird

Other tech titans are aiming for something weirder. Google co-founder Sergey Brin has given hundreds of thousands of dollars to scientists who are growing edible beef from stem cells.

Why? Raising actual cows for food is wildly inefficient: for every 15 grams of beef, you have to feed the animal 100 grams of grass and grains. Why not just grow the muscle and fat tissues *in vitro* instead? Other startups are working on stem cell pork and chicken. Lab-grown meat definiately says "This is the future."

Then there's the realm of human longevity. What good is the future if you aren't around to see it? To that end, billionaire and nascent supervillian Peter Thiel invests in biotech startups aimed at life extension.

One idea that interests Thiel, though he hasn't put any money behind it, is parabiosis, in which the blood of a younger organism is transfused into the body of an older organism. Lab experiments on mice have shown that parabiosis improves heart, muscle and brain functions of older mice who get younger blood.

A startup called Ambrosia is launching a human clinical trial in which people over 35 years of age will get infusions of blood taken from 16-to-25 year olds.

Anyone with $8,000 can participate in trial.

This development smacks of the dark dystopias that populate sci-fi, with rich people harvesting literal human resources from the poor. Perhaps instead of grass-raised beef, someday the wealthy will pay for free-range, organically raised humans.

If blood harvests are too dark, there's always the levity of the flying car, which for me is the canonical marker of a future come to fruition. And yes, somebody's working on it.

Google cofounder Larry Page is backing not one but two flying-car startups, Zee.Areo and Kitty Hawk. Technically, what the startups are working on are all-electric, vertical take-off and landing planes, but they may be small enough to drive on roads.

That's close enough for me. I just hope it doesn't take too long; my future won't last forever.

Internets Of Interest

A collection of pre-loved links that might interest you. "Pre-loved" because I liked them enough to put into this newsletter. It's not *true* love.

By Greg Ferro and Drew Conry-Murray

Dollar Shave Club And The Disruption Of Everything

Ben Thompson, who blogs at Stratechry, has a fascinating post that connects Unilever's billion dollar acquisition of Dollar Shave Club with cloud computing and Amazon Web Services. He explores how the Internet gave a packaged goods startup the leverage to compete with Procter & Gamble, and the disruption this represents for incumbents everywhere.

> *"AWS and Amazon itself, having both normalized e-commerce amongst consumers and incentivized the creation of fulfillment networks, made the creation of standalone e-commerce companies more viable than ever before. This meant that Dollar Shave Club, hosted on AWS servers, could neutralize P&G's distribution advantage: on the Internet, shelf space is unlimited.*"

How to Scale Your Limited Network Staff: Get Rid of the Yaks

Hiring and retaining skilled networking professionals is often a challenge. Lori MacVittie advocates for automation and orchestration of tedious tasks (a.k.a. yak shaving) as a way to help organizations reduce the workload of network professionals, and increase job satisfaction.

> *"By providing the tools to automate and orchestrate much of the yak shaving* that occurs just as frequently in networkers' lives as it does developers', CIOs can scale their valuable (and limited) networking professionals. ... That has the potential to reduce networker churn, which is as costly to IT organizations as it is to any industry in which time and money must be spent training and educating new hires*."

PACKET PUSHERS
WEEKLY SHOW

Where Too Much Networking
Would *NEVER* Be Enough

The Weekly Show channel is our one-hour deep dive on networking technology.

PACKET PUSHERS
Priority Queue

Where Too Much Networking
Would *NEVER* Be Enough

Priority Queue tackles niche and nerdy tech topics and cutting-edge research projects.

The Last Five

EtherealMind.com Latest

Logical Razors Can Take on Corporate Babble

Canned Response to BGP Networking Questions – Reddit

IETF RFC 8374 BGPsec Design Choices and Summary of Supporting Discussions

Net Neutrality Hasn’t Ended, We Don’t Know When

Next Market Transition ? Cheaper Buying, Less Selling

PacketPushers.net - The Last Five

Network Break 182: BGP Hijacked For Cryptocurrency Heist; Juniper, Big Switch Unveil New Products

Show 387: AWS Networking – A View From The Inside

PQ 147: Connecting Security And GDPR Compliance (Sponsored)

Datanauts 131: Masters And Mentorship

Network Break 181: Russia Accused Of Infrastructure Attacks; US Targets ZTE

PACKET PUSHERS

Link Propagation Newsletter

Our weekly newsletter delivering essential headlines, announcements, and useful news to your inbox

Can't get enough newsletters? Check out Link Propagation, our newest publication. We send you a free weekly digest with tech news, interesting blogs, and industry announcements, all curated by the Packet Pushers. It's an easy way to keep up and stay informed. Subscribe at packetpushers.net/link-propagation.

Quick Survey: Vacation Check-In

When you're on vacation, how often do you check in with the office (e-mail, text, Slack, phone calls, etc.)?

A. Total radio silence
B. A few times over the vacation

C. Once or twice a day
D. Three or more times a day
E. Whenever my spouse/family/partner is otherwise occupied

Did We Miss Something?

Got an link or an article to share? Email it to humaninfrastructure@packetpushers.net

The End Bit

Human Infrastructure is bi-weekly newsletter with view, perspectives, and opinions. It is edited and published by Greg Ferro and Drew Conry-Murray from PacketPushers.net. If you'd like to contribute, email Drew at drew.conrymurray@packetpushers.net.

We don't give away your email address or personal details because that would suck.

unsubscribe from this list   update subscription preferences