Introduction to UGP

This chapter introduces the Cisco Ultra Gateway Platform (UGP). The UGP is a distributed virtualized packet core platform that supports scalability for virtualized cloud architectures by extending the boundaries beyond a single virtual machine (VM).

Product Description

This chapter describes the StarOS UGP architecture and interaction with external devices.

Hypervisor Requirements

For information about Hypervisor Requirements, see *Software Requirements* section in the *Ultra Services Platform Deployment Automation Guide*.

Underlying Infrastructure for the System

This virtualized system can be deployed into a new or existing Infrastructure as a Service (IaaS) cloud datacenter. UGP runs in a set of virtual machines (VMs) using industry standard hypervisors on Commercial Off The Shelf (COTS) servers. This deployment model allows the management of physical infrastructure to remain outside of the scope of StarOS and UGP.

A typical instance runs on the following NFVi (network function virtual infrastructure):

- IaaS components
  - COTS blade chassis, blades, and fabric interconnects
  - Manager Software
  - Network Attached Storage (NAS) or Storage Area Network (SAN)
  - KVM hypervisor on each blade or server
  - KVM administration software
- StarOS software installed within VMs
- Each VM must run on a separate blade so if a blade fails, only a single VM is affected.
- Existing Management software (service, logging, statistics, etc.)
- Orchestration software (optional) [See Orchestration, on page 9]

A UGP instance is a grouping of VMs that act as a single manageable instance of StarOS. UGP consists of the following major components:

- Control Function
- Service Function
- DI Network, on page 2
- Orchestration Network, on page 4
- Service Network, on page 5

DI Network

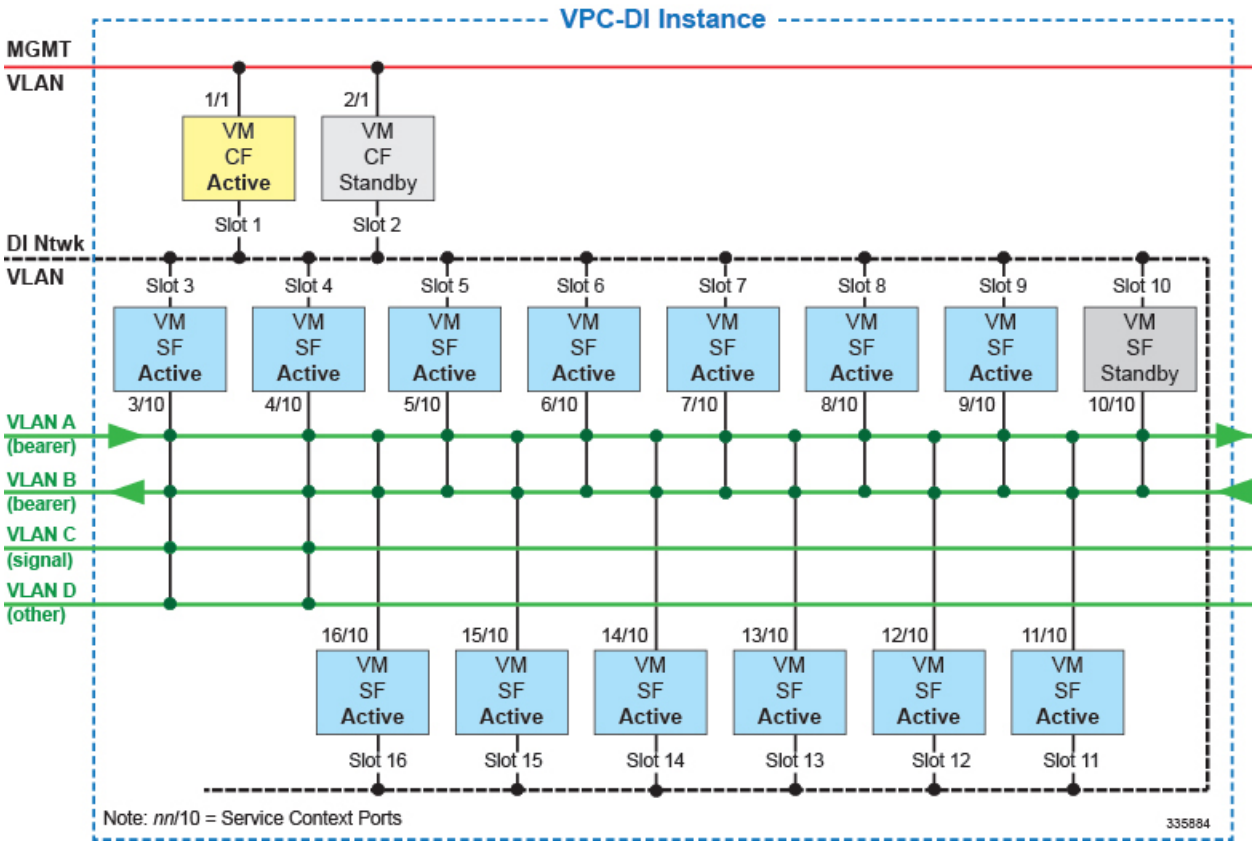
In order for the VMs within a UGP instance to communicate with each other, each instance must have a private L2 network that interconnects the VMs. This network should utilize a VLAN within the IaaS/virtualization infrastructure and be exposed untagged to each VM as the first vNIC.

The DI network must be for the exclusive use of a single UGP instance. No other devices may be connected to this network.

Note If more than one instance is instantiated within the same datacenter, each instance must have its own DI network.

All the VMs within an instance must be physically located in the same site, ideally in the same few racks with minimal interconnecting devices. The reliability of the DI network is important for the stability of the UGP instance. Using L2 tunneling protocols across a WAN or congested links is highly discouraged.

Figure 1: DI Network



Network Requirements

The reliability and performance of the DI network is critical to the reliability and performance of UGP. The DI Network is used for internal control, signaling, and bearer traffic. Bearer traffic may traverse the DI network multiple times, so any packet loss in the DI network would impact the perceivable packet loss of UGP as a whole.

Note The infrastructure connecting the VMs should be 10 Gbps or higher between all VMs and have a redundant configuration. A redundant configuration can be provided in one of these ways:

- on the host using a vSwitch (for Virtio/VMXNET3 interfaces)
- on the hardware, such as the Cisco UCS virtual interface card (VIC)
- in the UGP using network interface bonding

The IaaS/hypervisor must provide a DI network that can:

- Perform as a L2 Ethernet bridge/switch.
- Support jumbo frames up to at least 7200 bytes. If your installation does not support jumbo frames, you can still use the UGP.

The infrastructure/hypervisor should provide a DI network that can:

- Support 802.1p priorities sent from the VMs with VID=0.
- Honor 802.1p priorities end-to-end between all VMs within the instance.
- Provide redundant L2 paths in all physical infrastructure or 802.1p priorities end-to-end between all system VMs within the instance.
- Provide a secure network that limits access to authorized users only.

Specifically, the DI network should have the following minimum reliability requirements:

- Fully redundant L2 paths
- No outage longer than 1.5 seconds, including STP and LACP outages (if applicable)
- Packet prioritization
- Sufficient network bandwidth to minimize any control or bearer packet loss

Disruptions in the DI network or excessive packet loss may cause false failure detection or unpredictable behavior in the UGP instance.

Each system VM monitors the reachability of the other VMs and the reliability of the DI network on an ongoing basis.

Jumbo Frames

We recommend that the DI network support jumbo frames up to at least 7200 bytes. On startup, each VM issues a series of ping commands on the network to determine that jumbo frames are supported. Support of jumbo frames provides better system performance.

If your installation does not support jumbo frames, you can still use the UGP.

The CF and SF do not start if an MTU of less than 7200 is detected and the appropriate boot parameter is not set.

Service ports on SFs can also support a maximum MTU up to 9100 bytes in Release 21.4 and higher, or 2048 bytes in older releases if configured appropriately in the StarOS configuration.

Record Storage

Record storage is available on instance-wide storage devices available at **/records**. Both CF VMs are provisioned with a second vHDD (/hd-raid) of suitable size for record storage (minimum of 16GB). The CFs share a RAID configuration to mirror data between their vHDDs. The SFs send data records to the active CF over the DI network for transfer to external ephemeral storage that was created and mounted manually or orchestrated by the VNFM.

Orchestration Network

This network is for CF/SF and VNFM (ESC) communication. Specifically VNFM, CF and SF are attached to this network.

Service Network

This is a dedicated network to provide service interfaces (for instance S11, S10, and so on). All SFs will provide service using this network segment. Individual SF can be configured to have up to 4 ports as maximum to connect to Service network. SF ports are used to receive and transmit bearer and signaling packets. To simplify network settings and address usage, only VLANs for high-bandwidth (bearer) packets need to be connected to all SFs. Low-bandwidth interfaces (signaling) can be connected to just two SFs.

Packet Flows

SF ports are used to receive and transmit bearer and signaling packets. To simplify network settings and address usage, only VLANs for high-bandwidth (bearer) packets need to be connected to all SFs. Low-bandwidth interfaces (signaling) can be connected to just two SFs. In the diagrams below, the bearer VLANs are connected to all SFs, while signaling and other VLANs are only connected to the first two SFs.

Note This asymmetric arrangement means that fewer interfaces are needed, however careful consideration should be paid to failures since the loss of two VMs results in loss of services.

ECMP does hashing based on a hash and can send traffic to any SF VM.

On ingress, the SFs perform flow lookups and direct packets to the specific SESSMGR task on a specific SF. Some of this ingress traffic is processed by local SESSMGR tasks, otherwise it is relayed via the DI network to the correct SF. On egress, each SF sends out packets from its local port (provided ECMP is used). In most cases, the number of VMs that packets traverse is less than two. However, ACLs and tunneling may increase the number of hops for specific flows depending on the EPC configuration.

Packets Received on SF Demux VM

On the demux and standby SF, all session traffic received is relayed to another SF for session processing. The figure below shows how ingress packets are distributed via the Demux SF to other session SFs for processing.

Item	Description
1	Receive NPU does flow lookup to determine SESSMGR, (Ingress)
2	SESSMGR processes packet.
3	Transmit NPU sends packet out local port. (Egress)

Packets Received on SF Session VM

The figure below shows how ingress packets received by a session SF are distributed to other session SFs for processing.

Item	Description
1	Receive NPU does flow lookup to determine SESSMGR, (Ingress)
2	SESSMGR processes packet.

Item	Description
3	Transmit NPU sends packet out local port. (Egress)

DPDK Internal Forwarder

The Intel Data Plane Development Kit (DPDK) is an integral part of the VPC architecture and is used to enhance system performance. The DPDK Internal Forwarder (IFTASK) is a software component that is responsible for packet input and output operations and provides a fast path for packet processing in the user space by bypassing the Linux kernel. It is required for system operation. Upon CF or SF instantiation, DPDK allocates a certain proportion of the CPU cores to IFTASK depending on the total number of CPU cores. The remaining CPU cores are allocated to applications.

To determine which CPU cores are used by IFTASK and view their utilization, use the **show npu utilization table** command as shown here:

```
[local]mySystem# show npu utilization table

Wednesday July 06 10:53:55 PDT 2017
          -------iftask-------
   lcore     now   5min  15min
--------  ------ ------ ------
  01/0/1     38%    53%    52%
  01/0/2     51%    55%    55%
  02/0/1     66%    72%    68%
  02/0/2     66%    63%    67%
  03/0/1     57%    55%    55%
  03/0/2     51%    47%    45%
  03/0/3     89%    89%    89%
  03/0/4     88%    88%    89%
  04/0/1     67%    59%    58%
  04/0/2     54%    40%    48%
  04/0/3     89%    89%    90%
  04/0/4     90%    89%    89%
  05/0/1     55%    55%    56%
  05/0/2     68%    45%    45%
  05/0/3     90%    89%    89%
  05/0/4     90%    89%    89%
  06/0/1     50%    58%    58%
  06/0/2     24%    24%    25%
  06/0/3     89%    90%    90%
  06/0/4     91%    90%    90%
```

To view CPU utilization for the VM without the IFTASK cores, use the **show cpu info** command. For more detailed information use the **verbose** keyword.

```
[local]mySystem# show cpu info card 6
Tuesday July 05 10:39:52 PDT 2017
Card 6, CPU 0:
  Status                : Active, Kernel Running, Tasks Running
  Load Average          : 7.74, 7.62, 7.54 (9.44 max)
  Total Memory          : 49152M
  Kernel Uptime         : 4D 5H 7M
  Last Reading:
    CPU Usage           : 25.4% user, 7.8% sys, 0.0% io, 0.1% irq, 66.7% idle
    Poll CPUs           : 4 (1, 2, 3, 4)
    Processes / Tasks   : 177 processes / 35 tasks
    Network             : 164.717 kpps rx, 1025.315 mbps rx, 164.541 kpps tx, 1002.149 mbps
 tx
    File Usage          : 8256 open files, 4941592 available
```

```
   Memory Usage        : 21116M 43.0% used
 Maximum/Minimum:
   CPU Usage           : 32.9% user, 8.9% sys, 0.0% io, 0.4% irq, 59.1% idle
   Poll CPUs           : 4 (1, 2, 3, 4)
   Processes / Tasks   : 184 processes / 36 tasks
   Network            : 178.388 kpps rx, 1270.977 mbps rx, 178.736 kpps tx, 1168.999 mbps
tx
   File Usage          : 8576 open files, 4941272 available
   Memory Usage        : 21190M 43.1% used
```

Bandwidth Requirements

Modeling of bandwidth requirements on the L2 switches that host a UGP instance is required for each operator deployment.

In addition to the predominant bearer traffic, the DI network also passes session signaling and internal control data between the VMs.

Internal control traffic will be heavy during redundancy operations, but significantly less under normal operation. Heavy use of control traffic occurs during:

- Migrations of tasks from an active SF VM to the standby SF
- Startup or restart of a standby SF
- Startup or restart of an SF
- Startup or restart of an SF or standby CF
- Heavy signaling traffic (high Call Events per Second [CEP] rate)
- Significant CLI and/or Bulkstats usage

Depending on the CEPS rate, configuration, and management operations, each VM places a load on its DI network interface regardless of bearer throughput. This load is expected to be highly variable, but average less than 1 Gbps per VM with some VMs having higher usage than others.

Feature Set

Interfaces and Addressing

Each VM in a UGP instance is represented as a virtual card with a single CPU subsystem. This makes many CLI commands, logs, and functions work similarly to StarOS running on ASR 5500 platform.

StarOS concepts of contexts, services, pools, interfaces, cards, and ports exist on each VM just as on ASR 5500 platform.

When the VM boots, the vNICs configured in the VM profile are detected and an equivalent number of "Virtual Ethernet" type ports appear in the StarOS CLI.

By default, the system assigns the vNIC interfaces in the order offered by the hypervisor.

- CF VMs (slots 1 and 2)
    - First interface offered (1/0 or 2/0) is for the DI Network.

- Second interface offered (1/1 or 2/1) is for the management network.

- SF VMs (Slots 3 through 16)
  - First interface offered (*slot*/0) is for the DI Network.
  - Traffic Interfaces *slot*/10 through *slot*/21 are for IaaS VLAN control and data traffic.

Note StarOS supports up to 12 service ports, but the actual number of ports may be limited by the hypervisor.

It is critical to confirm that the interfaces listed in the supported hypervisors line up with the KVM bridge group in the order in which you want them to match the VM interfaces.

Note You cannot be guaranteed that the order of the vNICs as listed in the hypervisor CLI/GUI is the same as how the hypervisor offers them to the VM. On initial setup you must use the **show hardware** CLI command to walk through the MAC addresses shown on the hypervisor vNIC configuration and match them up with the MAC addresses learned by the VMs. This confirms that the VM interfaces are connected to the intended bridge group.

Encryption

VMs within a UGP instance perform software-based encryption and tunneling of packets (as opposed to the higher-throughput hardware-based services). Call models that make heavy use of encryption for bearer packets or have significant PKI (Public Key Infrastructure) key generation rates may require significant compute resources.

If your COTS server hardware uses the Coleto Creek chipset based on the Intel 89xx chip, the system automatically utilizes this hardware chip for encryption and decryption of packets. However, all service function VMs must use this chipset in order for the system to use the hardware chipset for encryption and decryption.

Security

Security of external traffic including tunneling, encryption, Access Control Lists (ACLs), context separation, and user authentication function as on existing StarOS platforms. User ports and interfaces on the CFs and SFs are protected through StarOS CLI configuration.

The virtual system adds additional security concerns on the customer because network communication travel over the DI network on datacenter equipment.

The DI network must be isolated from other hosts within the datacenter by limiting membership in the system network's VLAN to VMs within that specific UGP instance. Unauthorized access to the DI network through other hosts being inadvertently added to that network or the compromise of a router, switch or hypervisor could disrupt or circumvent the security measures of StarOS. Such disruptions can result in failures, loss of service, and/or exposure of control and bearer packets. Properly securing access to the DI network is beyond the control of StarOS.

Communication between DI network component (e.g. CF and SF) VMs is now only possibley via authentication over externally supplied SSH keys. In addition, the system enforces public/private key-based SSH authentication for logins within the DI network. No passwords, keys or LI information are stored or sent in clear text.

If an operator requires physical separation of networks, such as management versus bearer versus LI (Lawful Intercept), then physical separation of the DI network should also be done since it carries sensitive data. In a virtualized environment, the physical separation of networks may not be possible or practical. Operators that have these requirements may need to qualify their hypervisor and infrastructure to confirm that it will provide sufficient protection for their needs.

Orchestration

When a UGP instance is deployed, there are several expectations of the environment on which UGP is running that are beyond the control of StarOS. Most of these fall into requirement of the Orchestration System.

- Provisioning of UGP VMs including install and parameters assignment: configuration, connectivity, and persistent block storage for each VM.
- L2 provisioning of the DI network to ensure that the DI network meets reliability requirements.
- Policy enforcement of network separation, if applicable.
- Physical placement of VMs that enforce redundancy rules.
- Providing useful monitoring tools for physical resources, such as CPU, RAM, NIC, etc.

If an orchestration system is not used to deploy a UGP instance, these requirements must still be maintained. However, they should be enforced manually or through other means. Refer to VM Hardware Verification for information on monitoring the VM hardware configuration.

Provisioning

Provisioning of a UGP instance has two phases:

- VMs and network interconnections are created and linked.
- UGP instance is configured for services.

IaaS administrators set up and interconnect the servers and use hypervisor VM templates or orchestration software to create a set of VMs, the DI network, and the redundancy configuration to meet Service Level Agreement (SLA) requirements.

Deploying a UGP instance requires a detailed configuration plan that addresses the operator's deployment requirements.

Boot Sequence

StarOS is installed on each VM using pre-installed disk templates in QCOW2 format. Slot numbers are managed by ESC and OpenStack. A slot number is assigned as part of the VM configuration. The slot number is auto-detected during install. Installation of the installer image is completely automated provided that the slot number can be detected from the hypervisor. For additional information, see Software Installation and Network Deployment, on page 11.

Each VM will reboot and attempt to join the UGP Instance. A bootloader boots the instance via automated (scripted), network or manual booting.

Upon completion of the virtual BIOS, the VM boots from its local vHDD and runs CFE (Common Firmware Environment). CFE looks on the vHDD for the presence of the parameters file that was created during installation. If this file is found and parses correctly, CFE takes different paths depending on the VM's type and slot number. In all cases, the first vNIC becomes the interface for the network.

CF Boot Sequence

The CF performs the following functions during its boot sequence:

- Checks to see if the other CF is alive (via the DI network).
- If other CF is alive, attempts to boot from it.
  - Tries to obtain parameters and a boot image from the other CF.
  - If successful, transfers the boot image and runs it.
- If the other CF is not alive or booting from it fails, boots independently.
  - Finds and parses a boot.sys file on the local vHDD for boot/config priorities.
  - Performs a boot via instructions in the boot.sys unless interrupted by a user (via the Management network or local vHDD).

CFE on a CF supports downloading a starfile (bootable image) from the peer CF, via the CF management vNIC to an external HTTP or TFTP server, or from a local file on its vHDD. This is driven by the boot.sys and the StarOS **boot** CLI command.

Note HTTP and TFTP booting are only supported on VIRTIO interface types.

A network protocol on the DI network determines which CF is master. Mastership is then communicated over the DI network to SF VMs.

SF Boot Sequence

An SF boots from its vHDD. It then contacts the active CF via the DI network to determine if it booted the correct software version. If the SF did not boot the correct software version, it transfers the correct version from the CF and reboots itself. Once it boots the correct the software version, the boot sequence is complete.

Capacity, CEPS and Throughput

Sizing a UGP instance requires modeling of the expected call model.

Many service types require more resources than others. Packet size, throughput per session, CEPS (Call Events per Second) rate, IPSec usage (site-to-site, subscriber, LI), contention with other VMs, and the underlying hardware type (CPU speed, number of vCPUs) will further limit the effective number of maximum subscribers. Qualification of a call model on equivalent hardware and hypervisor configuration is required.

Software-based transmit batching greatly enhances the system performance.

Diagnostics and Monitoring

Because UGP runs within VMs, no hardware diagnostics or monitoring are provided. Retrieval of hardware sensor data (temperature, voltage, memory errors) are accomplished via the hypervisor and external monitoring systems. To determine the configuration of the underlying VMs, refer to VM Hardware Verification.

UGP monitors and exports vCPU, vRAM, and vNIC usage per VM through existing mechanisms including CLI **show** commands, bulkstats and MIB traps. However, an operator may find that monitoring physical CPU, RAM, and NIC values per host in the hypervisor is more useful.

Because vNICs have a variable max throughput (not defined as 1 Gbps or 10 Gbps for example), counters and bulkstats that export utilization as a percentage of throughput may have little value. Absolute values (bps) can be obtained from the VM, but where possible physical infrastructure utilization should be obtained from the hypervisor. This would not apply to pass-through PF NICs, as those have a fixed maximum throughput.

Cisco Prime Analytics

The Cisco Prime for Mobility suite of analytics provides scalable management of a UGP instance.

Cisco Prime for Mobility supports the following:

- Integrated operator workflows across the Radio Access Network (RAN) backhaul and packet core
- Centralized network visibility and advanced troubleshooting and diagnostics
- Pre-integrated network management software components that reduce time and resources required for integration

For additional information, contact your Cisco account representative.

StarOS UGP Build Components

The following StarOS build filename types are associated with UGP:

- **.qvpc-di-<version>.iso** initial installation or startover ISO file.
- **.qvpc-di-<version>.bin** update, upgrade or recovery file for a system that is already running.
- **.qvpc-di-template-libvirt-kvm-<version>.tgz** KVM libvirt template plus ssi_install.sh.
- **.qvpc-di.qcow2.tgz** KVM QCOW2 disk template.
- **.qvpc-di-template-vmware.tgz** VMware files.

Software Installation and Network Deployment

For additional information on supported operating system and hypervisor packages, as well as platform configurations, please contact your Cisco representative. The Cisco Advanced Services (AS) group offer consultation, installation and network deployment services for the UGP product.