DRIVENETS

Multiservice: Maximizing Infrastructure Utilization

The Cross-Service Synergy Effect

WHITE PAPER

Monolithic Architectures Reflect an Inherent Resource Utilization Wall

The old rule of a “single service router” typifies much of today’s network architecture. This means that each service, whether a mobile backhaul, provider edge, business service, internet peering or core network, is tightly coupled with its own batch of dedicated chassis-based routers, for connectivity.

While the inefficiency of this model may seem obvious, nevertheless, for years already, network vendors have perpetuated this approach and service providers have tagged along for the ride.

The classic chassis, which was, and still is, the basis for most networking functions is built in a manner that drives inherently inefficient resource utilization, and therefore a sub-optimal CapEx and OpEx structure.

Such chassis are based with a fixed set of resources, of different types:

- Compute resources – x86 CPU executed instructions
- Storage resources - RAM, storage etc.
- Bandwidth resources – traffic forwarding in and out of and within the chassis (AKA data plane)
- TCAM (Ternary Content-addressable memory) resources – include high-speed memory resources, used to store Forwarding Information Bases (FIB), Access Control Lists (ACL), counters etc.
- Quality of Service (QoS) resources – include buffers, queues, policers etc. used for SLA (Service Level Agreement) mechanisms implementation.

As monolithic chassis are designed to accommodate various networking scenarios, they are implemented to one specific networking scenario. the set of resources is built as a superset to accommodate any scenario and is fixed, for any given platform.

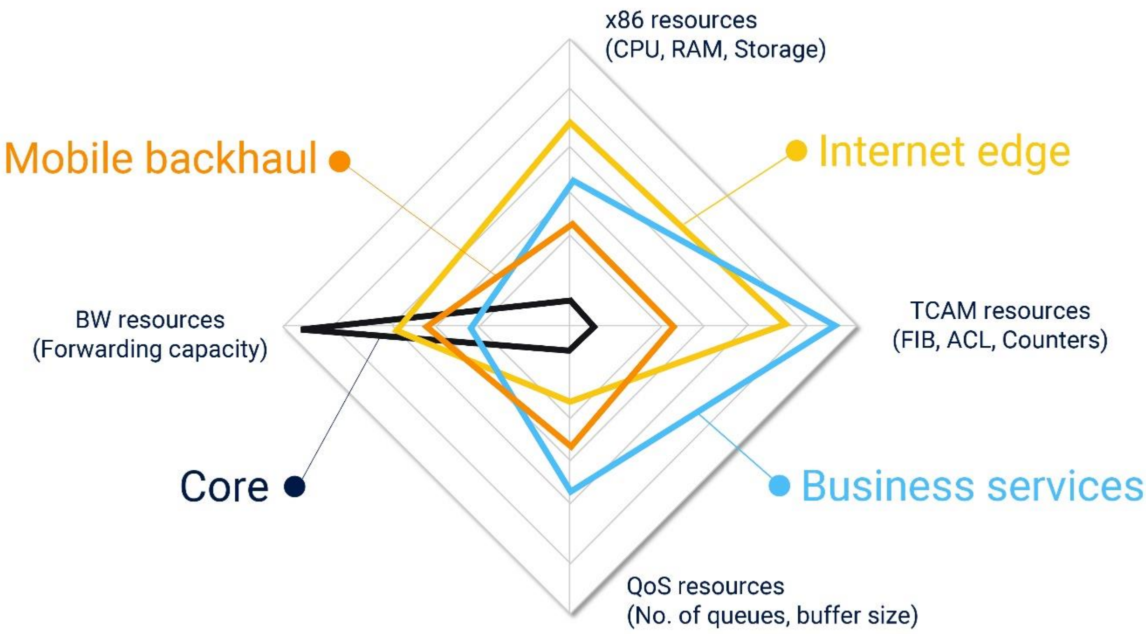
When using the platform in a specific network scenario, the inherent inefficiency of the monolithic architecture makes it under-utilized, for at least one of the resources mentioned above.

This is because most networking functions require a different set of resources, which is usually a mixture of 1 or 2 intensely used resources and other, lesser used resources.

For instance, a core function (e.g., P router) requires intense bandwidth resources as it deals, predominantly, with packet forwarding and label switching (no routing, access lists and QoS are required). For this function, x86, TCAM and QoS resources will be hardly utilized.

On the other hand, an edge router, in the same network, will require more x86 and TCAM resources but not that many bandwidth resources, as it deals with lower capacity streams that need specific forwarding, access lists etc.

The drawing below illustrates the resource requirements of a few typical networking functions and services, highlighting the difference in the requirement mix – leading to under-utilization of resources in any networking scenario.



This means that there are wasted resources in almost any monolithic router deployed in any specific network scenario.

Those resources are a major part of the router cost-structure, so the outcome of that, is that operators pay for hardware resources they do not use, regardless of network growth trajectory.

In practice, this outcome starts from day one of launching the network and peaks when the intensely used resources in a monolithic chassis are exhausted. At that point, the chassis is blocked, even though there is still an abundance of unused resources (the mildly used resources in that specific networking scenario) within it.

As the network becomes larger, it grows in the consumption of these favored resources, and further intensifies the abuse of unused resources. with more addressed networking scenarios, this inefficiency eventually expands to all hardware resource types and accumulates, turning into a significant operator investment that does not yield any benefit. Wasted money, in other words.

The Solution: Shared Resources Across Different Networking Functions

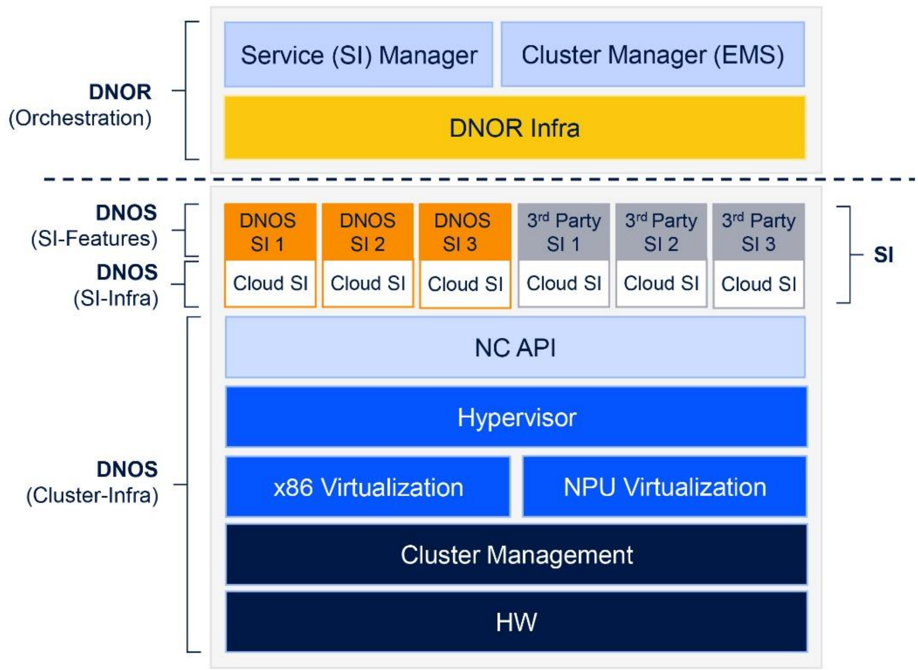
The DriveNets Network Cloud solution addresses this inefficiency and offers operators a way to significantly increase their resource utilization while gaining service & architecture flexibility, optimal scaling and software-paced innovation and time to market.

The base for this significant value is the way the Network Cloud is built – in a disaggregated, cloud native architecture.

This means the hardware resources of a cluster of multiple white boxes are abstracted by the DriveNets Networking Operating System (DNOS) to a level in which it is consumed as a virtualized resource pool. Each networking function, which runs a containerized Service

Instance (SI), can be allocated with its required hardware resource (Physical interfaces, NPU, CPU, TCAM, QoS etc.) out of the underlaying shared hardware infrastructure.

The following diagram illustrates the software architecture that allows this:

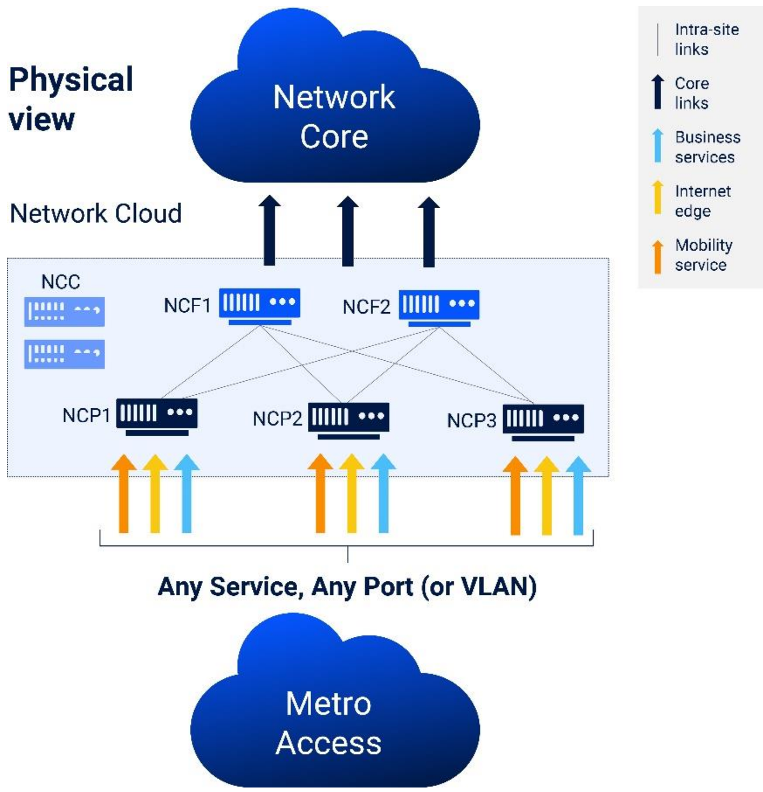


While different hardware resources are abstracted and different cloud SIs (from DriveNets or 3rd party) run over it, the system represents each network function as a standalone node, keeping the network manageable. This means that a single cluster can accommodate multiple networking functions that are physically collocated, yet logically separated. This is enabled by the microservice based architecture of DNOS as a multiservice architecture.

A multiservice architecture creates a separation of the data plane from the service plane (or control plane). Multiple services can coexist over the same virtualized physical infrastructure and as long as resources can be made available for a service to perform, the network cloud can launch this service into production.

In a typical scenario – P and PE routers, as well as DPI and DDoS mitigation applications will co-exist over the same cluster, sharing a pool of abstracted hardware resources.

The illustration on the right, further depicts such a scenario:



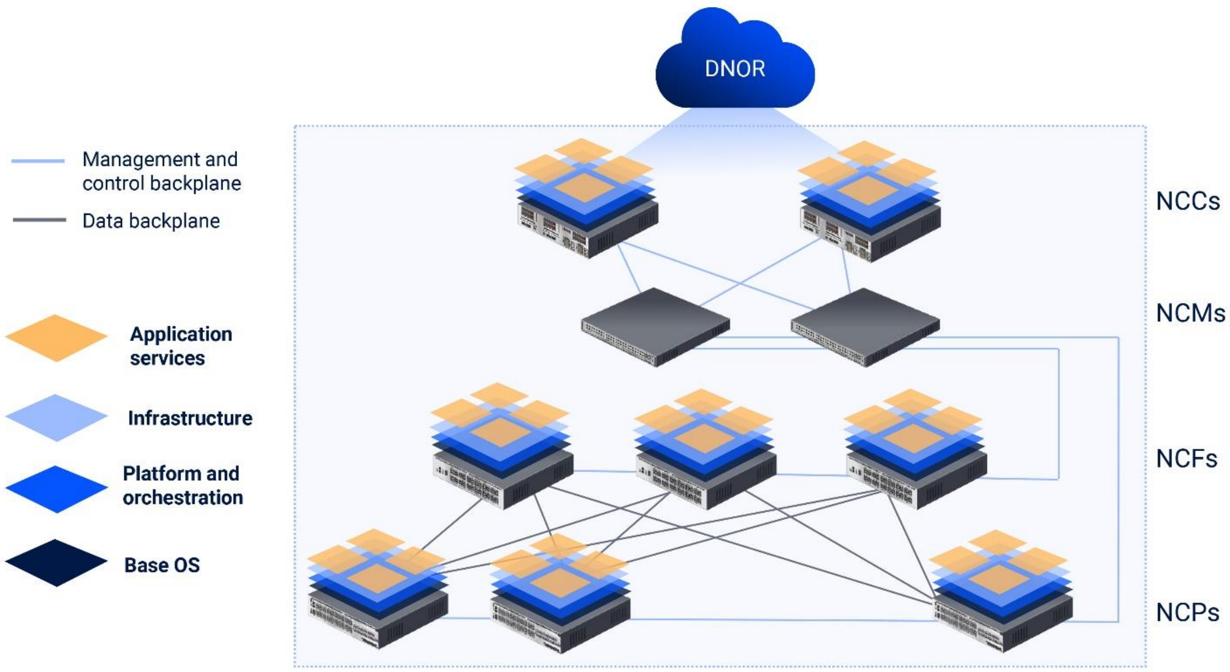
This shared use of resources enables operators to optimize the above mentioned resource utilization. Moreover, as this is a software-based allocation of resources, it can be dynamic, which allows immediate response to network events and changes in traffic and service patterns.

This translates into CapEx savings, as fewer ports and boxes are needed, as well as OpEx savings, in terms of footprint, power and managed devices.

Network Cloud Multiservice Implementation

In order to implement a multiservice architecture in a manner that will allow the network operator to keep managing the network as if it was built with monolithic routers, a couple of virtual entities were introduced:

- Service Instance (SI): a network function (e.g., PE router), which runs independently of any other networking function co-located on the same cluster. While the service instance is logical, it is assigned with dedicated resources out of the shared pool of hardware resources in the network cloud cluster.
- Inter Service Link (ISL): A logical connection between two SIs, which is represented as a (physical) link connecting the two networking nodes. This link allows control plane interconnection between the instances as well as data path features, including QoS and access lists. The ISL implementation is done over the abstracted physical fabric connectivity (NCFs in the below picture) and is totally logical, hence no cabling needed for setup or changes to such a link.

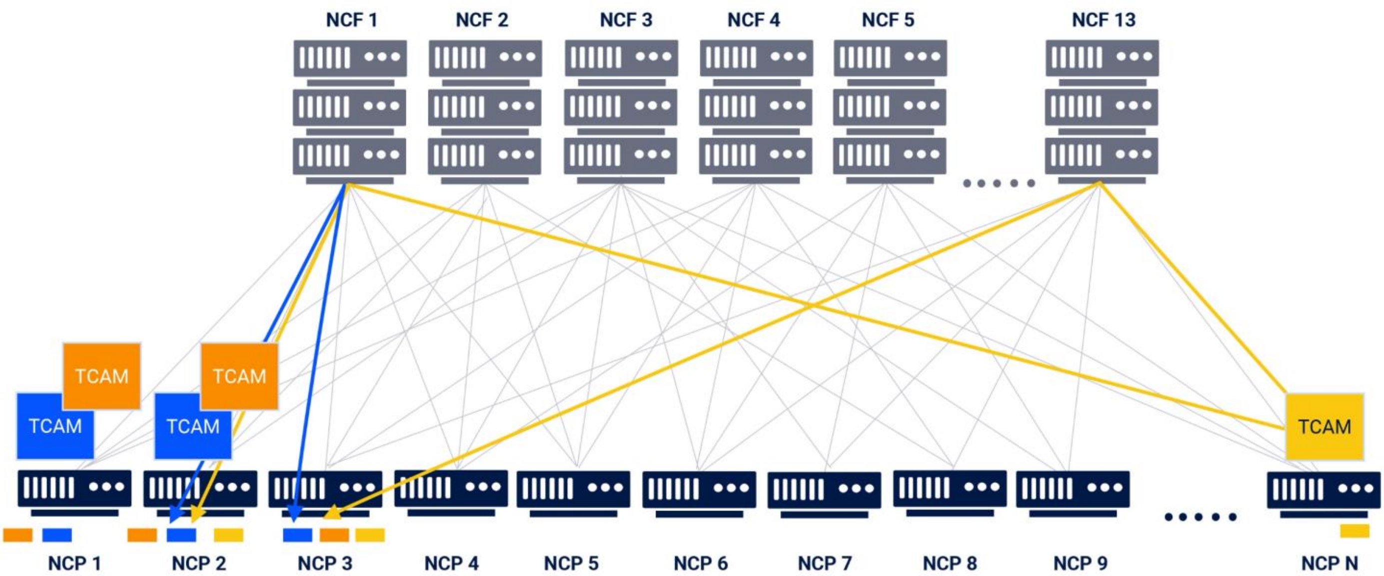


Once the SIs and ISLs are in place, the main task left is resource allocation. These resources are allocated to a service based on attributes assigned by the user as basic service requirements.

Applying the same concept of a pool of shared resource in the cloud, leveraging the virtualization of the compute and storage space, Multiservice optimizes the use of network resources by dynamically handing them as a shared pool regardless of their physical location in the cluster.

The multiservice solution allows for replacing the model of multiple routers in a single location, serving different networks (e.g., Enterprise, Mobile and Broadband), with a single unified infrastructure that can support all the networking services on a software-based network and cloud-native technologies (i.e., microservices, containers).

Let's take a TCAM allocation scheme as an example. Three different services are allocated to the same physical ports (Yellow could be a VPN service, Orange – an internet peering service and blue – a core instance). Their TCAM requirements are different, so there might be a case in which a packet arrives to a port, under the yellow service and encore a state in which no TCAM resources are available on that port. In such a case, it will use a TCAM resource from a different whitebox, which is available. This will be done over the cluster's fabric, as illustrated below.



SI	Maximum rate	Maximum TCAM size
Yellow	100% rate	100% TCAM size
Orange	200% rate	50% TCAM size
Blue	200% rate	50% TCAM size

Software-Paced Innovation

Multiservice within Network Cloud is not only about grouping several well-known services under one roof. Due to the decoupling of the data and service planes, a new service can be added to an existing installed base of the network. The introduction of the new service will include containerizing it and interfacing it with the Network Cloud open API to achieve peak performance. At this point the service can be piloted into any location in the network, tested, measured and eventually evaluated for full commercial deployment, which by itself, is a mere click of a button.

This opens a new era where the first cost of launching new revenue generating services is decoupled from the hardware and therefore lowers the barrier of entry for new services. The result is more active revenue generating services and fewer abandoned services, having already failled to convincingly deliver sufficient ROI.

An example of such new services can be a anything from a new VPN service to be mounted onto the network, via IOT control function that requires a presence within the network and can expand as far as unique differentiating services incepted by the operator to increase ARPU and the subscriber base while preventing user churn.

Conclusion

DriveNets Network Cloud multiservice architecture allows operators to enjoy synergies between different network functions, which require a different set of hardware resources, in order to resolve the inherent inefficiency of the monolithic router architecture.

This allows operators to leverage significant cloud attributes translating into CapEx and OpEx savings, derived from reduced port-count and hardware resources required in the network, simplified operations, lower footprint, power consumption, cabling efforts etc.