White Paper

Class-Based Weighted Fair Queuing on Cisco Nexus 1000V Series Switches: Manage Congestion for Virtualized Data Center and Cloud Environments

What You Will Learn

This document describes the benefits of Class-Based Weighted Fair Queuing (CBWFQ) and how it works on Cisco Nexus® 1000V Series Switches. You will be able to:

- Describe the fundamentals of congestion management and CBWFQ on the Cisco Nexus 1000V Series
- Understand important planning and configuration considerations for implementing CBWFQ on the Cisco Nexus 1000V Series
- Configure and verify CBWFQ on the Cisco Nexus 1000V Series

Intended Audience

This document is intended for virtualization architects, network engineers, and any administrator interested in quality of service (QoS) and network resource allocation for traffic in the virtualized data center. Readers should have some knowledge of the Cisco Nexus 1000V Series and VMware vSphere 4.1 and higher.

Overview

The Cisco Nexus 1000V Series Switches are edge switches on the virtual access layer and therefore provide a full QoS solution that includes:

- Traffic classification and marking
- Traffic policing (rate limiting)

With Cisco Nexus 1000V Series Switches Release 4.2(1)SV1(4) and higher, virtualization environments can now also take advantage of Class-Based Weighted Fair Queuing for congestion management.

Congestion Management for the Virtualized Cloud Environment

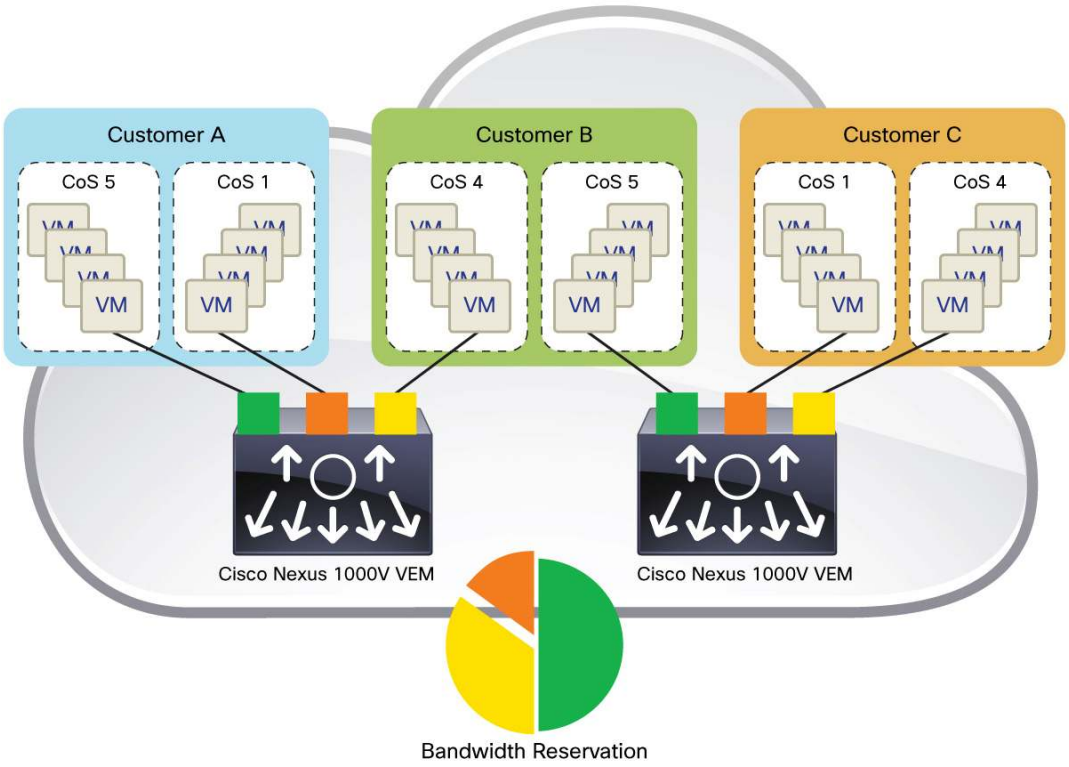
Congestion management is important in virtualized data centers and also for infrastructure in a service cloud environment. In today's enterprise and service provider data centers, servers and network devices often encounter contention with different types of network traffic. Certain applications and services can generate traffic that uses network links, with heavy load either in intermittent bursts or constantly transmitted. This network traffic should be carefully categorized to help ensure that important traffic is not dropped and is properly scheduled out to the network. Queuing is needed for congestion management to reserve bandwidth for many classes of traffic when the physical network links encounter congestion.

Using the Cisco Nexus 1000V Series as virtualized data center switches, along with existing physical infrastructure, an enterprise or service provider can achieve intelligent queuing on a per-hop basis, beginning with the virtual access layer at the hypervisor level.

© 2012 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

Virtual data centers can integrate queuing to provide customers and organizations with the necessary network bandwidth reservations for all traffic that originates from virtual machines, virtual appliances, virtual applications, and critical infrastructure traffic. To meet the demands of cloud providers and enterprise data center customers, the Cisco Nexus 1000V Series can efficiently classify traffic and provide specific, detailed queuing policies for various service levels for virtual machines, such as platinum, gold, bronze, and best-effort classes (Figure 1). Together with traffic marking, this queuing enables virtualized cloud environments to meet service-level agreements (SLAs).

Figure 1. Bandwidth Reservation for Multiple Customers Based on Marking or Protocol



Rate Limiting (Policing) Compared to Bandwidth Reservation (Queuing)

Both rate-limiting and bandwidth reservation are effective tools used in providing QoS. However, both differ in behavior when used on traffic. Rate limiting, also referred to as policing, has a maximum defined value that traffic cannot exceed. When applied to a class of traffic, policing is used to limit (and potentially drop) this traffic based on the configured maximum. For example, if an administrator wants to limit the amount of FTP traffic coming into the switch, a QoS policy can be configured to detect this traffic and then drop it if the configured maximum threshold is met.

Bandwidth reservation, often referred to as queuing, however, provides a minimum defined value that traffic cannot go below after congestion has occurred. This setting provides the capability to guarantee that a certain class of traffic will have adequate resources to continue forwarding out of a device. Bandwidth reservation is the preferred QoS tool when managing communications performance and network congestion.

© 2012 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

Class-Based Weighted Fair Queuing

CBWFQ is a network queuing technique that allows the user to configure custom traffic classes based on various criteria. Each of these classes can be assigned a share of the available bandwidth for a particular link. This queuing is typically performed in the outbound direction of a physical link, with each class having its own queue and specific bandwidth reservation value. CBWFQ is important because it allocates bandwidth to the configured traffic classes when the network link encounters congestion. When congestion occurs without an intelligent queuing mechanism, packets are handled in a First In First Out (FIFO) fashion. This approach can potentially cause any traffic that is currently in transit out of the link to take precedence over more important critical traffic. This approach can also cause subsequent traffic to be dropped with out any chance of proper scheduling. CBWFQ solves these problems by providing a queue for each class of traffic, guaranteeing bandwidth to the most critical traffic when it is needed the most.

How CBWFQ works on the Cisco Nexus 1000V Series

Before queuing can take effect, traffic has to be classified. After the classification is defined, each traffic class will have its own queue with an associated bandwidth percentage assigned to it. The modular QoS command-line interface (MQC) is used to configure the queuing behavior through a queuing policy.

A queuing policy consists of the following components:

- Class map: Used to classify traffic
- Policy map: Used define queuing behavior
- Service policy: Used to apply this policy to a port profile or interface

The Cisco Nexus 1000V Series has eight predefined protocols that can be used to define traffic classes as listed in Table 1.

Table 1. Cisco Nexus 1000V Series Predefined Protocols

Protocol Classes	Description
Cisco Nexus 1000V Series Control	All control traffic that carries synchronization and programming information for the Cisco Nexus 1000V Series
Cisco Nexus 1000V Series Packet	Packet traffic that carries any communication packets that require processing by the supervisor (Cisco Nexus 1000 Series Virtual Supervisor Module [VSM])
Cisco Nexus 1000V Series Management	Management-plane traffic for the Cisco Nexus 1000V Series to and from the mgmt0 interface (includes VMware vCenter communications)
VMware vMotion	All VMware vMotion traffic needed for live migration of virtual machines between hosts
VMware Fault-Tolerance Logging	Replication data needed to achieve fault tolerance for virtual machines
VMware Management	Management traffic associated with the service console and the management-enabled vmk interface
VMware Network File System (NFS) Storage	NFS I/O packets
VMware Small Computer System Interface over IP (iSCSI) Storage	iSCSI I/O packets

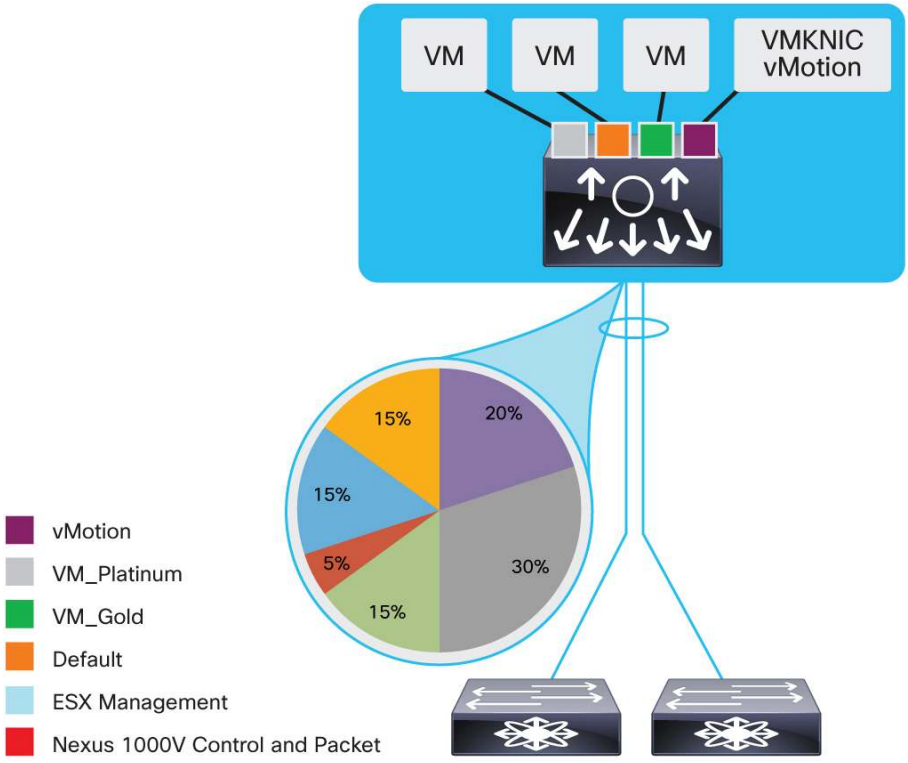
Apart from these predefined traffic classes, the Cisco Nexus 1000V Series offers the capability to classify traffic based on the class-of-service (CoS) marking in the priority header of an IEEE 802.1Q tagged Ethernet frame. Using a QoS-type class map, the Cisco Nexus 1000V Series can categorize incoming frames for internal switch processing and also mark outgoing frames to maintain this marking on the physical infrastructure. A queuing class map is used to put these frames in the proper traffic category to prepare for queuing within the switch.

© 2012 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

CBWFQ is effectively applied in the outbound direction of the physical interface of the switch. In a VMware environment, these physical Ethernet interfaces are referred to as the physical network interface card (pNIC) on the VMware ESX or ESXi server. Regardless of the number of physical interfaces (whether you are using a PortChannel or a individual link), queuing is performed on each discrete link. So if a Cisco Nexus 1000 Series Virtual Ethernet Module (VEM) is using a PortChannel as its uplink with two physical links, the queuing policy affects the bandwidth of each PortChannel member link independently.

As Figure 2 shows, when CBWFQ is used, traffic will be reserved for a particular class of traffic on a per-link basis, so if a policy that specifies reserved bandwidth for vMotion, VM_Platinum, VM_Gold, etc. traffic is applied to a PortChannel, these values are guaranteed on a per-link basis.

Figure 2. Bandwidth Reserved on Individual Link

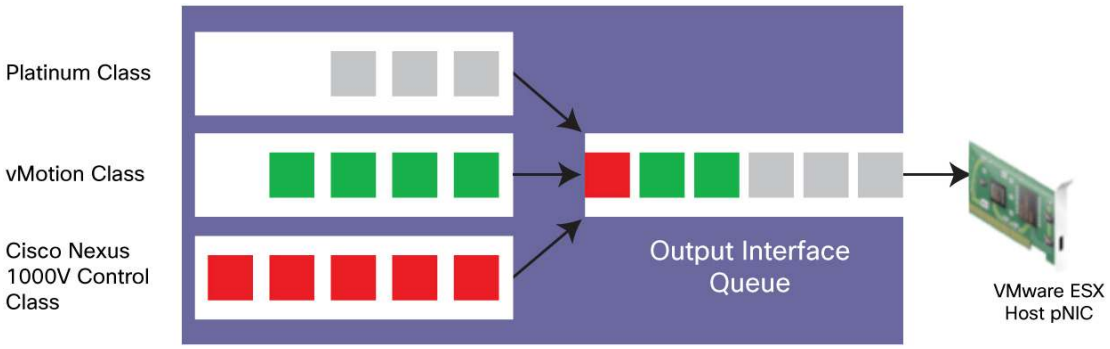


On the Cisco Nexus 1000V Series, a queue is created after the user defines a policy map for a particular traffic class. The queue is actually constructed on the VEM on a per-physical interface basis. On the basis of the configured bandwidth percentage, each class will have the associated resources allocated to it.

The bandwidth percentage in effect determines how the packet is scheduled from its associated queue to the output interface queue that belongs to a pNIC on the VMware ESX or ESXi host. The packets are then transmitted to the physical network from this output interface queue. Therefore, in Figure 3, the Platinum VM class has more bandwidth reserved than both the vMotion and N1KV Control classes. If these three classes were the only ones configured as part of the queuing policy, then any remaining traffic would simply be put in a default class. This default class would use the remaining available bandwidth, indicating best-effort delivery.

© 2012 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

Figure 3. Traffic Class Output Queuing



Important Infrastructure Traffic in Virtualized Environments

Some infrastructure traffic, such as VMware vMotion, VMware management, and Cisco Nexus 1000V control, packet and management traffic should always have bandwidth reserved for times of congestion. Because congestion can cause this traffic to be dropped, certain features and functions can become degraded or stop working completely. In the case of concurrent VMware vMotion (four instances for 1-Gbps and eight instances for 10-Gbps environments), live migration can fail without the proper resources.

Likewise, during times of congestion, if the Cisco Nexus 1000V Series control, packet, and management traffic does not have the proper resources allocated, VEM-to-VSM communication can stop working, which can affect overall system performance. Cisco highly recommends allocating bandwidth for traffic related to the Cisco Nexus 1000V Series as part of every deployment.

Additional Guidelines and Considerations

- CBWFQ is supported only with VMware ESX and ESXi 4.1 and later. The queuing feature is available only on VMware vSphere 4.1 and later deployments.
- When you specify the bandwidth percentage in a policy map, the total value should add up to 100 percent. To efficiently allocate queuing resources, Cisco recommends that the sum of bandwidth percentages for all classes in a policy map, including a user-defined default class, should equal 100 percent. The configuration example at the end of this document can be used as a reference.
- The total number of active queues per VEM is 16. Therefore, there can only be 16 classes of traffic in use at any given time. This number does allow the use of all classification types (for example, eight CoS values and eight traffic protocols).
- Only one queuing policy per VEM is supported on a single interface. A single service policy can be assigned to a single interface per host. This interface can be a discrete physical interface or a PortChannel. For example, if a host has two 10 Gigabit Ethernet interfaces, the service policy can be applied to both if they are members of a PortChannel or to only one if a PortChannel is not used.

© 2012 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

Configuring CBWFQ on the Cisco Nexus 1000V Series

This example uses Figure 2 as a reference topology.

Step 1. Create the queuing class map and define the traffic class.

```
class-map type queuing [match-all | match-any] {class-map-name}
   match [cos | protocol] {cos-value | protocol-name}
```

Example:

```
class-map type queuing match-any n1kv_control_packet_class
  match protocol n1k_control
  match protocol n1k_packet
  match protocol n1k_mgmt
class-map type queuing match-all vm_platinum_class
  match cos 5
class-map type queuing match-all vm_gold_class
  match cos 4
class-map type queuing match-all vmotion_class
  match protocol vmw_vmotion
class-map type queuing match-all vmw_mgmt_class
  match protocol vmw_mgmt
```

Step 2. Create the queuing policy map to define the queue.

```
policy-map type queuing [match-first] {policy-map-name}
   class type queuing {class-map-name}
   bandwidth percent {value}
   [queue-limit] {value} packets
```

Example:

```
   policy-map type queuing uplink_queue_policy
     class type queuing n1kv_control_packet_class
       bandwidth percent 15
    class type queuing vm_platinum_class
      bandwidth percent 30
 class type queuing vm_gold_class
   bandwidth percent 15
 class type queuing vmotion_class
   bandwidth percent 20
 class type queuing vmw_mgmt_class
   bandwidth percent 15
```

Step 3. Apply the queuing service policy to the uplink port profile outbound.

The service policy can also be applied to physical Ethernet and PortChannel interfaces.

```
port-profile type ethernet {port-profile-name}
```

© 2012 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

```
service-policy type queuing output {policy-map-name}
```

Example:

```
port-profile type ethernet uplink
  service-policy type queuing output uplink_queue_policy
```

Verifying CBWFQ on the Cisco Nexus 1000V Series

Correct behavior of a queuing policy can be confirmed by verifying the control-plane and data-plane (optional) components.

Verify the Control Plane

Step 1. Verify the running policy.

```
show policy-map type [qos | queuing] {policy-name}
```

Example:

```
show policy-map type queuing uplink_queue_policy

  Type queuing policy-maps
  ========================

  policy-map type queuing uplink_queue_policy
    class type queuing n1kv_control_packet_class
      bandwidth percent 15
    class type queuing vm_platinum_class
      bandwidth percent 35
    class type queuing vm_gold_class
      bandwidth percent 15
    class type queuing vmotion_class
      bandwidth percent 20
    class type queuing vmw_mgmt_class
      bandwidth percent 15
```

Step 2. Verify that the policy is applied to the interface or PortChannel.

```
show policy-map interface {interface-name}
```

Example:

```
show policy-map interface port-channel 1

Global statistics status :   enabled

port-channel1

  Service-policy (queuing) output:   uplink_queue_policy
```

© 2012 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

```
policy statistics status:   enabled

Class-map (queuing):   n1kv_control_packet_class (match-any)
  Match: protocol n1k_control
  Match: protocol n1k_packet
  Match: protocol n1k_mgmt
  bandwidth percent 15
  queue dropped pkts : 0

Class-map (queuing):   vm_platinum_class (match-all)
  Match: cos 5
  bandwidth percent 30
  queue dropped pkts : 0

Class-map (queuing):   vm_gold_class (match-all)
  Match: cos 4
  bandwidth percent 15
  queue dropped pkts : 0

Class-map (queuing):   vmotion_class (match-all)
  Match: protocol vmw_vmotion
  bandwidth percent 20
  queue dropped pkts : 0

Class-map (queuing):   vmw_mgmt_class (match-all)
  Match: protocol vmw_mgmt
  bandwidth percent 15
  queue dropped pkts : 0
```

Verify the Data Plane (Optional)

Step 1. On a per-module basis, verify that the queues were constructed as expected.

`module vem {`**`module-number`**`} execute vemcmd show qos node`

The output of this command lists all the defined traffic classes, each with a unique identifier (called the node ID).

Example:

```
module vem 4 execute vemcmd show qos node
nodeid   type     details
-------- -------- --------
       0    class op_DEFAULT
       1    class    op_OR
                   protocol
                  n1k_cntrl
                   protocol
```

© 2012 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

```
                n1k pkt
               protocol
               n1k mgmt
       2    class   op_AND
                queuing
       3    class   op_AND
                queuing
       4    class   op_AND
              protocol
           vmw vmotion
       5    class   op_AND
              protocol
              vmw mgmt
```

Step 2. Verify that traffic classes are being matched outbound on the module uplink.

```
module vem {module-number} execute vemcmd show qos {uplink LTL}
```

Note that the local target logic (LTL) is a local identifier that pertains to each interface. In this case, you are looking for the PortChannel interface on the module. You can determine which LTL belongs to each interface by running the following command beforehand: **module vem {module-number} execute vemcmd show port.**

The output of the following command lists all the defined traffic classes that will be queued outbound in the **Egress_q class** column. This class number belongs to the node ID for that class, which can be correlated with the command in Step 1 to view the name of each class. For instance, you can see that **Egress_q class 4** pertains to the **vmw vmotion** protocol.

Example:

```
module vem 4 execute vemcmd show qos pinst 305

id       type
-------- --------
     305 Egress_q
            class        bytes matched         pkts matched
            -------- -------------------- --------------------
                  1             42657350               164666

                  2              9864507                22202

                  3               150048                  508

                  4          10893643289               173391

                  5                 4574                   63

            filter id 72352191
```

© 2012 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

```
                40535036                155536
filter id 72352192
                2122314                  9130
filter id 72352193
                      0                     0
```

Conclusion

As virtualized data centers and cloud environments grow, network congestion and application degradation may occur. Intelligent queuing is needed to sustain service levels and retain the desired performance to allocate resources for important communications. The Cisco Nexus 1000V Series meets these challenges with CBWFQ. CBWFQ provides the capability to define flexible and specific queuing policies for important infrastructure protocols as well as various levels of virtual machine traffic.

For More Information

Please see the following documents and guides for further information:

- Cisco Nexus 1000V Series Deployment Guide
- Cisco Nexus 1000V Series QoS Configuration Guide
- Cisco® Data Center and Multi-Tenancy Reference Architecture Guide
- Cisco IOS® Class-Based Weighted Fair Queuing

For general Cisco Nexus 1000V Series information, please visit http://www.cisco.com/go/nexus1000v.

Appendix

The following is the complete configuration related to the QoS content in this document:

```
version 4.2(1)SV1(4)
class-map type queuing match-all vm_gold_class
  match cos 4
class-map type queuing match-all vmotion_class
  match protocol vmw_vmotion
class-map type queuing match-all vmw_mgmt_class
  match protocol vmw_mgmt
class-map type queuing match-all vm_platinum_class
  match cos 5
class-map type queuing match-any n1kv_control_packet_class
  match protocol n1k_control
  match protocol n1k_packet
  match protocol n1k_mgmt
policy-map type qos gold_in_mark
  class class-default
    set cos 4
policy-map type qos platinum_in_mark
  class class-default
    set cos 5
policy-map type queuing uplink_queue_policy
```

© 2012 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

```
  class type queuing n1kv_control_packet_class
    bandwidth percent 15
  class type queuing vm_platinum_class
    bandwidth percent 35
  class type queuing vm_gold_class
    bandwidth percent 15
  class type queuing vmotion_class
    bandwidth percent 20
  class type queuing vmw_mgmt_class
    bandwidth percent 15

port-profile type ethernet uplink
  service-policy type queuing output uplink_queue_policy

port-profile type vethernet VM_VLAN170_ORG_A
  service-policy type qos input platinum_in_mark

port-profile type vethernet vApp_VLAN880
  service-policy type qos input gold_in_mark
```

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Printed in USA C11-704041-00 04/12

© 2012 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.