This document discusses the application of NFV and SDN concepts to the access platforms in general and wireless base stations in particular. It shows the ‘vBTS’ (virtual base station) development platform that is used to discover and solve implementation challenges as well as jump start system development.

Introduction
The goal of the wireless industry is to start the roll-out of next generation, “5G” networks in the year 2020. The ambition for this network is to meet increased performance requirements (latency, throughput) above what is provided by the LTE/LTE-A standards, while reducing network costs (CAPEX/OPEX) and improving network agility to quickly deliver new services. Network flexibility will be critical in enabling network operators to develop new business models beyond the current voice and data services. Next generation wireless networks such as LTE-A and 5G will be characterized by a heterogeneous network deployment model where the traditional macro cell base station model is augmented by small cells (femto, pico, micro) deployment, as well as very large centralized systems supporting hundreds of sectors at a time. At the same time, we envision a mix of access technologies (i.e., WiFi® and LTE) to co-exist and act in a coordinated manner. A challenge for network equipment vendors is to support this wide range of systems with as few as possible software and hardware architectural changes.

SDN and NFV concepts can help face this challenge by offering a well-defined and scalable software platform that can be used as a baseline for access network development. SDN and NFV hold promise to optimize for capacity, as can be seen with cloud technologies that leverage these concepts. However, in order to provide required system latency and throughput performance, a number of challenges still need to be solved.

Table of Contents
1 Introduction
2 vBTS, Access Platform Vision
5 The vBTS Development Platform
6 Wireless Access Platform Provisioning Tool Kit
6 Conclusions
vBTS, Access Platform Vision

Network Function Virtualization (NFV) is the concept of providing a standardized software development platform on which different networking service functions can be executed and managed. This allows for decoupling the software innovation cycle from the compute platform that the software is running on. By standardizing the platform to General Purpose Processors (GPP) and standard interfaces, “network appliances” become software applications that can be developed by anyone, no longer tied to a specific hardware platform. Virtualization is one of the key building blocks supporting this concept, allowing for protection, resource partitioning and fault isolation – essentially enabling different network appliances to be developed and executed independent of each other in a multi-vendor environment, without having to be aware of each other.

Applying the NFV concepts to the Radio Access Network (RAN) application provides several benefits that similarly apply to a myriad of networking applications:

- **Reduced service cost and increased service velocity due to unprecedented agility.** ETSI NFV MANO and Openstack initiatives can play a key role in standardized orchestration, automated provisioning and VNF/VM/vBTS life cycle management.
- **Greater service innovation due to abstraction and decoupling of HW and SW via virtualization.** ETSI INF and SWA initiatives (including in conjunction with ONF and ODL/ONOS SDN initiatives) are intended to enable rapid service enhancements through service insertion and chaining.
- **Improved service quality and reliability with virtualized containers, i.e. VNF/vBTS.** ETSI NFV REL and SEC and OPNFV initiatives contemplate how virtualization aids by providing the application developer with a well-defined sandbox, impossible to corrupt and/or pose a threat to other SW instances, and thus limiting the impact of an application crash/restart/malfeasance to the scope of the VM in which the application is executed. Such containment becomes increasingly relevant with virtual tenant growth that is enabled in multiple ways, e.g. cores in SoCs, VMs in nodes, fatter pipes, etc.
- **Scalability and agility.** Decoupling hardware from software and providing HW independence allows for both scaling as well as rapid adaptation to new platforms.

Software Defined Networking (SDN), for example, as defined by the Open Networking Foundation (ONF), is about separation of control plane (CP) and data plane (DP) operations. Separation of these components from a software/API perspective allows them to potentially run in different hardware, physically co-located or separated and in fact non-virtualized or virtualized, where in the latter case NFV concepts can also come into play. It allows for a system-wide view/control of the network and a standardized control mechanism for a multi-vendor environment. It allows operators to be in control of network behavior and feature set.

The concept of CP/DP separation has been around since the early 2000’s with the advent of network processors, and the concept of standardization of the CP/DP API has been proliferated through initiatives such as the Network Processor Forum (NPF) (http://www.oforum.com/public/NPF_IA.html). A fair question at this point is: “Why should people adopt SDN explicitly?” Part of the answer is: SDN allows for the replacement of control that is standard, application-dependent, and often distributed with control that is centralized, and generalized across standards and applications. This means that SDN matters to entities that can benefit from centralized control, and that are in a position to develop application-specific control software.

Motivations often involve simplicity of network management, but even the management becomes network specific: SDN provides for a standardized interface between control plane and data plane, as such “exposing the network” to networking applications developers.
OpenFlow™, a SDN standardization vehicle driven by the Open Networking Foundation (ONF), is a concept that is closely associated with SDN. OpenFlow defines an API level interface between (today) Layer2 (L2) switching functions executing in the DP and higher level CP functions, allowing for separation of DP control and monitoring. By defining a dedicated DP component, the OpenFlow concept accepts the notion that certain functions are indeed best executed on an optimized hardware platform such as a network processor. The OpenFlow concept introduces centralization of the CP using the SDN principles, allowing (but not forcing) the CP to move to a remote (centralized/"cloud") environment.

A prime candidate in the LTE base station for standardization through SDN is the S1/X2 interface (base station backhaul). Current implementations of the L2/L3 functions (Ethernet, QoS, IP/IPSec, etc.) associated with this interface are exposing system vendor specific APIs. Managing evolving APIs from multiple system vendors is not efficient and makes network management more complex. Standardized control through SDN allows for backhaul networking automation.

SDN is a useful tool, especially in evolving HetNet type of deployment scenario's with non-uniform backhaul evolving over many product generations and system vendors.

**Challenges**

Current base station implementations rely on vendor proprietary software that executes on a combination of General Purpose Processors (Power Architecture®, ARM®, MIPS executing L2/L3 stacks) and dedicated hardware acceleration/signal processors (executing L1 stack as well as L2/L3 offload). L2/L3 processes execute either on a proprietary RTOS, or on Linux®, enabled with appropriate realtime extensions, like PREEMPT_RT. This partitioning approach between hardware and software is taken due to a number of challenges with 3GPP/LTE, and need to be solved in a NFV environment. Specifically:

- Latency (realtime behavior) & performance (maintain throughput) are critical in LTE
- Hardware acceleration (for performance/efficiency) needs to be included in an NFV environment
The virtualized base station

Taking into account the challenges highlighted above, a practical approach to a SDN/NFV development platform – using the virtual base station (vBTS) as an application example of a virtual access platform (vAccess) is shown below.

Key proprietary implemented platform hardware components that build the vBTS development platform include:

- A GPP server platform that hosts the realtime and non-realtime components of processing stacks.
- An Intelligent Network Interface Card (iNIC). The iNIC framework supports the growing demand for intelligent network acceleration and application offload for converged datacenter applications, such as storage, security, deep packet inspection (DPI), firewall, wide area network (WAN) optimization and application delivery computing (ADC).
- L1 accelerator. We use a standalone L1 accelerator that is connected to the L2/L3 stacks through Ethernet. The L1 accelerator contains a “translator” agent that forwards a standardized message interface (FAPI) over the physical connection. We offers a range of devices targeted for L1 acceleration, scaling from low- to high-end.

The vBTS Development Platform

Our company is developing a prototype virtualization platform that is intended for use as a development platform for NFV-based applications. The purpose is to provide all required hardware and software components that build the infrastructure in which any application can be developed. The virtualized wireless base station application is used as a prototype, showing how timing and performance challenges can be met, without enforcing specific system partitioning. The platform is using the base station application as an example, but is targeting other applications (eg. generic access platform) as well.
Key software components are:

- **KVM-QEMU hypervisor.** Our proprietary SDK supports virtualization through Linux Containers, a proprietary type 1 Hypervisor (Topaz) and KVM Hypervisor for Network Function Virtualization. Our proprietary SDK is used for board bring up and installing OpenStack Components like Nova-API, Neutron Agent, OpenVSwitch etc to use our proprietary GPP platform as Compute Node of OpenStack Setup.

- **PREEMPT_RT.** Latency requirements target some very specific software partitioning goals. The “L2” application needs to be prioritized to start within a fraction of the 1msec TTI boundary. This puts limits on CPU resource sharing between multiple base station applications that need to start on the same TTI boundary. It also requires a bounded/guaranteed maximum to this application start time. The purpose of the PREEMPT_RT patch as available for Linux is to provide support for realtime operation of Linux applications assuming the application uses a subset of Linux services. Some applications, such as the virtualized L2 of a base station stack have been proven to execute in realtime using PREEMPT_RT in a non-virtualized environment for many years.

- **NFVI Acceleration (NFVixl).** As mentioned, the NFVixl iNIC application offloads the VMM networking from the general purpose processor cores of networking nodes that host the VNFs, like vBTSs, thus mitigating the performance impact of the VMM virtualization layer introduced in NFV.

- **OpenStack cloud orchestration.** OpenStack is an open source cloud computing platform for public and private clouds, which is simple to implement, massively scalable, and feature rich. The technology consists of a series of interrelated projects delivering various components for a cloud infrastructure solution, as shown below. For more information about OpenStack, please refer to application note AN4646.
**Wireless Access Platform Provisioning Tool Kit**

The OpenStack controller and associated agents in the compute nodes provide the ability to bring up and bring down vBTS VMs. OpenStack is extendable to add additional components for enabling the configuration and management of applications. The Wireless Access Platform Provisioning Tool Kit is a reference tool kit that is developed on top of the OpenStack controller to configure/provision L1 and vBTS VM applications. It also has some additional software components (agents) running on vBTS and L1 devices. This toolkit provides two key benefits: automating the mapping of L1 devices with vBTS VMs and configuring the applications of L1 devices and vBTS VMs.

**Conclusions**

SDN and NFV concepts have an identified set of advantages, but current deployment in networking access platforms is limited due to technical challenges, mainly associated with latency and performance. Being a market leader in all aspects of the RAN (from Transport/Control, L2/L1 up to RF), our company is uniquely positioned to help system vendors analyze and implement vBTS systems.

The next generation access network development platform is intended to prove out solutions to technical challenges as well as jump-start customer system development.