Intel

Network Disaggregation & Quality Assurance

CORPORATE PARTICIPANTS

Moderator
Intel

Jason Chandralal
Happiest Minds Technologies – General Manager, Network & Embedded Systems Testing

Laxman Patil
Happiest Minds Technologies – Senior Architect, Network Testing

PRESENTATION

Moderator

Welcome, everyone, to the Intel Network Builders webinar program. Thank you for taking the time to join us today for our presentation titled “Network Disaggregation and Quality Assurance” with Happiest Minds. Happiest Minds is a digital technology services company that is a solution plus partner in the Intel Network Builders program.

Before we get started, I want to point out some of the features of the BrightTALK tool that may improve your experience. There is a Questions tab below your viewer. I encourage our live audience to please ask questions at any time. Our presenters will hold answering them until the end of their presentation. Below ~~Through~~ your viewing screen you will also find an Attachments tab with additional documentation and reference materials, including a number of websites and documents mentioned in this presentation. Finally, at the end of the presentation, please take the time to provide feedback using the Rating tab. We value your thoughts, and we'll use the information to improve our future webinars. Intel Network Builders Webinar Series takes place live twice a month, so check the channel to see what's upcoming and access our growing library of recorded content. In addition to the resources you see here from our partners, you will also find a link to our comprehensive NFV and SDN training program through Intel Network Builders University.

Today, we are pleased to welcome Jason Chandralal and~~,~~ Laxman Patil from Happiest Minds.

Jason Chandralal

Hi, everyone.

Moderator

Thank you for joining us today, Jason and Laxman, and I'll hand it over to you to make the introductions.

Jason Chandralal

Great. Hi, everyone. Good morning and good evening, wherever you are. My name is Jason Chandralal, and I'm the general manager in the networking and systems area. Along with me I have Laxman Patil, who is a senior architect in my group, and we will be jointly walking you through this webinar today on the topic of network disaggregation.

Just a small info about Happiest Minds, our company was started by our current chairman, Mr. Ashok Soota, in the year, August 2011, and since the humble beginnings of a few people, today our employee size has gone to over 3,000 Happiest Minds, as we call our employees, and it’s spread across seven countries. Our company is born digital in terms of 97% of our companies are from digital side, and 94% of our projects are agile driven. We have an active customer base of 173-plus and growing. And just to highlight, among those, we have 46 Fortune 2000 and Forbes 200 billion-dollar corporations. Our company is built on that mission statement, which is happiest people and happiest customers, and it is based on our tenet SMILES, which is expanded as sharing, mindful, being mindful, integrity, learning, excellence, and social responsibilities. Over the last few years, our company has garnered several awards, the most--

one of the highlights being Great Places to Work, which we're ranked 21st, and in the recent survey, and also being placed in the Leaders zone in the ER&D services.

At Happiest Minds, we have a networking industry vertical focused only on data center and service provider technologies. It is grouped under these four heads of programmable and disaggregated system. This is our topic of the webinar today, edge networking, cloud networking, network orchestration, and automation, and the services we provide here, engineering service, our system design, software development, testing and validation, where I come from, analytics and DevOps. Over the last 10 years, we have leveraged and actually participated and contributed to various industry forums, mainly calling out our current and continuous relationship with the Intel Builders forum and most recently last year with the Linux Foundation and the DANOS side.

In today's webinar, I plan to walk you through these five areas. The first one is network disaggregation. We will learn what is the definition, what are the different types of network disaggregations, what are the-- how is network disaggregation being carried out in the service provider network with some examples? Once we do that, we'll identify the key challenges and success factors for deploying a disaggregated network, the importance of quality assurance, and an end-to-end automation approach. We will define and explain the network testing needs with specific use cases that we have learned in our experience, and we would end in this webinar by going through our contribution to the Linux Foundation, DANOS open source forum community.

Network disaggregation. Just to give you a bit of an intro, one of the most happening and interesting topics today in the network industry is network disaggregation, which is often and quite seen as a disruptive technology. But actually, network disaggregation is not really new. The original adopters have been the hyperscale data center providers, and in the last two years, we see a lot of attraction happening on the service provider sense. Now, network disaggregation, or disaggregation in its raw form, means to take something that was previously combined and break it apart into components. So, literally speaking, network disaggregation means separation of the network into its component parts. It won't just mean looking at individual routers and switches, but specifically network device disaggregation. I would go ahead and define network disaggregation as the separation of networking equipment into functional components, and allowing each component to be individually deployed.

What I'm talking here is the ability to source switching hardware and networking operations separately. It's like buying a server from any manufacturer and loading analysis, and this option today exists just because of the proliferation of white-box switching platforms, and the explosion of network operating systems, which are driving down costs and putting innovation – and driving innovation pressure as well against the legacy aggregated systems.

Now, network disaggregation can be applied throughout the network. Using the concepts that we just heard from network – device disaggregation, which is the separation of hardware and software, along with open networking concepts like software-defined networking and network function virtualization, we can now disaggregate the complete network.

Now, what we have in a disaggregated system is actually a white-box that is commodity hardware, right, on Merchant Silicon, and then on top of that, you have an abstraction layer, you have a software environment configurator and tools. And couple that with network function virtualization, sharing shared resources across the multiple layers of the network, introduction of cloud native functionality like microservices and container networking. And then, at the top layer, being the system orchestration and management platform.

When you combine the two, now, what you have in terms of benefits is an open architecture, a vendor-agnostic approach, no vendor locked in here, the ability to choose your hardware, your white-box, or your bare metal solutions from the vendors out there. Scalable networks are functions that help you create on-demand services faster to market solutions.

And last is the virtualization of physical resources, thereby allowing you optimize your network infrastructure.

Now, network disaggregation, as you would know, is across three separate stages, as I would call it. Initially, we had a black box, that was typically the proprietary OEM hardware. We would have, you know, backplane hardware, multiple line cards, and that would be fused with a proprietary operating system and the network functions that work with it. And that was managed by a traditional NMS or an OSS system via CLI.

And with the advent of network disaggregation, now, what you have is something like a gray box. Here, you have multiple disaggregated models, and this is where most of the industry is today, wherein you still have OEM hardware with open software, or you have hardware disaggregation in terms of white-box and bare metal solutions fabric switches, and still running proprietary software. And this is managed or exposed via open APIs or native APIs to the SDN controller or your orchestration system.

The final stage, and that is where network disaggregation is headed in the last two years – over the last two years is a complete white-box system. Here, you have open hardware, that is a commodity hardware running Merchant Silicon that you can choose from the variety of vendors out there. That is connected via a hardware abstraction layer and a software SDK, primarily, to install the NOS that you choose. On the software side, you have open software solutions out there that you can try. I have mentioned DANOS, you have SONiC and so on, and the ability to choose and get that integrated, and that entire system is actually open via open APIs. These are connected to your controller management system.

Now, the solution that you see in the white-box is what we are looking at today in terms of all the benefits that I have explained, primarily in terms of your OpEx and CapEx reduction, and the ability to spawn multiple services in the form of microservices.

Now, in terms of network disaggregation types, there's primarily four types. I'll just touch upon each type to give you an idea of what I'm talking about.

The first one is the Chassis to Fabric Disaggregation. This type of network disaggregation replaces a single chassis device with a multiple fixed configuration device. Here, a network operator might undertake this type of disaggregation to create a fixed delay switching path through a fabric, and to enable the user for single type of device across the network fabric. What this typically means in an environment, for example, is you could have – where microservices are deployed – you could have H2H delay in a network that is additive. And you could use white-box solutions to actually, you know, simplify your configuration, and give you the scale that is required.

The second type of disaggregation is Software from Hardware. In this type of disaggregation, software is separated from the hardware, wherein a network here, a network operator acquires hardware separately from the software. In addition, he is able to manage the lifecycles both from a software and hardware perspective separately.

The third type of disaggregation is Function and Appliance Disaggregation. This form of disaggregation means taking a common appliance, for example, a security gateway or a firewall, and a load balancer and placing them in a software-only platform. For example, a firewall can be separated into three distinct components, like it's network address translation, stateful, packet inspection, and deep packet inspection service. Now, these services can be tied together with a concept called Service Chaining to create scalable services that can be created, inserted into a traffic flow and scale as needed. The trade-off here is in this kind of disaggregation, the abstraction of the hardware platform, which ultimately reduces the performance of the services is something that you would experience.

The last type of disaggregation is Control Plane from Appliance Disaggregation, and this is the most common type or instance of network disaggregation, and is the original version behind software-defined networking, especially when OpenFlow was introduced. The primary advantage of disaggregation is a reduction of the individual network complexity, and the ability to centralize policy calculation and [inaudible].

So, just to quickly summarize, disaggregation is not one big thing, but rather a group of methods and tools, each having its own positive and negative trade-offs.

Now, how does it all pan out from a network disaggregation perspective. And just to tell you, in terms of how the network is structured, from a high level, you have the core, edge, access, and transport layers today.

So, at the core, we already see that disaggregation has, you know, has already happened. Last year, towards last year, AT&T has already deployed its first next generation disaggregated core out of the platform into its production network. They followed a DDC design, which is actually a disaggregated chassis design along using Broadcom Jericho2 family and they used DriveNets' network cloud solution

along with its converged network operating system, which actually meant that they could actually support all the sophisticated features that are required in the core of the network like MPLS.

On the edge side, we see a lot of traction, a lot of initiatives being happened. This is driven by the OpenEdge Industry Forum and covering both open edge networks and edge computing at that layer. And just to be sure, edge has two distinct parts, and I'm talking about edge network, which is actually the service oriented network part of the network, and edge computing is a complement of the same.

At the access layer, we have seen quite a bit of adoption early with network function virtualization, as you see depicted by the vOLT, vBNGs, and most recently, the disaggregated cell gateways that have been using disaggregation techniques in the runup to the rollout of IT services.

And the transport layer is a further area that has seen, or is starting to see adoption of white-box and disaggregation on the ROADM and DWDM in the transport network.

So, how does it all span out today? Yyou have to take legacy network infrastructure being already – that is already available today, that was actually purpose-built hardware, monolithic systems that had a role, typically with the backplane line cards, and they would provide a fixed service. And that created boundaries in the network. So, you would have access devices connected to the metro, which is aggregated, the traffic is aggregated and that would get into the edge and then into the provider core.

Now, with the advent of disaggregation methods and tools, the disaggregated network cloud would actually see that boundary between access and metro tend out, which is a flagged service oriented architecture, which has access up to the network entry point, be it a fixed or a wireless service. And now, you have commodity hardware that can be horizontally scaled and can run each of those services that you have in the edge and the core, running as a microservices in its infrastructure today.

Coming to the challenges. There are some key challenges to network disaggregation. The first being complexity. Now, we have a lot of white-box hardware, fabric switches, and an option to load any kind of NOS. Testing these disaggregated elements manually is a challenge.

The second is the performance. You need to... performance is, in terms of the OEM solution or versus a disaggregated solution, you need to come up with network KPIs and testing methodologies to test ACU type of workloads, microservices.

Third is the coexistence and compatibility with existing OEM hardware/software solutions along with disaggregated elements at the network, providing the need – the challenge to actually test both traditional APIs and also, you know, the new type of REST NetConf based services.

Interoperability. As we all know, in a service provider network, interoperability is one of the key challenges and that is further compounded by the need to test different elements in the network.

The fifth challenge is in terms of the non-standardized test tools, software, open standards that are not there today. But however, I see cross-industry collaboration happening today to address these challenges, and they're moving in the right direction, although this is a challenge that we need to look out for.

And lastly, the limited availability of skilled manpower, and having networking test automation and NetOps skills.

Moving onto the success factors. These are the key success factors, as I view it, in terms of deploying and disaggregating network from a testing point of view. The first thing is to define strategy and approach. A well designed strategy is required, which is a move away from the traditional type of testing. A lot of manual testing will not work out here, and we need a new type of approach.

Framework and tools, this is across all the disaggregation models that we just spoke about. You need to define, identify, and try using these tools and frameworks in advance to deploying your networks. Benchmarking performance, that is OEM based services versus a white-box or open software based KPIs.

Continuous testing. You know, test automation and DevOps is another key aspect or success factor that is required to run continuous builds, continuous workloads.

And last is the solution component that is completely testing a disaggregated solution is something that we need to look at.

Having put these challenges and success factors together, just coming out with a proposed test framework to test a complete disaggregated system. The left part of the screen is what you see is a device under test, or a system under test as I call it, which is a completely – which is a complete disaggregated system.

It has four layers out there. Just to highlight the layers, that is the hardware layer or the COTS layer, were we have an x86 system, Merchant Silicon sitting on a white-box. A hardware abstraction layer that gets connected to the independent NOS. And the third layer is your control plane functionality, here, where you have your routing protocols like OSPF/BGP, hardware services into the applied services-led MPLS. And then you have the management plane that supports both your traditional CLI, and also, your REST and NetConf based services.

Now, the central part is the test orchestration system. It is following a multi-layered design and it focuses on three key aspects. That is the test engine by itself, having the test data, the test repository, the test files, the configuration engines that are integrated both within CI/CD, and also talks to the primary interfaces, be it CLI, or REST.

The second layer of the automation framework is the network infrastructure preparation or the automation that is to create these infrastructure components together, you know, via orchestrating software. It could be open stack, it could be open source manner, it could be tools like Kubernetes to orchestrate your containers or bare metal tools.

And the last layer is your performance testing layer, which is actually used and has a combination of emulators and simulators, and that is required to run your complete end-to-end system.

I would like to hear back from you on what you think could be some areas that we could add in terms of having an end-to-end system.

Having identified the challenges, the key success factors, and then now an end-to-end framework, the… it's very clear to see that quality assurance takes a primary seat when you're planning a disaggregated network. And to speak more about it in-depth, I would like to hand it over to my colleague, Laxman, who will walk you through the QA approach.

Laxman Patil

Thank you, Jason. Hello, good morning everyone.

The quality assurance we are proposing is based on three fundamental blocks, the new test process that we plan to show in terms of how we will cover the different parts and different types of the network disaggregation. The test automation to ensure different network operating system running on various types of underlying hardware with multi-layered design and customized networking domains. We have also ensured the framework is based on reusable components like keyword driven approach. Last but not the least, QA also includes and not limited of adapting and learning new test methodologies in terms of KPI benchmark testing that are being done in the open source forum, which should result in what we call as "end-to-end test approach".

Proposed quality approach covers all the aspects and the areas of the network. On the left side is vertical approach, and on the right side is the horizontal approach. The vertical approach aligns to the network cloud, where we have the different blocks or the components of disaggregated network cloud. The right side, as we spoke in the past, is at the functional area across the traditional networking infrastructure.

To begin with, vertical approach describes what needs to be tested and validated at each block. The base OS network operating system should be tested against different hardware entities like termination or restart of processes, software updates, and patching updates, crash simulations, which we usually see in these blocks. On the cloud native component, testing to validate support of types of

interfaces like SR-IOV, vOLT I/O, different encapsulation, and tunnel such as VLAN, VxLAN, GRE, and the support for both IPv4 and VPNv4 routes.

And last but not the least, on the protocol stack, we need to verify the L2/L3 protocols conforming to both the open and industry standards, applications such as security features, role and policy based access controls and all.

Now, moving onto the right side of the area, testing of access network should include all types of networks based on ethernet, IP, new generation FTTx based services. The approach should take a provisioning of all these services under one single platform instead of managing from different platforms. The application or edge layer, what we see is one point where density of features is more compared to any other network. Also, it is multiple services with wide range of features, including and not limited to BRAS, MPLS, carrier ethernet, mobility and VPN services gets merged, aggregated, and distributed to and from access networks and sent over to core networks.

In core networks, although there is not much of features overload as compared to the edge, focus should be retained in validating performance and stability, and scalability as it demands in the core network. Switching and routing of traffic is based on MPLS label lookup or through routing using IP services. The main focus in this layer is how quickly a router can send and receive traffic at full wire speed.

And finally, to the last layer, which is the transport. Transport network is completely served by vertical integrated solutions provided by OEM vendors. Implementation and development over the records are driven by operators. Selecting a vendor with all optic elements such as transponders, ROADMs, multiplexers being coded by the CM vendor. With the adoption of disaggregation at optical networks gaining traction, testing should be focused on newer challenges of disaggregated optical networks in validating interface and APIs of terminal devices. SDN is the key technology and ability in making optical disaggregation feasible. Testing should also take care of the northbound and the southbound interfaces used for configuration and operations management, alarms and fault tolerance cases via SDN controllers.

The next phase of the test interoperability and the portability testing, testing of any software on top of any hardware is an important aspect to make sure disaggregated elements are working together as one simple entity or a system. Validation of different network operating systems from various vendors on different hardwares like white-box, bare metal switches, and virtualized environment, that is cloud or hypervisor-based VMs, the hardware needs to be verified from both x86 and non-x86 platforms to quantify all aspects of testing, such that there are no issues when it comes to interoperability and portability.

And the last part of our quality assurance is the test automation framework where the need is to cut the manual intervention of test execution, resulting in a reduction of testing cycles. The framework should be able to configure/manage disaggregated as well as the legacy networks. They should be able to configure open config protocols like REST, NetConf, and command line interface.

And one important aspect of the framework, which I would like to touch is it should be able to handle all types of testing, such as features, sanity, performance, and aggregation. It should also have support for open source based CI/CD services, which would further help in reducing development and porting efforts, indication with open source based traffic generators with different traffic profiles, and patterns as demanded for next generation networks. Care should also be taken to run or execute the test suites in serial or parallel on testbed requirements.

Moving onto another important aspect in case of disaggregated networks is the use cases which we see in today's networks. Disaggregated systems based across different network infrastructures based on the functionalities and applications as compared to traditional vendor boxes. But these new systems define tickets over hierarchical construct of networks simpler and dynamic flat service oriented infrastructure to cloud, and virtualization based disaggregated infrastructure.

If you see from the slide, starting from the left, the devices which we see in access layers have come to be known as universal CPE or the virtual CPE, which serves to end customers. These devices are connected to the next layer, which is an aggregation or edge layer. There

can be a couple of other devices in between access and edge, like multiplexers for BRAS services, or a signal splitter in case of GPON based services.

In the case of aggregation or edge, all the access links terminating on this layer are aggregated or merged. Policy is applied based on the services adopted by the end customer. Say, for example, a BRAS services using end users will be based on the virtual templates, or in case of MPLS based VPN services, each customer is placed in a unique VRF routing table, unless you are on a specific VRF interface. Also, before sending to the core network, customer traffic is marked and prioritized using the QA policies.

From the core perspective. Core network generally consists of high end routing systems, where the main functionality is to switch or route the traffic with minimal or least time. And on the data center, the general content services running from small, medium, or large enterprise market with servers running behind firewall and load balancers based on the applications and network functionality. These data centers, in turn, have high availability connectivity to more than one ISP for redundancy purpose.

And finally, the transport networks contain a host of optical elements such as multiplexer/demultiplexers, ROADMs, DWDM, optical systems and switches. Transmission links are made up of optical fibers and transmission of signals use different methodologies such as WDMs, DWDM, OTN and so on.

Now, we will focus on areas each and what needs to be tested in these specific layers. Focus should be to test functional and non-functional aspects in all these areas. Each of these test areas require testing and validation of boxes against functional and non-functional performance, scalability, characteristics before qualifying as a production-ready node.

So, from the test area CPU perspective, at this layer, testing needs to focus based on the type of service the access device is going to serve. For a uCPE device, testing needs to verify industry standard routing protocols, VPN, and security features. And in the case of FTTx based service, focus should be on the ONU and OLT based features. On a performance or a scalability testing, elements need to be quantified based on the RFC-2544 for standard bare metal switch, and RFC-8172 in case of virtualized VMs.

The telco edge is where service providers can process data in near real-time, gain substantial operational efficiencies and grow revenues. From an enterprise service perspective, businesses have moved from on-premise to cloud or hybrid IT models, with the ability to dynamically scale services, plus pay only for what they use kind of model. The disaggregated box should confirm all the basic tenets as compared to the legacy devices from a functionality perspective. For example, testing should validate the maximum of number of IBGP or the EBGP for MPLS PE router, the number of PS sessions for IP networks towards other ISPs, broadband subscribers. A BRAS router can handle at full throttle.

And test areas of core mainly should focus on functionalities like nonstop routing support for control and data play, full availability and switch off functionalities, or full mesh or any mesh kind of topologies. For example, test should cover when a single or multiple processes within an OS crashes, it should not bring down the entire network with high flow traffic.

From a data center perspective, the focus should be more on VM mobility based tests, something like north-south or the east-west on the VM location. The VMs launched in the same rack or a different rack. Test should also include data center interconnectivity between two different data centers in the same location or across geographical locations.

It should also take care of high availability of services in the case of disaster management, where services should be up and running in no time. Another important aspect is to verify different encapsulation types, which are used for VM-to-VM communication.

And finally, the last part is the transport network for functionality point of view. Main focus areas for testing disaggregated transport elements are open source based optical network operating system for control and configuration of disaggregated DWDM. Open or common data models, not found on solid point APIs and interfaces. From the non-functional perspective, tests should include network failures, platform failures, configuration rollback, network state and leases, and monitoring through alarms, which helps in resolving the network issues with quick turnaround time.

Now, with that, we have seen by adopting a new and different QA approach can be used to test each part of disaggregated system and network. There are further innovations happening on the access side as well. To talk more about this, I'm handing it over back to Jason.

Jason Chandralal

Yes, thanks, Laxman. So, we started our journey, you know, with our Linux Foundation project around last year May, when the first source code for the disaggregated vRouter was open source. Happiest Minds' contribution is actually in the area of test and validation, specifically covering three areas. The first one is configuration guides that are based on the features that are supported by the vRouter code. The second one is in terms of – we have created a test automation framework, and we have upstreamed that to LF-DANOS foundation. The details of which you can see on the screen now. And the third aspect is the test cases and the test scripts that are associated with those test cases that are also being written and upstreamed to the community.

In specific, the functional areas that we looked at is OSPF, BGP, MPLS security features, like NAT and firewall, and you know, IPsec services in terms of VPN.

We did do some level of traffic testing, having different types of traffic profiles, and different, say, frame sizes and protocols.

The tech stack that we used in terms of the framework is the following an entity approach. That is the ROBOT FRAMEWORK, Python libraries, Python3, integrated with CI/CD pipeline and having a lot of open source that we have learned over the years, like iPerf for performance testing and Scapy and similar areas.

In terms of the framework, just to give you a quick brief. The framework actually has these four layers, as I will call it. It's the framework core engine. That's where all the processing, the execution, the correlation, and the results are published. It uses the next layer, which is the test repository, that has the actual test cases, test scripts, with the test config and test data required for those scripts to run. It has a validation... it has a config tool layer, which actually passes the configs that need to be run on the DriveNets vRouter, and a validator to ensure that the configs are right. It does have plugins in terms of traditional CLI, REST, and NetConf. We typically use mostly REST and NetConf, but actually this outbound function in terms of, you know, setting the configurations and really the features. And these are retained in Python libraries.

You can look this up, our contributions in the DANOS website. You can just go to GitHub DANOS as you see the link before, and you can just have a look at what we have done.

This is an area in evolution, what it means that you are free to download the code, have a look at it, make modifications, and share your experiences and best practices back to the foundation, which will be very beneficial to actually come up with a better framework for testing a DANOS vRouter.

Coming to the last slide of my webinar today, I would like to actually leave you all with a couple of viewpoints here. The first one is network disaggregation is here to stay. It is indeed transforming the network and telecommunication infrastructure by enabling agility, common platforms, and significantly lower TCO. Standard bodies, cross-industry collaboration, open source communities, vendors, and service providers, together, are actively working on network disaggregation and thus driving innovation, resolving technical challenges on various, and trying to set standards. This is all possible, as I spoke earlier, due to the availability of powerful white-box solutions in conjunction with superior open source network operating systems that have enabled common platforms that are actually seen reshaping the network service provider infrastructure, especially at a time when 5G rollouts are happening.

The second point is in terms of testing. Testing is very crucial to ensuring the success of a disaggregated network deployment. With the constant transformation of the network and associated technology or technical complexities faced by network engineers, verification has become a very important step for enabling a disaggregated network infrastructure. Therefore, your focus, I would suggest, would be on a well-defined QA framework or an end-to-end strategy which provides significant benefits related to early detection of faults, improved test cycles to test automation, predictability of outcomes, enhanced flexibility and scalability as new technologies are added. And finally, adequate testing to ensure a smooth deployment.

With this, I come to the end and thank you all. Over to you,-Lilian~~Lidia~~.

Moderator

Thank you. Thank you both for the great presentation. We do have a few questions that have come in while you were presenting, so let's get started on the questions.

Our first question here is, "What do you think are the business drivers for telco edge disaggregation, and what would be the implications on the technology side to meet these expectations?"

Jason Chandralal

OK, I'll take that. There are actually four to five key business drivers driving the need for telco edge disaggregation. The first one is new services and cloud workloads which depend on low latency. So, those will be streaming services, conferencing, gaming, IoT as we have it now, and 5G. The second would be highly reliable services. This would be targeted towards the financial services, remote access as we see during the pandemic, and emergency services. The third one is faster time to market. This would be instant service availability and on-demand based on the customer's wishes, or their economic needs. The fourth is a low cost model, that is reduced cost of services that is required today by a lot of services providers. The fifth one is a collaboration that I see between CSPs and hyperscalers today. And last, is a unified policy, you could have access from your office in terms of fiber, you could use a Wi-Fi at home and then have a wireless access as well, and all of that needs to have a unified policy.

In terms of the second part of the question, to the technology implications, I would say innovation driving these areas, and therefore, you have cloud native technologies coming out in terms of microservices, open networking, disaggregated methods, you know, scalable network operations based on cloud. Virtualization of physical services – physical resources allowing better sharing, optimization of those resources.

And finally, a transition from a VNF to a CNF, which is to container network functions.

Moderator

That's great. Thank you. A second question we have here, a member from the live audience said, "You outlined the challenges of network disaggregation earlier, ~~but~~ building from that, what are some of the associated risks and the best ways to mitigate them?"

Laxman Patil

That's an interesting question. ~~so~~So, disaggregated networking is not without risks. I would like to break it into a two-part answer. The first one is compatibility issue and the second is integration and the testing activities.

Coming to the first one. Disaggregation provides flexibility, a reduction in OpEx and CapEx, but also it comes with complexity. Although it will become more and more difficult to test disaggregated devices and networks manually, change in QA approach and adoption of newer and open source based automation frameworks which is agnostic to underlying elements of disaggregated networks will help in evaluating interoperability and compatibility issues.

The second part of the question, like data integration and the testing activities, network disaggregation involves integrating and running the networks plus systems, as a whole. Specific testing needs are required to test and integrate the software, the new hardware, software upgrades and patching updates, and so on. A flexible network should test and validate software running on networking devices, irrespective of whether it's disaggregated or legacy devices.

To overcome delays in network and service rollouts, automation framework should have continuous evolution with open source based solutions and helping addressing sanity and the regression cycle times.

Moderator

Thank you. Thank you for the answer. We have a third question here. “It’s interesting that you mentioned the role of QA in network disaggregation, what is the upside in choosing the right QA strategy?”

Laxman Patil

Yes, let me take that question. So, networking industry, especially telcos, small, medium, large enterprise systems are getting the benefits of reduced cost with respect to software, hardware, and open source software packages. What lacks are the tools based on open standards to test disaggregated devices in terms of features, scalability, and performance.

A complete disaggregated QA approach should satisfy major requirements for testing, including and not limited to as the first point, open source based automation framework, layered approach for multi-networking domain needs that is vertical and horizontal, what we discussed earlier, testing for – based on vendor-agnostic open config and native and unstructured CLI based testing. And finally, and not the last, a reduced final manual and increase in automated sectors for quick time to market. Thanks.

Moderator

Sure, thank you. We do have a last question here from the audience. “Can you outline some of the key requirements of an end-to-end test framework in ensuring the success of network disaggregation?”

Jason Chandralal

Yes, I’ll take that. I can think of many such requirements that are required for an end-to-end framework and just highlight the key ones. The first one would be an open source based framework. This will ensure adoption, familiarity with the network engineers and the network developers. The second one would be to have an agnostic approach to the OS, to the chip, to the technology and the testbed, be it virtual or real. The third one would be recommended multi-layer design, as we spoke within the earlier part of our webinar, covering all the aspects of a disaggregated system. The framework should be customized for the network domain. It should be, you know, extensible in terms of ensuring that all the protocols that we know exist in the service provider network are, you know, are factored in.

The framework should be… this framework should be a single framework in terms of running functional, non-functional, you know, sanity type of testing, so that you have one framework in your test strategy that does all, rather than going with a combination of multiple frameworks. This framework should be scalable and extensible to add the expected inflow of new protocols and features.

The second last one would be having an integrated or a plugin to integration in terms of CI/CD, so that we can run continuous testing against the various builds and workloads that are presenting itself.

And last would be – the framework should enable both serial and parallel execution so that you can run multiple test suites on multiple testbeds at the same time, thereby reducing your cycle time.

Moderator

All right, thank you. Thank you, Jason, and Laxman for the great presentation. Thank you all for joining us today. Please do not forget to give our team a rating for the live recording, and this concludes our webcast today. Thank you everyone and bye for now.

Laxman Patil

Thanks everyone, bye.