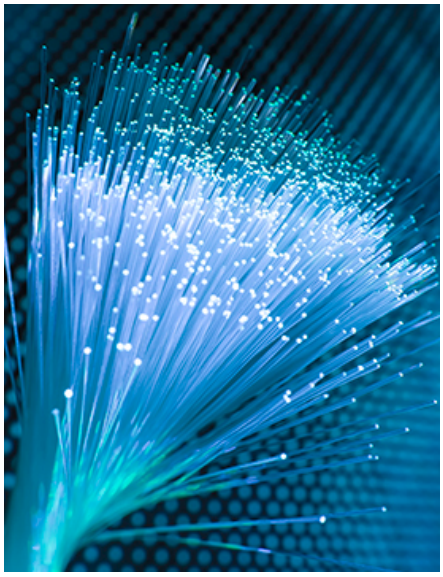
www.pipelinepub.com
Volume 17, Issue 10

Powering Agility with Openness and Automation

By: Prayson Pate



The dictionary defines agility as "the power of moving **quickly** and **easily**." You want to achieve agility in your communications environment. But the question is how.

Let's start with speed

Look at the two adverbs in the definition of agility. The first is **quickly**. Being able to move quickly means breaking out of the old mode of relying on traditional networking solutions. These are provided by a single vendor—and are closed. This means you must go to the supplier and convince them to add new capabilities or rip them out and replace them with another closed solution. This is not a great way to achieve quickness—or to maximize the value of your investments.

Fortunately, we have a good model to follow: the cloud. We've built the cloud as an open multi-vendor system, with best-of-breed software running on standard servers. We can load new applications on demand, giving us innovation with the speed we want.

We call this approach to openness disaggregation. It means we take formerly closed systems (such as a single-vendor computer and proprietary software), define an open and segmented architecture, break up the functionality, and reassemble it with components from different suppliers. And while this may sound complicated, we've been doing it for years.

The first great example was the personal computer. We can now buy the computer hardware from one supplier, buy (or download) the operating system from another, and use applications from a diverse set of authors.

Another great example is the smartphone. With the advent of open smartphone platforms, we've seen an unprecedented wave of innovation, often tied to backend apps in the cloud. Would we have Uber without the smartphone and the cloud? No, because we didn't have the required platform to enable the neccessary agility.

We're now taking the idea of disaggregation from the personal computer, cloud, and smartphone into the world of communications. Let's look at some examples.

Bringing the cloud to the network

Back in 2013, the telecom operators invented network functions virtualization, or NFV. They saw the innovation occurring in the cloud and wanted to achieve that same agility in networking. With NFV we can take a closed network device such as a router or firewall and disaggregate it into hardware and software. We can create an equivalent function by running a best-of-breed application on a standard server, just like with a smartphone. And we're now seeing that approach come to market. With virtualized services, end users can add, change, and stop managed services by going to a portal in real time. Operators can provide value-added services like local hosting of applications for distributed cloud. All this is possible because we can download new functionality on demand.

There are other benefits to NFV. We can select brand A hardware today. But we might not be able to get brand A tomorrow due to trade wars or pandemics. We can then switch to brand B with minimal effort. Likewise, we can quickly change software. Think of the case of a service provider delivering managed firewall services. One customer may insist on a brand X firewall, while another customer wants brand Y. The service provider can now give each of them their

choice of firewall, while maintaining a common platform for hosting—all of which powers agility and brings us some beneficial competition as a bonus!

High-performance hardware

The NFV approach does enable speed for innovation, but it may not be able to meet the high-performance requirements in some parts of the network. Even so, the ideas of openness and disaggregation are still valuable. As before, we can break up a closed networking device into hardware and software. Now the new hardware is a standard switch rather than a server, though. We can pick the generic switch that best meets our needs, and load onto it a best-of-breed network operating system for high-level features like routing, encryption, or management. We can later change the software without changing the hardware, or vice versa. We've opened the networking devices and set the stage for agility.

Fat pipes

The optical network is another area that is ripe for disaggregation. Traditional optical networks have been implemented as closed end-to-end systems delivered by a single supplier. Now we're seeing disaggregation, but it's a little different from the previous cases where we separated the hardware from the software. In the optical domain we're separating the two main building blocks: the terminal equipment and the open line system, or OLS. The terminal equipment includes transponders, muxponders, and wavelengths from devices such as routers. And the OLS includes components such as optical multiplexers and amplifiers. In this disaggregated model, enterprises and operators can buy the OLS from a separate supplier than they select for the terminal equipment. This approach has some of the same advantages as listed above, such as being able to work with multiple suppliers. There is another big advantage, however, that is specific to the optical domain: the rate of change of the terminals versus the OLS.

Optical terminals are evolving rapidly to provide higher connection speeds, better reach, and lower cost. The OLS evolves much more slowly, though, and often has a useful life of ten years or more. With disaggregation, we enable agility in the terminals without requiring changes in the OLS. This is important, because changing the OLS is expensive and affects service.

Agility in software development

We've seen from the examples above that openness and agility are strongly coupled for networking infrastructure. How do we achieve agility in the corresponding software applications?

Once again, the cloud provides a useful model. In fact, one of the essential aspects of cloud-centric development has agility in its name: agile development.

We sometimes refer to traditional development as a waterfall model. This name comes from the appearance of the tasks in a Gantt-style chart. Each task is followed by its dependent task, which is placed below it and to the right. Stepping back a bit we see that the series of tasks looks like a waterfall.

The problem with waterfall development is that we define everything before we do anything. We build everything before we test anything. Then we test everything before we try anything in the field. And we try everything in the field before we show anything to the end user. All of this takes a long time. In the case of telcos, the development cycle could be years.

In contrast, cloud operators move much more quickly. They have adopted a model called agile development. The scope of development is reduced to a few features. These features are scoped, developed, tested, and shown to the customer in a much shorter timeframe: possibly months or even weeks. This approach ensures that everyone is in alignment, and that mistakes in direction are quickly corrected. As a result, we can move quickly and ensure we're going in the right direction.

What about ease?

We have now covered the adverb **quickly** and seen how openness and disaggregation can help us get there. Now let's talk about the other: **easily.**

We've already seen that by embracing a disaggregated model, we have greatly eased the effort of making changes in the network. We can change each layer independently, and we minimize the need to rip and replace.

It's not enough to move quickly, though, if doing so requires an army of developers, testers, and operators to realize the value of an open and disaggregated system. We need a model in which making changes to our network is analogous to loading an app on our smartphone: not something we do every day, but also not something that requires lots of time and effort.

Once again, it's the cloud to the rescue. Along with the idea of agile development, we have some other useful tools with their roots in the cloud. One of these is continuous integration and continuous delivery, or CI/CD. With CI/CD we set up automated machinery to ease the task of developing, testing, and delivering software into production.

Another place where automation can help make things easier is in the realm of management and orchestration, or MANO. With MANO software, we can quickly and easily deploy new software and applications into the network. And we can even turn up new systems in a zero-touch manner, removing the need for pre-configuration or the presence of a technician onsite for installation.

Finally, we can make our lives easier by getting help. Whether they are suppliers, or value-added resellers, or pure integrators, there are partners available. These partners can take a set of best-of-breed components, integrate them, and deliver them as a turnkey system. You get agility with

an open system that supports quick and easy innovation without the effort of having to integrate yourself.

You can also select an open architecture and buy all the components from one supplier. This may sound like sticking with the closed model, but it's not. Because the architecture is open, you are choosing to buy from one supplier, not being forced to. And later you can make changes independently from the supplier—again, because it's your choice.

Shifting how we think

If we want to be more agile, we must change how we do things—and how we think.

Doing the same old thing will yield the same old results. By embracing openness and automation, we can achieve the vision of agility. We can move more quickly and more easily with our planned innovation, as well as in response to unplanned external events. We must also embrace new architectures that are open and disaggregated. We must also change how we work to use shorter cycles and a broader circle of collaborators. Finally, we must change how we think. We must be willing to go faster and make some mistakes, and correct when needed. Together, these changes will enable us to achieve true agility.