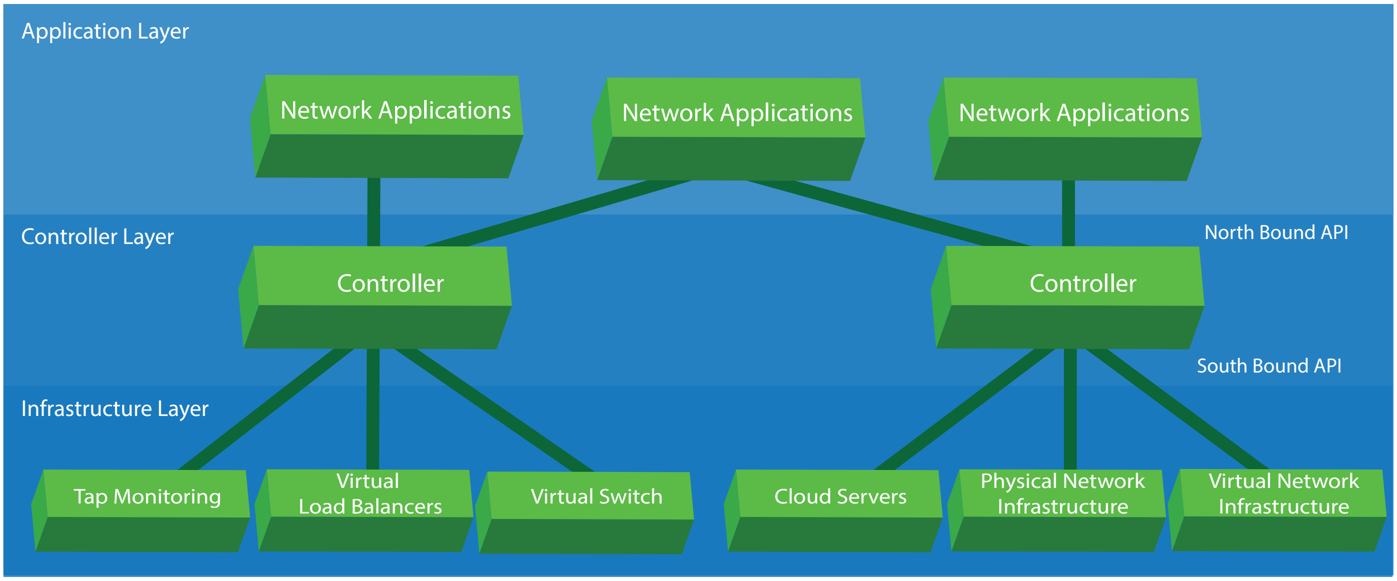
Software Defined Networking:
Should Do Now? Should Do Never? Simply Don't Know!

By Jeff Roper, Chief Technology Officer, Entuity

Should an organization embrace Software Defined Networking (SDN) now, soon, or never? In this paper, Jeff Roper, Chief Technology Officer for Entuity, provides an introduction to SDN—it's background, implementations, pros and cons, deployment considerations and more—to help organizations determine when, or whether, SDN is right for them.



The software defined networking architecture separates the control plane from the data plane. In the Controller Layer, centralized controllers use APIs to link applications in the Application Layer with elements in the Infrastructure Layer, such as virtual switches, cloud servers, and physical and virtual networks.

Background

The primary goal of SDN is the separation of the data plane from the control plane and centralization of the control plane intelligence enabling holistic network routing decisions to be made. The data plane is then free to pass packets in an efficient per-hop manner proscribed by the control plane using virtual overlay (or underlay) networks.

Virtual networks are not new and have existed in many forms over many years. MPLS, VPNs, ATM, Frame Relay, and VLANs, are a few examples. SDN aims to decouple the overlay from specific hardware technologies and to provide pan-network overlay controllers, which perform provisioning, thus significantly increasing network agility. These controllers are able to dynamically reconfigure network paths to avoid congestion, implement new services, add in-path virtual infrastructure, and so on.

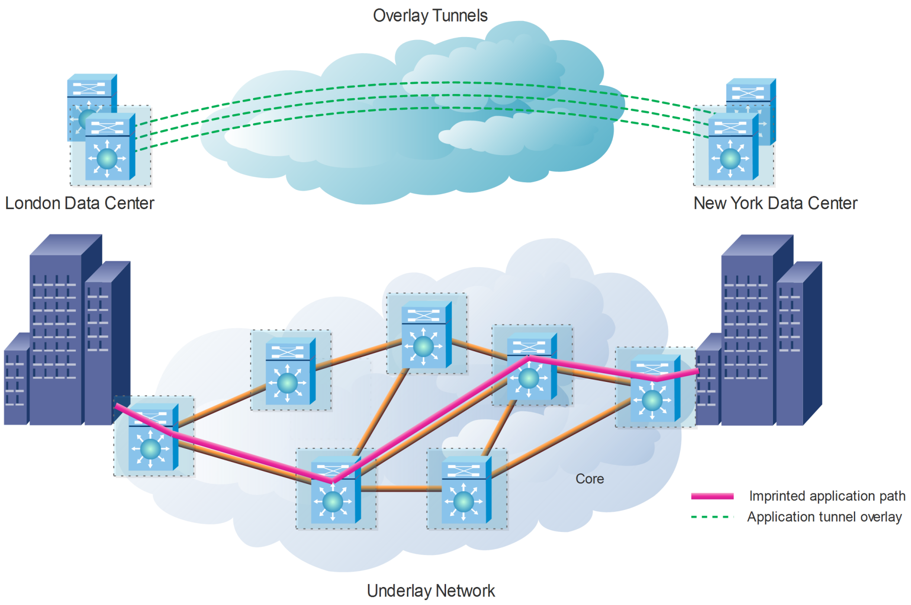
Network Function Virtualization (NFV) and SDN are related, though it is possible to implement either one in isolation or both together. NFV has been evolving for some time and most enterprises have deployed some form of virtualized switches, such as load balancers, WAN accelerators, and firewalls. The virtualization of such network functions enables rapid virtual appliance instantiation and relocation for optimal placement and/or appliance distribution, for example, pushing firewall capability out throughout the network rather than simply occurring at fixed physical (or virtual) network locations.

Overlay SDN or Underlay SDN?

SDN technologies are broadly split into two fundamentally different paradigms: "overlay" SDN and "underlay" SDN. With overlay SDN the SDN is implemented on top of an existing physical network. With underlay SDN the fabric of the underlying network is reconfigured to provide the paths required to provide the inter-endpoint SDN connectivity.

Overlay SDN (e.g., VMware NSX and Contrail) use tunneling technologies such as VXLAN, STT and GRE to create endpoints within the hypervisor's virtual switches and rely on the existing network fabric to transport the encapsulated packets to the relevant endpoints using existing routing and switching protocols. One advantage of using encapsulation is that only the tunneling protocol end-point IP addresses (TPEP IPs) are visible in the core network. The IP addresses of the intercommunicating VMs are not exposed (of course the downside of this is that without specific VXLAN awareness, traffic sniffers, flow analyzers, etc., can only report on TPEP IP-IP conversations and not inter-VM flows). Another advantage of encapsulated overlay networks is that there is no need for tenant segregation within the core (e.g., using MPLS VPNs, 802.1q VLANs, VRFs, etc.) as segregation is implicitly enforced by the tunneling protocol and the TPEPs.

One of the major drawbacks with overlay SDN (such as NSX) is that there is little, if any, network awareness. In other words, it cannot control, influence or see how traffic flows through the network from one TPEP to another. This has serious implications for traffic engineering, fault isolation, load distribution, security, and so on. Proponents of overlay SDN often assert that since datacenter network fabric is invariably highly resilient and significantly over-provisioned this is not a significant issue. The argument is less convincing when heading out of the datacenter into the campus and across the WAN.



With overlay SDN, applications are "tunnelled" over an existing physical network, which offers no visibility into the paths applications are taking. With underlay SDN, the fabric of the underlying network is reconfigured to "imprint" the application paths onto the network fabric, providing knowledge and control over application paths.

Underlay SDN (OpenFlow, Cisco ACI, QFabric, FabricPath, etc.) directly manipulate network component forwarding tables to create specific paths through the network. In other words, they intrinsically embed the end-to-end network paths within the network fabric. The SDN controller is responsible for directly manipulating network element configuration to ensure that the requirements presented at the controller's northbound API are correctly orchestrated. With intimate knowledge of network topology, configured paths through the fabric and link-level metrics (e.g., bandwidth, latency, and cost), much more efficient utilization of network infrastructure can be achieved using more complex route packing algorithms, for example, sub-optimal routing. Another advantage of underlay SDN is that the controller dictates exactly where in the network each traffic flow traverses, which is invaluable for troubleshooting, impact analysis, and security.

The industry is currently split between network architects preferring overlay networks to those preferring underlay networks. It is not a decision to be taken lightly as it has far-reaching implications on complexity, troubleshooting, monitoring, SLA compliance, performance management, RCA and cost.

All Virtual or Some Physical

It is possible to implement SDN solely using existing network infrastructure and to provide SDN overlay connectivity using tunneling technologies. That said, most people concede that some physical hardware will be needed to perform certain key network functions or at key locations within the network. Fortunately there are a growing number of physical network appliances which are SDN conversant and can form an integral part of an SDN deployment. The most widespread SDN southbound protocol is OpenFlow, and many hardware vendors now provide switches, routers, and so on, offering OpenFlow APIs including Cisco, Brocade, A10, Extreme, HP and many other mainstream network hardware vendors. For proprietary SDN solutions (such as Cisco ACI) the vendor's own kit is mandated.

Pros of SDN

The three most commonly propounded benefits of SDN are efficiency, agility, and security.

Efficiency

Early, hyper-scale pioneers of SDN such as Google astounded the networking community with the increased network utilization they achieved using SDN. The industry norm for WAN link utilization is between 30% and 50%, whereas, by deploying SDN, Google drove utilization up to 95% (without impacting critical flows, losing traffic, etc.). This is primarily due to SDN's holistic view of the network and deeper understanding of inter-application requirements allowing SDN controllers to perform far smarter traffic engineering, route determination and load balancing than traditional QoS implementations. Using techniques such as sub-optimal routing for less time-critical traffic allows more circuitous routes to be employed to better utilize less desirable links, thus reducing congestion on faster, more expensive links for time critical traffic.

Another emerging feature of SDN controllers is the ability to pre-compute failover paths for critical flows, especially over particularly vulnerable or unreliable links.

Agility

Within the datacenter SDN can massively help with automation of network reconfiguration and enhance virtualization agility. By having a complete view of the datacenter, virtual machines, virtual switches, load balancing services/appliances and the underpinning physical network infrastructure, combined with the virtual machine to service and security policy mappings, the SDN controller can reconfigure the SDN to allow seamless migration of virtual machines around the network. This ensures that security constraints are maintained, and service chains (i.e., the linkages between VMs to network services such as load balancers, firewalls, IDS, etc.) are preserved.

> "To use SDN effectively there needs to be a sharing of knowledge between business requirements, applications, servers and networking teams. Without this, mapping services to application requirements and ultimately network requirements cannot occur effectively."

Within the broader campus, SDN can provide similar advantages for network device mobility combining wireless and wired network management, ensuring dynamic QoS compliance, traffic engineering, resource limiting, and security (e.g., RBAC).

Throughout the SDN estate, the ability to dynamically restructure service chains provides yet greater flexibility. The ability to dynamically insert a virtual load-balancer or a firewall, for example, into a service chain without needing to rack a new unit and re-cable is a powerful feature of SDN. Such service chain modification using NFV is an integral part of increasing network agility (and therefore business service agility).

Security

SDN can improve network security by providing basic (typically layer 2 to 4) packet filtering at network ingress and throughout the network, thus reducing the amount of undesirable traffic entering and traversing the network. Similarly with the ability to dynamically modify service chains and network connectivity it is easier to insert a physical or virtual firewall/IDS/IPS into a network path or orchestrate packet captures and flow analyses. With more dynamic (and therefore more up to date) security policies and RBAC there will be less scope for security and resource allocation loopholes to occur.

Cons of SDN

The majority of the drawbacks of SDN are organizational or financial and not technical. The most commonly cited reasons for actively not deploying SDN are outlined below.

Staffing

SDN requires significant staff re-training or recruitment. There are very few staff with proven SDN deployment/management skills making recruitment difficult. Therefore re-training of existing staff is necessary. Of course having retrained those staff, their retention with their new, highly desirable, scarce skills might prove problematic too!

Reorganization

To use SDN effectively there needs to be a sharing of knowledge between business requirements, applications, servers and networking teams. This might take the form of cross-discipline teams, or individuals with broad cross-discipline understanding. Without this, mapping services to application requirements and ultimately network requirements cannot occur effectively. This type of breakdown or merging of traditional “silos” is already occurring, for example, with the emergence of DevOps—albeit slowly!

Cost

Whilst proponents of SDN cite the real cost benefits of running a more highly utilized network and the less quantifiable benefits of a more agile network with more rapid application and service deployment, re-scaling, and so on, they often fail to factor in the costs of retraining, reorganization, new hardware and software licenses and the hidden costs of loss of business continuity during initial deployment.

Security

Whilst tighter, more dynamic security enforcement is recognized as an advantage of SDN, it is also a new, rapidly evolving technology with new protocols and new weaknesses and vulnerabilities. Such a lack of maturity and the possibilities for compromising the network control layers make SDN an appealing target for hackers, industrial espionage, and more.

> “Whilst proponents of SDN cite the real cost benefits of running a more highly utilized network and the less quantifiable benefits of a more agile network ... they often fail to factor in the costs of retraining, reorganization, new hardware and software licenses and the hidden costs of loss of business continuity during initial deployment.”

Obstacles To Deployment

Having covered the pros and cons of deploying SDN, for those deciding to embrace SDN there are still several obstacles to be overcome.

Which Vendor?

Probably the hardest decision is which vendor to select. Given that the various SDN options are predominantly not interoperable, the decision as to which vendor to select is a difficult one. Given the ongoing battle for market share between the major players (Cisco ACI, VMware NSX, OpenFlow) it would be unfortunate to pick a vendor who loses the battle and whose product atrophies and whose support wanes. It's rather like the battle between competing video formats in the 1970s!

Given the proprietary nature of NSX and ACI, selecting either of these products implies vendor lock in, whereas OpenFlow should offer broader vendor choices (although inter-vendor compatibility issues could still arise).

In terms of cost, Cisco ACI is the most expensive and has a very restricted set of supported hardware (currently just the Nexus 9000 and 7000) and software requirements (Cisco's APIC). This invariably implies an expensive network hardware overhaul as part of the SDN rollout. VMware NSX is less expensive as is Juniper Contrail. Solutions based on OpenFlow and Open-Contrail are far cheaper, but offer little support and the lack of enterprise-grade support may make these offerings unpalatable.

Which Architecture?

As discussed, the decision between overlay or underlay SDN is not a trivial one, with implications for complexity, virtual/physical network separation, performance and security management, troubleshooting, and so on, and obviously vendor.

Immaturity

One of the biggest obstacles to SDN deployment is the lack of maturity of any of the solutions. Being new technologies, none of them have a proven track record and many enterprises are hesitant to gamble their production networks on such new technologies. Furthermore the rapid, ongoing evolution of the standards and SDN controllers could require frequent upgrades to production networks which most are reluctant to embrace.

Given the newness of the SDN standards and the flux in the SDN controller market (especially for OpenFlow), most traditional network management

Monitoring SDN

One of the key requirements for a successful SDN will be comprehensive, deep, network visibility. With the dynamism of network overlays that SDN offers, there are many new challenges that SDN monitoring tools will need to address:

- ✓ Traditional elemental monitoring will still be required. Link level utilization, device availability/reachability, elemental health metrics (e.g., CPU and memory utilization, fan, PSU and module status monitoring, etc).
- ✓ Performance issue localization. When an application performs poorly, is it the application or the server or the network?
- ✓ Application flow visibility. Where is the application flow traversing, how are the devices and links along the path performing, is there any congestion, can a better path be selected?
- ✓ Proactive monitoring. If a link is getting congested, which application flows are being impacted or likely to become affected based on baseline traffic variation?
- ✓ Configuration change detection, verification and notification.
- ✓ RBAC awareness.
- ✓ Relevant network protocol support (e.g., VXLAN routing).
- ✓ SDN controller integration to garner topology, overlay configurations, and so on. This may require support for a range of SDN controllers.
- ✓ Capacity planning. This becomes more complex with SDN routing. For example, with sub-optimal routing there are many possible paths between two end-points, so the available capacity needs to be derived as a potential aggregate. Similarly, spare capacity between two end-points depends on which other overlay networks are passing traffic along shared links.
- ✓ Security monitoring will change with the advent of distributed firewalls, IDS, and IPS and their dynamic, distributed location. Protection of SDN control traffic (northbound and southbound) will be essential, as will security provisioning of applications using northbound APIs.
- ✓ The dynamic nature of SDN and virtualization technologies in general places additional demands on management and monitoring tools in terms of the timeliness of the data they present and their ability to perform historic replay.
- ✓ Given the complexity and path-oriented nature of SDN, network path visualization will become essential.

vendors have yet to provide meaningful (if any) SDN support. This is of particular concern for overlay SDN. For example, when the SDN controller reports that it has correctly configured a network path yet the applications are experiencing sporadic connectivity or performance problems, will network teams have the information they need to pinpoint and resolve the problems in a timely manner? What are needed are network management systems that understand the physical and virtual infrastructure (including SDN) and can present a holistic end-to-end view.

Whilst immaturity of solutions and tools is an issue, it is a blocker that will dissipate as products mature over time.

SDN Applications

The lack of standardized northbound APIs is leading to the production of controller specific orchestration applications rather than applications decoupled from controller specifics. Application developers are understandably reluctant to produce applications with multiple adaptors to interface with the many non-standard northbound APIs (for example, with OpenFlow there are many controllers, the most common being OpenDaylight, OpenStack Neutron, Beacon, Floodlight, IBM PNC, etc.). To obviate this obstacle the ONF is looking to standardize the northbound API.

Upscaling

There is fear that small scale proof of concepts that perform well might not behave equally well in large scale production deployments. This is partly due to virtualized network functions running on hypervisors, virtual switches and VMs and being unable to attain the throughput of traditional dedicated hardware with custom ASICs.

Business Case

The lack of compelling business drivers (instead of business "nice-to-haves") makes many of these obstacles show-stoppers, or at the very least, show-delayers! Should compelling "must-have" business reasons emerge, many of these obstacles could be mitigated, or the risk embraced.

Implications of Adopting SDN

To deploy SDN there are a number of pre-requisite changes, both technical and organizational. The most fundamental and problematic changes involve the need for interdisciplinary teams, as these boundaries between business drivers, applications teams, server teams, and network teams become blurred.

"For effective SDN the network team will need an understanding of an application's network footprint (bandwidth, QoS requirements, security settings, etc.) Similarly the server team will need an understanding of the network infrastructure in terms of packing applications onto virtual machines and migrating these around the data center."

For example, for effective SDN the network team will need an understanding of an application's network footprint (bandwidth, QoS requirements, security settings, etc.) Similarly the server team will need an understanding of the network infrastructure in terms of packing applications onto virtual machines and migrating these around the data center. In order to orchestrate the end-to-end application flows, the network team are likely to require some programming skills not just networking knowledge. The provisioning of new network overlays fits well within a DevOps environment where new applications require new overlays to be designed, implemented, QA'd and brought into production with greater rapidity than in a more traditional organisation.

Clearly with SDN, the way in which networks are designed, implemented, and maintained (including everyday troubleshooting, break-fix, network expansion, etc.) will change, requiring new security models and new network architectures. Underpinning this will be new and different network monitoring tools.

Timing

Aside from a few hyper-scale SDN deployments and service providers, few organizations are aggressively deploying SDN and reaping the potential benefits. In Google's case, significantly improved WAN utilization relates to a tangible business benefit (reduced WAN cost), but for most enterprises and SMBs there isn't yet a compelling business case, and it clearly comes with significant reorganizational and training costs as well as CAPEX. Couple this with the potential for business disruption as the existing network infrastructure is migrated to a hybrid SDN and it's obvious why organizations are being cautious.

Given the lack of familiarity with SDN technologies and their deployment, the immaturity of SDN technologies and tools (especially for SDN orchestration), the lack of monitoring ability and the rapid evolution of all of these, it's unsurprising that most organizations, whilst interested in SDN and experimenting with SDN in their test labs, are not yet rolling out to production.

Conclusions

The key question for this paper is: Should an organization embrace SDN now, soon, or hold off for now?

Most organizations are keen to learn about SDN but are not yet ready to deploy in production. Many are experimenting with various SDN technologies to try to assess which vendor's solution is best from them, be it Cisco's ACI (all encompassing, but expensive), VMware's NSX/Juniper Contrail (cheaper, simpler, tunneling technology but with concerns around the lack of application path visibility) or an open solution such as OpenFlow, which avoids vendor lock in, offers the best chance of API standardization and therefore should provide a wider range of orchestration application options, but without enterprise-grade support.

With the lack of compelling business drivers, most organizations are leaving deployment to those who are less risk averse, can justify the expense of being in the vanguard for a specific SDN benefit, or are more experimental in nature (e.g., universities).

My advice for now is to watch, read, experiment, learn from others' mistakes and be ready to engage when the appropriate business drivers emerge and the technology stabilizes and is proven. Also assess NMS vendor's current offerings (and roadmaps) in light of the issues discussed above. The only exception to waiting is where a significant network upgrade is imminent in which case consideration should be made of the possibility of purchasing SDN compliant hardware (e.g., networking infrastructure with OpenFlow support, ACI supported hardware, etc.) as a possible precursor to future SDN deployment.

ABOUT ENTUITY

Entuity takes the work out of network management. Our highly automated, unified enterprise-class solution puts deep network insight at your fingertips, frees IT staff to focus on strategic projects and easily integrates with major frameworks and networking environments. Entuity's support and services teams are frequently praised for their rapid response, networking expertise and involvement in special engagements. Founded in 1997 by two senior-level IT executives from the financial industry, Entuity is headquartered in London with US operations in Boston. For more information, visit entuity.com.

entuity
Taking the Work out of Network Management

entuity.com