Network Design

This chapter describes the different network topologies for the Cisco TV VDS, the different network connections of the VDS servers, the VDS workflow, and network configuration considerations. The topics covered in this chapter include:

Overview

The TV VDS enables cable operators and multiple service operators (MSOs) to offer VOD and MediaX services to consumer customers over their existing hybrid fiber coaxial (HFC) network, with existing next-generation digital STBs. The TV VDS solution uses a gigabit Ethernet (GE) transport network from the headend to the distribution hub, where the HFC network terminates.

TV VDS grows seamlessly from a single server implementation to multiple servers. As growth continues, TV VDS allows operators to install distributed servers to address concentrations of subscribers while leaving content ingest and management centralized.

Stream Groups can be distributed close to the subscriber and linked back to the central Vault locations by way of the Cisco Cache Control Protocol (CCP). Cisco CCP automatically ensures that any new content that is required by a customer edge device is transferred within a maximum of a 250-millisecond delay to the appropriate edge location; as a result, all content appears local to each edge site, even though most content is stored at the central Vault location.

The TV VDS offers different configurations with regards to network topology, business management systems (BMSs), and streaming modes.

VDS with Vaults and Streamers

In a TV VDS with Vaults and Streamers, MPEG-2 transport stream (TS) video is stored on the Vaults with the associated trick-mode files. Content is transported from the Vaults to the Streamers as needed, by using CCP over gigabit Ethernet networks. Content is sent unicast from the Streamers and delivered to the quadrature amplitude modulation (QAM) devices over gigabit Ethernet or asynchronous serial interface (ASI), and then is modulated onto the HFC plant to the subscriber set-top box (STB) for viewing.

VDS with ISVs

For the smallest networks, Cisco packages the VDS in a single server, the Integrated Streamer-Vault (ISV), offering a solution for VOD services with large content libraries but small stream counts.

In a TV VDS with ISVs, MPEG-2 TS video is stored on the ISVs with the associated trick-mode files. Content is sent unicast from the ISVs and delivered to the QAM devices over a gigabit Ethernet network, and then is modulated onto the HFC plant to the subscriber STB for viewing.

VDS with Caching Nodes

For larger networks, Cisco offers the VDS with Caching Nodes in the Virtual Video Infrastructure (VVI). In a VVI, Caching Nodes are the intermediary fill source for Streamers, which removes a large portion of the distribution traffic from the Vaults.

In a TV VVI, MPEG-2 TS video is stored on the Vaults with the associated trick-mode files. Content is transported from the Vaults to the Caching Nodes as needed, by using CCP over gigabit Ethernet networks. Content is distributed from the Caching Nodes to the Streamers as needed, by using CCP over gigabit Ethernet networks, or by using HTTP over gigabit Ethernet networks. Content is sent unicast from the Streamers and delivered to the QAM devices over a gigabit Ethernet network, and then is modulated onto the HFC plant to the subscriber STB for viewing.

TV VDS and VVI Topologies

The TV VDS (using Vaults and Streamers, or ISVs) and the TV VVI (using Vaults, Caching Nodes, and Streamers), supports centralized, decentralized, and hybrid gigabit Ethernet network designs. Because the use of Vaults and Streamers separates storage from streaming, streaming requirements can be satisfied on an as-needed basis and the streaming can be centralized or distributed among multiple locations. Caching Nodes separate the ingest and storage of content from the distribution of content, offering greater flexibility and network efficiency.

The TV VDS topology and TV VVI topology can change with the evolving needs of the system operator. If the need to decentralize becomes evident, you can move the Streamers or Vaults to remote hubs without disrupting service. The VVI offers additional flexibility in designing your network. Vaults can be centrally located at a national network, and content may be classified by market (city, state, or a broader region) depending on the AMS or BMS used. Caching Nodes can be located centrally, or distributed closer to the regional networks where the Streamers are located. Using Caching Nodes in the network design takes the distribution traffic off the network backbone.

Caution All Cisco servers are connected through a switch. Because all Vaults, CCP Streamers, and Caching Nodes in the same array exchange heartbeat messages through the cache interfaces, it is important to ensure there is enough bandwidth among switches involved in delivering cache traffic, as well as to support the same aggregated amount of traffic on all cache interfaces.

Note When using ISVs, with the Vault and Streamer functions contained in one server, the only topology possible is centralized.

Centralized Topology

In a centralized topology, all VDS servers are located in either a single video headend or a remote hub. This is the right solution for certain situations, for instance, very small starting systems or where a large amount of bandwidth is available. A centralized topology has advantages in reducing operational cost by placing equipment in one physical location. Figure 2-1 illustrates the centralized topology for Vaults and Streamers.

Figure 2-1 Centralized Topology with Vaults and Streamers

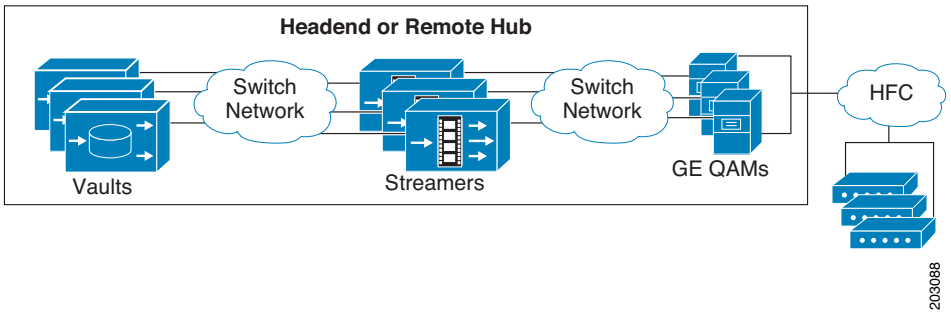


Figure 2-2 illustrates the centralized topology for ISVs.

Figure 2-2 Centralized Topology with ISVs

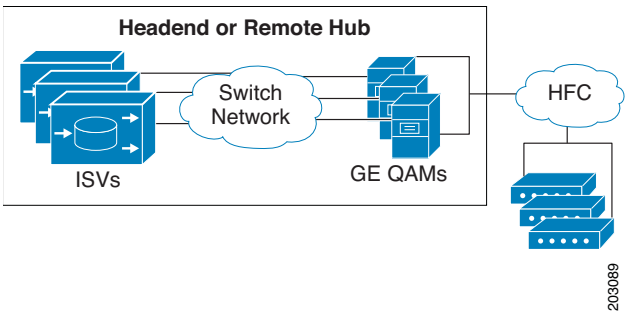
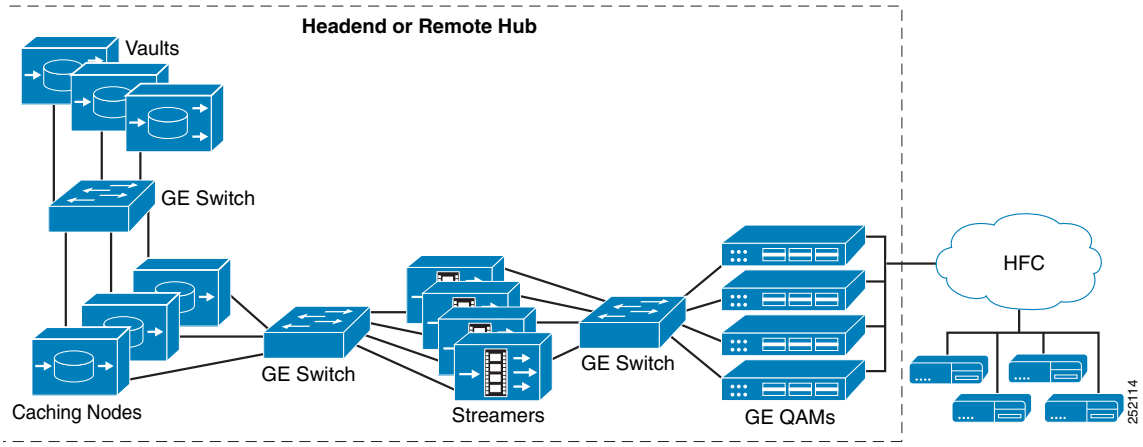


Figure 2-3 illustrates the centralized topology for a VVI.

Figure 2-3 Centralized Topology with Caching Nodes



Decentralized Topology

The decentralized topology is a hub-and-spoke topology between the headend site and multiple hub sites, where the Vaults are located at the headend and the Streamers are in the hub sites. For a VVI, a decentralized topology provides a three-tiered approach by having the Vaults located in the headend, the Caching Nodes in intermediary sites, and the Streamers in the hub sites. The decentralized topology works well for distributing Stream Groups close to subscribers. A decentralized topology has advantages in reducing the amount of long-haul fiber transport bandwidth needed—typically by a factor of ten or better. Figure 2-4 illustrates the decentralized topology.

Figure 2-4 Decentralized Topology

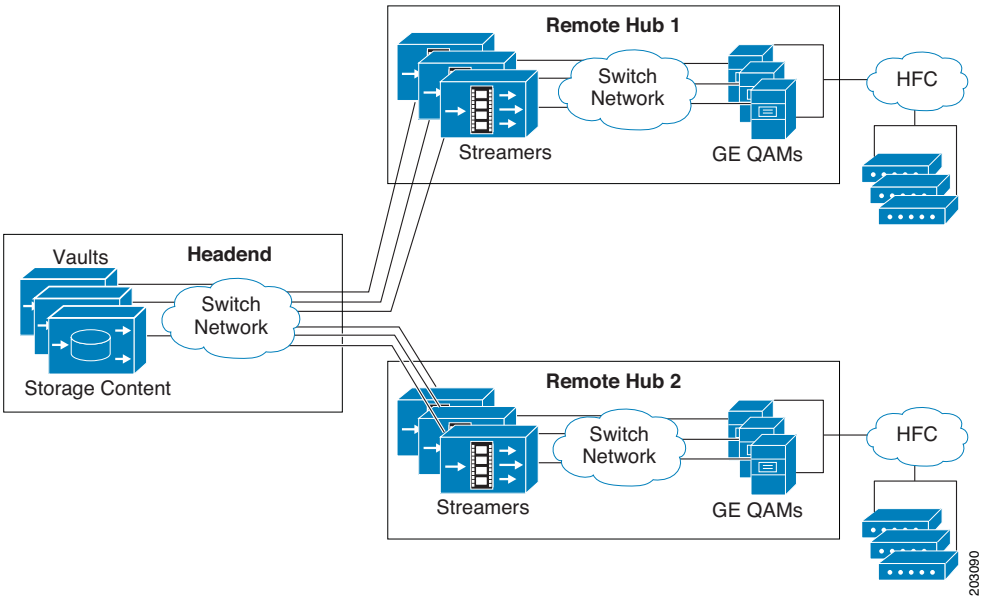
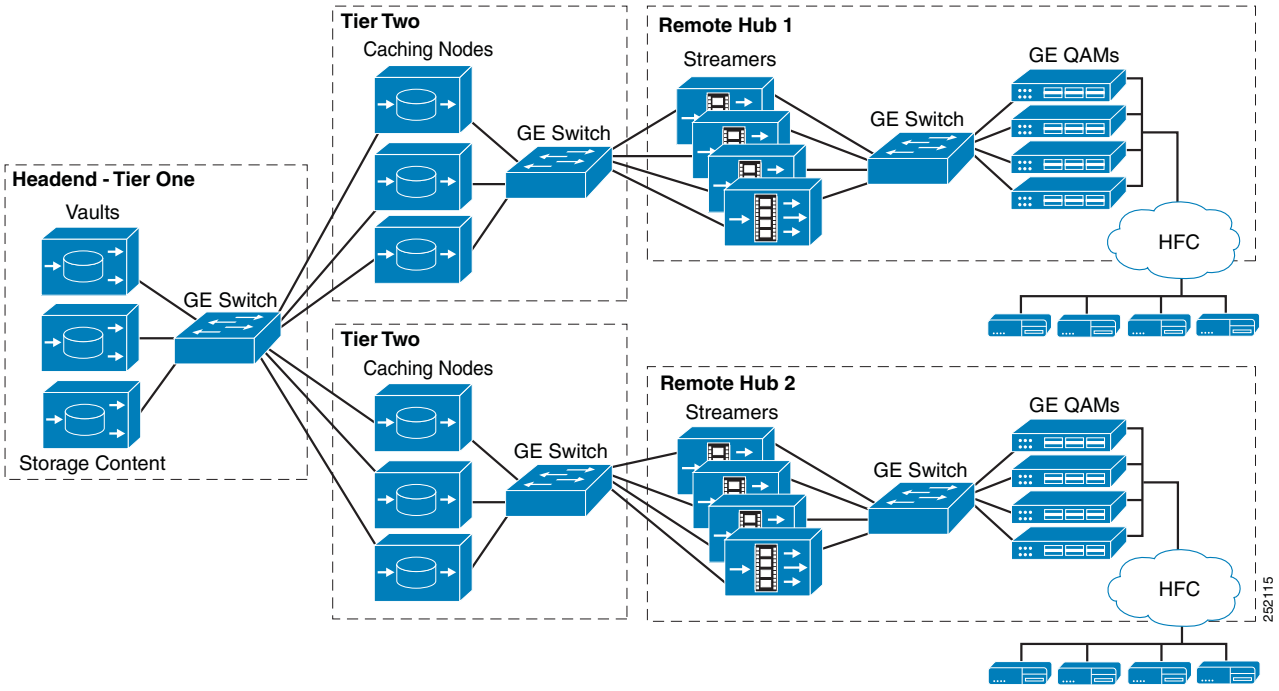


Figure 2-5 illustrates the decentralized topology with Caching Nodes.

Figure 2-5 Decentralized Topology with Caching Nodes



Hybrid Topology

In a hybrid topology, the Vault servers and backup Streamer servers are located at the headend, with the active Streamers at a remote hub site. If the remote hub site goes down, the Streamers at the headend take over. A hybrid topology blends the advantages of centralized and decentralized topologies that is based on needs of the system implemented. Figure 2-6 illustrates the hybrid topology.

Figure 2-6 Hybrid Topology

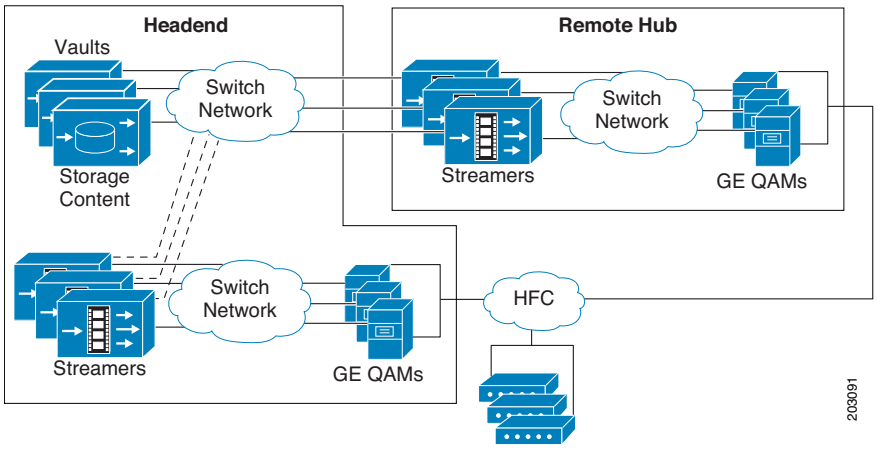
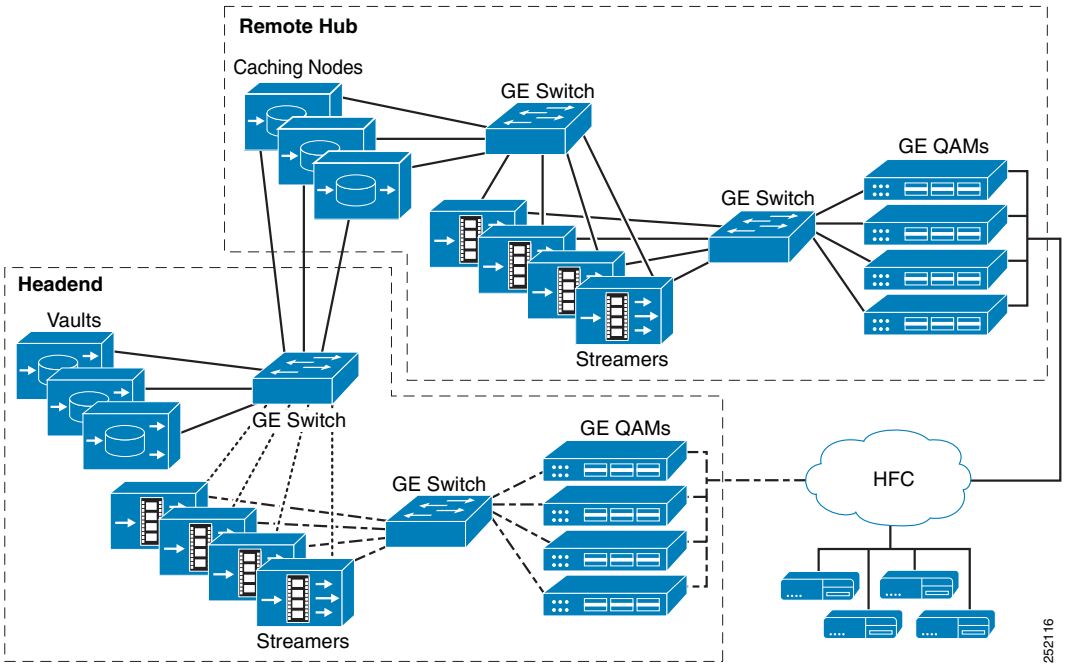


Figure 2-7 illustrates the hybrid topology with Caching Nodes.

Figure 2-7 Hybrid Topology with Caching Nodes



TV VVI Management

The TV VVI offers two types of management, centralized and split-domain.

In a VDS, Streamers cannot communicate with Streamers in other groups. In a VVI, Streamers in other groups can communicate with each other on an as-needed basis.

All Vaults, Streamers, and Caching Nodes are identified by an array ID, a group ID, and a server ID. In the CDSM GUI, the array ID identifies servers that are part of the same system, the group ID identifies servers that are part of the same group (Vault Group, Cache Group, and Stream Group), and the server ID is a unique number that identifies the server. Table 2-1 lists the CDSM GUI ID names and maps them to the CServer names in the setupfile and .arroyorc files.

Table 2-1 ID Names in the CDSM GUI and CServer Files

CDSM GUI ID Name	CServer Files ID Name
Array ID on the Array Name page	groupid
Group ID on the Server-Level pages	groupid
Stream Group ID on the Server Setup page	arrayid
Cache Group ID on the Server Setup page	arrayid
Vault Group ID on the Server Setup page	arrayid
Stream Group ID on the Configuration Generator page	arrayid

Centralized Management

Centralized management uses one Virtual Video Infrastructure Manager (VVIM) to manage the Vaults, Caching Nodes, and Streamers in a VVI.

Split-Domain Management

Split-domain management uses one VVIM to manage the domain of Vaults and Caching Nodes, and separate managers, the Stream Managers, to manage each domain of Streamers.

In a split-domain VVI that uses HTTP for communication between the Caching Nodes and Streamers, the databases for each domain are separate. The information stored in each database is not shared with the servers in the other domains. The Stream Managers communicate with the VVIM over port 80. If port 80 is not open for communication, the managers cannot communicate with each other and configuration settings need to be uploaded to the Stream Managers from information downloaded from the VVIM.

In a split-domain VVI that uses CCP for communication between the Caching Nodes and Streamers, the database is replicated among all servers in the Vault/Cache domain and the Stream domains. Because the VVI allows intercommunication among different Cache Groups and Stream Groups when CCP Streamers are used, the server ID and group ID must be unique across the system. The Stream Managers communicate with the VVIM by using the database replication.

Note Split-domain management is supported in an RTSP environment, and an ISA environment with the Content Storage feature and CCP Streamers.

VDS Workflow

Content is ingested and stored in the Vault array. The Vault array can consist of two Vault Groups, which in turn consists of two or more Vaults that are either colocated or distributed to multiple locations across an Ethernet network. Content ingest is initiated by the backoffice based on a subscriber request, and based on schedule or barker channel content. Manual ingest, which is operator initiated, is also offered as an optional feature.

Note The ability to differentiate between a DVD asset and a video asset to support ingest, trick-play creation, and streaming of content files as large as 120 GB is supported. The content files could span multiple days.

As the content is ingested into the Vault, any necessary trick-mode files are created. The content and trick-mode files are then mirrored within the same Vault or across the Vault array. The replication of content allows for data recovery should a Vault undergo a failure.

Content is delivered from the Vault array to the Stream Group in response to cache-fill calls from the Streamers. Content is also distributed across the network in response to scheduled or barker stream content fulfillment.

As Streamers need to fill content, they issue locate requests to the Vaults for the specific content. The Streamer makes a decision on which Vault to pull content from based on the responses. The process of determining where to pull content from includes memory capacity and disk capacity of the Vault, as well as network capacity.

If a VVI is deployed, content is delivered from the Vault Group to the Cache Group in response to cache-fill calls from the Streamers. The Caching Nodes are explained in more detail in the "Caching Node Workflow" section on page 2-12.

Within the Streamer array are one or more Stream Groups. The following section describes how the Stream Groups deliver streams to the subscriber STBs.

Note All servers can be on different subnetworks. However, because of backoffice restrictions, the externalized IP address is constrained to migrate among servers on the same subnetwork. This means the Content Store server in an Interactive Services Architecture (ISA) environment can migrate only among Vaults that are on the same subnet, and the Setup and Control servers can migrate only among Streamers on the same subnet.

Popularity-Based Caching

Popularity-based caching reduces the write rate to the storage devices on the Streamer and Caching Node while maintaining the best possible cache-hit rate on the available storage.

To control peak and average write rates to cache (flash or disk storage), the algorithm that determines when content is written to cache is changed so that only content that is likely to be accessed most often is cached. Content is only cached if it is more popular than the least popular content that is currently cached. Otherwise, the content is transmitted from the Vaults to the end-users by way of the cut-through mode, where content is temporarily stored in the Streamer and Caching Node RAM without ever writing it to disk or flash storage, and then streamed directly from the Streamer's RAM to the end-user. When cache space is needed, the least popular content is evicted from cache first.

The write rate for caching content is determined by the rate at which previously popular content becomes less popular to the point where it no longer makes sense to keep it in cache, and previously unpopular content becomes more popular to the point where it does make sense to keep it in cache. Content popularity is measured by the time-decaying average of the number of play requests on each Global Object Identifier (GOID).

Previously, all content was written to cache (except when overloaded) and the Least Recently Used (LRU) content was evicted first.

With the Popularity-Based Caching feature, only popular content is written to cache and the least popular content is evicted first.

Bandwidth Manager for Thin Pipe

The bandwidth manager controls the traffic leaving the site to any other site and queries all the VDS servers in the site for the thin pipe mapping configuration of each VDS server. One server in the site is elected as the bandwidth manager for all servers in the site. A site is defined by the Site Setup page, which associates groups with a site. Initially, the bandwidth manager allocates bandwidths of whatever the VDS servers have already committed, provided the committed bandwidths are within the pipe bandwidth limits; otherwise, the bandwidth manager allocates a percentage of what is committed. After the initial allocation, the bandwidth manager distributes the bandwidth equally among all the remaining VDS servers in the site.

Each VDS server in each group reports the bandwidth each one is using to the bandwidth manager every ten seconds. The bandwidth threshold for each server has an upper limit of 90 percent and a lower limit of 5 percent. If a server reaches either limit, the server reports this to the bandwidth manager

immediately, and does not wait for the ten-second report interval. For example, if the server is given 100 Mbps and the streams that were just started uses 90 Mbps, the upper threshold limit has been reached and the server asks the bandwidth manager for more bandwidth.

A separate entry is maintained for each thin pipe with a list of servers that have the same thin pipe configuration. Servers that belong to the same thin pipe are added and removed as they become reachable or unreachable.

The bandwidth manager service runs on each server in either the primary mode or the passive mode. The one server at the site that is running the primary mode is selected through a discovery mechanism. The primary bandwidth manager maintains all the thin pipes and associated server structures. If the server running the primary bandwidth manager fails or loses connectivity, the newly elected bandwidth manager takes over and when the old primary bandwidth manager becomes available again and connectivity is restored, the thin pipe information and structures are deleted from the old primary.

All bandwidth manager messages are logged in the bwm.log file. The following logging levels are defined (default level is Information):

- Critical
- Error
- Warning
- Information
- Debug
- Debug Verbose

Streamer Workflow

A Stream Group is a configurable group of Streamers that are designated to serve specified QAM devices, and subsequently, specific service groups. From a session setup and control perspective, there are three logical types of servers in a Stream Group:

- Setup server
- Control server
- Play server
- Remote Setup and Control server

The Setup and Control servers have both a primary and a backup server. The primary server services all messages, while the backup server simply maintains states. If a primary server is unreachable, the backup server takes over control and creates another backup server. Thus, there is always a primary and backup pair of servers for setup and control. The Play server does not have a backup server. However, the Control server selects a new Play server in the event of a failure of the existing Play server.

Note The ability to have both a primary and backup server depends on the number of Streamers in the Stream Group.

The Setup and Control server IP addresses are configurable. For an ISA environment, the Setup IP address is the same as the Stream Master IP address. For RTSP, the Setup server and Control server must be the same server. For both ISA and RTSP environments, the Stream Service selects a Streamer in the Stream Group to be the Setup server, and another Streamer (sometimes the same Streamer) to be the Control server.

Setup Server

A Streamer designated as the Setup server interfaces with the backoffice and forwards the setup messages to the appropriate Stream Group that is assigned to the destination service group. One Streamer in the Stream Group that is colocated with the backoffice server is assigned as the primary Setup server. The Setup server receives the setup request from the backoffice and maps the service group.

The Setup server returns the IP address of the Control server, and the STB issues subsequent control messages to this IP address.

Control Server

The Control server assigns requests to specific Streamers and dynamically migrates streams between Streamers based upon changes in stream states (for example, content splice boundaries, maintenance trickle down, or server failures). One server in the Stream Group is assigned as the primary Control server. The Control server runs the Lightweight Stream Control Protocol (LSCP) proxy in an ISA environment and the Real-Time Streaming Protocol (RTSP) proxy in an RTSP environment.

For each and every setup message received from the backoffice, a CCP message is generated and sent to the Control server. In the initial setup request, the Control server receives the setup parameters but does not choose a Play server. After a control message is received from the STB, the Control server gets performance information (for example, server load) from the potential Play servers within the Stream Group and sends a CCP message to the best candidate. Subsequent control messages, whether from the STB or from the Setup server, are forwarded to the chosen Play server.

Play Server

The Play server is the Streamer that is assigned to play the stream. This Streamer acquires the content, whether in RAM, a local disk, or a Vault, and ensures guaranteed service delivery of the stream. Every Streamer in a Stream Group is a possible candidate to be the Play server.

Remote Setup and Control Server

The Remote Setup and Control Server Support feature allows the Setup and Control servers of the Streamers to be placed in a different location than the Play servers of the Streamers. All control traffic (setup and control) goes to one Streamer (Setup and Control server), and all video data traffic is served from the Streamers designated as the Play servers.

As part of this feature, there is never just one stream transmitted on a single Ethernet interface. There is always at least two active streams transmitted on an interface.

Note The Remote Setup and Control Server Support feature is supported on a Virtual Video Infrastructure (VVI) with split-domain management in an ISA environment and Content Storage configured as either Shared or Distributed.

The VOD Error Repair feature is not supported with the Remote Setup and Control Server Support feature.

Remote Setup and Control

The setup and control traffic between the set-top boxes (STBs) and VDS is sent to a location that is separate from the location where the video data streams originate. The Session Traversal Utilities for NAT (STUN) traffic is structured so that it is sent to the Setup server instead of the Play server. The data path through the end-user's NAT device complies with RFC-5389.

Stream to Interface Relationship

The Remote Setup and Control Server Support feature requires at least two stream requests before sending the first data stream, and makes sure there are at least two data streams on an active Streamer interface at all times. The Control server makes sure there are at least two streams on an active Streamer, and the active Streamer makes sure there are at least two streams on an active stream interface.

If the first play request on a Play server reaches the session timeout period before a second play request is received on that Play server, the first session fails and the stream is sent back to the Control server for relocation. If there are only two sessions created on the VDS and one session is destroyed or completes, the remaining session is destroyed.

Global Source Address

The Remote Setup and Control Server Support feature introduces the *Global Source Address*, which is the IP address and associated port number that is used by all Play servers for transmitting stream data. The Global Source Address is defined on all Streamers (Setup, Control, and Play).

The Global Source Address is defined in the setupfile on all Streamers (Setup and Control servers, and Play servers). This address is hosted on the primary Setup server and is managed in a fault-tolerant manner; that is, it moves from interface to interface as needed if an interface fails, and it transitions to a new primary Setup server if the original primary Setup server becomes unreachable.

Each stream interface on the Streamers continues to have a unique IP address so that diagnostic packets (and cache- fill traffic if configured as a stream/cache interface) can be sent and received on those interfaces. However, all stream data packets are sent using the Global Source Address as the source.

The Control server uses the Global Source Address as the stream source address in the reply to the NAT Setup request from the STB. This occurs whenever the Streamer Play server indicates that a remote STUN handshake is needed.

The Global Source Address has the following benefits:

- Mid-session STUN handshakes are not needed, which eliminates the overhead and associated temporary black-screens that occur on STBs when STUN handshakes happen mid-session.
- Streams can be moved more easily for load-balancing purposes. A Streamer can move a stream from one interface to another without involving the Control server, Setup server, or STB. A Control server can move a stream from one Play server to another without involving the Setup server or STB.
- Address management on the Setup server and on the network is simplified. There is only one stream source address that needs to be hosted on the Setup server, and there is only one routing setup in the network configuration.

The following additional information should be considered when configuring the Remote Setup and Control Server Support feature:

- To trace the source of a stream, use the stream session ID along with the associated log files on the Streamers acting as the Play server, as well as the Streamers acting as the primary Setup server and primary Control server. Other diagnostics such as the **ping** command can still use the unique IP address of each stream interface.
- Additional router configuration may be necessary to ensure that the Global Source Address s hosted on the Setup server is used for inbound traffic and that packets sent to that address are never sent to the Play servers.

Caching Node Workflow

A Cache Group is a configurable group of Caching Nodes that serve content to specified Stream Groups. When a content request is received by a Streamer, the Streamer first checks to see if the content is stored locally, which includes DRAM, disk cache, and Streamers in the same Stream Group. Content on the Streamers is always the most popular content, so user requests are generally served from local storage.

Streamers send cache-fill calls to remote servers for content that is not found locally. The remote servers can be Streamers in other Stream Groups, Caching Nodes in Cache Groups, or Vaults in Vault Groups (Vault Groups must be enabled). The cache-fill source selected, whether another Streamer, Caching Node, or Vault, is based on the network capacity and fill-source capacity (disk and memory), as well as on the preference configured for that group of servers. Caching Nodes could respond to the request with a message stating the content is not currently cached, but there are other fill sources the Caching Nodes can contact (Caching Nodes in other Cache groups, and Vaults).

The Caching Nodes use CCP to communicate with the Vaults, and use either CCP or HTTP to communicate with Streamers.

Note ISA environments support only CCP, while RTSP environments support only HTTP for VVI.

HTTP Streamers

HTTP can be used for communication between the Caching Nodes and the Streamers. The HTTP Streamer communicates with a proxy for locating a fill source and pulling content.

A locate service serves as a proxy for a group of Caching Nodes and Vaults. The service is accessed through a highly available virtual IP address hosted by the Caching Node. The virtual IP address is bound to a fill port (Locate Port).

HTTP Streamers request content by HTTP GET requests to the proxy service (the server with the locate service). The proxy server checks its own storage and peer fill sources (servers in the same group) for the content using extended-CCP. If the content is found, the best source is chosen based on capacity and a redirect response is sent to the chosen server. If the content is not found, a cache-fill request is sent to the remote servers.

After the best server is chosen to send the content to the HTTP Streamer, a single cache-fill port on that server is chosen for the HTTP transfer of the content. This is different from CCP transfers, which could potentially use all cache-fill ports to deliver the content.

HTTP Locate Port

With respect to resiliency, the Locate Port service is similar to the Setup and Control servers. The primary server of the Locate Port service has the locate port IP address bound to an interface. The backup server becomes the primary if the primary fails.

Peer Caching Nodes advertise among themselves about the ability to host the HTTP Locate Port service; this includes primary, backup, available, and not usable states. Available means the Caching Node can be either a primary or backup if needed. Not usable means that the server cannot host the service; for the HTTP Locate Port, this typically means that there are no usable network ports for the service.

A dedicated network port on the Caching Node is used solely for the HTTP Locate Port service. The primary server determines service availability based on the link status of the dedicated network port. Failover of the service occurs if the network port loses link status. A reestablished link results in the server becoming available.

CCP Streamers

The CCP Streamers use CCP to communicate with the Caching Nodes. They do not use the proxy address. CCP Streamers load-balance locate requests across fill sources.

The Streamer or Caching Node sends a locate-and-request message from the proxy server. The Proxy server sends a message to the best source to fill the request.

Streamers or Caching Nodes needing content first query peer sources (servers within the same group). Streamers also query local Streamers, if the content is not found, then a request to the remote sources is sent. Remote sources are queried based on a preference list. Sources are grouped and preferences are assigned for each group.

Vault Workflow

The Vaults ingest content using three different methods:

- FTP pull
- FTP push
- Live capture of MPEG-2 transport streams over UDP

With FTP pull, the original content is kept on an FTP server (catcher), for a period of time and mechanisms are in place to restart ingests until they have successfully completed.

With FTP push, only a window of data is buffered by a device that grooms the live (broadcast) feed and pushes the data to the Vault.

With live capture over UDP, the Vault captures the live multicast feed directly.

nDVR Support for NGOD Deployments

The nDVR feature for the RTSP NGOD deployment provides the following capabilities:

- Streamers can distinguish between requests for VOD content and requests for DVR content
- Streamers route cache-fill requests for VOD content to CDS servers (Vaults, Caching Nodes, and other Streamers)
- Streamers route cache-fill requests for DVR content to third-party sources (nDVR Recorders)
- Streamers generate trick-mode files for DVR content
- Streamers generate GOIDs for DVR content and associated trick-mode and index files
- Streamers support unique copy DVR content

In previous releases, Streamers received cache-fill content from Vaults, Caching Nodes, and other Streamers by way of the Cisco Cache Control Protocol (CCP). For RTSP NGOD deployments, the Streamers received cache-fill content from Vaults, Caching Nodes, and other Streamers by way of the C2 protocol.

Streamers are able to receive cache-fill content from CDS servers (by using CCP or the C2 protocol) and third-party sources. The Streamers can route cache-fill requests to Vaults, Caching Nodes, and other Streamers for VOD content, and to third-party sources for network digital video recorder (nDVR) recordings.

The nDVR feature supports unique copy content distribution from a third-party source (for example, nDVR Recorder) to the Streamer, and from the Streamer to end-user devices, which can be an IP set-top or QAM device.

Asset Metadata

Each content an end-user requests from a device has a unique title ID. For each content, there are different versions based on the encoding that is compatible with the end-user device (for example, high definition [HD] or standard definition [SD] for a STB or mobile device, as well as resolution formats), which are identified by content IDs. When a request for content is sent from the end-user device to the backoffice, it includes the title ID. The backoffice maps the title ID to the content ID for the content that is compatible with the requesting device. The backoffice uses the content ID when communicating with the Streamers on what content object to stream to the device.

For some CDNs, the content ID is a combination of the ADI Product ID and Asset ID (PAID), and it is used to convey both VOD and DVR content. Other CDNs send the content ID from the backoffice to the Streamers in a URI. For the RTSP NGOD deployment, nDVR content is identified with a URI, and VOD content is identified with a PAID.

Each unique content can have several unique data objects required for playback; such as the normal video object for standard forward playback, video objects for trick-mode content, and an index file used to map playback time offsets to corresponding data offsets within the various video files. This information can be referred to as vendor-specific content metadata, or asset metadata.

The Cisco TV CDS software uses a global object ID (GOID) to identify the different video and index data objects for a unique content. The TV CDS software contains an association of the content ID with the various GOIDs used to store the different objects for the content.

To support nDVR, Streamers use the third-party object identifier in cache-fill requests. The Streamer not only stores the content ID to GOID mappings, but also a GOID mapping to an external object identifier which is generated by the third-party vendor. In addition, to support cache-fill from third-party vendors, Streamers generate the GOIDs. The generated asset metadata is revalidated to ensure specifically that the PAID-to-GOID-to-external object mappings are still valid.

Cache-Fill Routing

Streamers are used for streaming out regular VOD content sourced from other CDS servers or DVR content sourced from the Recorders. The Streamers route the cache-fill request to the appropriate source based on the content type, which is derived from the asset name space.

A Streamer must know the origin from which the needed object is sourced when performing cache-fill. Normally, a Streamer is configured with static routes for cache-fill. The Streamer must be configured with different source routes for the different content types.

For VOD content, the content identifier is a PAID. For DVR content, the content identifier is a URI, which contains the hostname or IP address of the third-party source (nDVR Recorder).

nDVR Architecture

The C2 protocol is used for cache-fill of DVR content from the nDVR Recorders.

A request for VOD content is identified by the Provider ID and Asset ID (PAID). A request for DVR content is identified by a URI. The Setup server receives the URI or PAID over the R2 protocol in a NGOD RTSP deployment.

The R2 setup request from the backoffice to the Setup server sends a URI for DVR content to be played. The URI includes the routing and protocol information necessary to cache-fill the DVR content. Content is identified by a URI instead of a PAID.

Cut-Through Support

DVR content can be categorized as unique copy or common copy. Unique copy is a recording of content that belongs to a single subscriber. The Streamers perform cache-fill of unique copy recordings directly from the Recorders. Consideration is made such that any unique video content that is cached is only performed for a transitory period.

Note Only unique copy DVR content is supported

For unique copy recordings, there is no cache gain benefit of a hierarchical caching system, as the recordings cannot be shared across subscribers; therefore, the CDS servers do not cache unique copy DVR content.

Integration with Legacy VBOs

The nDVR feature integrates with legacy video backoffice (VBO) systems; such as Seachange, Axiom, and Ericsson Openstream.

Dynamic Trick-Mode Files

Normally, trick-mode files are generated by the Vaults at the time of ingest. For DVR content, trick-mode files are generated dynamically by the Streamers.

Trick-mode file generation for DVR content is enabled with the **Dynamic Trickmodes** field on the **Configure > System Level > MPEG Tuning** page.

Vault Virtualization

Vault Virtualization provides the following three types of configuration:

- ISA Regionalization
- Shared Content Store
- Virtual Content Store

Note Virtual Content Store provides enhanced features to Shared Content Store.

ISA Regionalization

The ISA Regionalization feature is a combination of the Virtual Video Infrastructure (VVI) and legacy Content Delivery System (VDS). This feature provides the ability to centrally store content on Vaults located in a centralized storage facility and allow remote sites to have a record of inventory of this content and access it by way of the Caching Nodes or directly on the central Vaults. The remote sites still operate as independent entities with their own local Vault Group, local Content Store, and local Streamers; managed by their own CDSMs and possibly accessing their own local BMS and AMS. The Streamers at each remote site can stream both locally stored content and centrally stored content.

The ISA Regionalization feature allows the use of a centralized storage facility containing both Vaults and Caching Nodes in a Virtual Video Infrastructure (VVI), while maintaining a localized or remote VDS at each headend.

For information on configuring ISA Regionalization, see the "ISA Regionalization Configuration Workflow" section on page 3-9.

Centralized Storage

The Virtual Video Infrastructure Manager (VVIM) manages the Vaults and Caching Nodes allocated in the centralized domain. The centralized domain can be distributed across multiple geographic locations; for example, the Vaults could be located in one location and the Caching Nodes could be located in another. The VVIM typically resides in one of these locations.

Each VDS has a virtual view of the VOD content stored on the central Vaults. The centralized content is ingested once, the first time it is requested; any subsequent ingest requests for that same content increments a reference counter.

Remote Site

Each remote VDS has a local Vault Group and communicates with a local BMS and local AMS located at the headend or at another headend nearby. Each remote VDS is able to ingest local content through the local Vaults and is able to access content stored in the central storage facility by way of the Caching Nodes and Vaults in the VVI. The centrally stored content is abstracted from the BMS by means of the local Content Store providing a virtual view of that content to the BMS. Both local and central content are available to fulfill streaming requests received by the Streamers in a remote VDS.

The ISA Regionalization feature uses the existing ISA architecture, but extends the ISA content component to support new behaviors associated with where content is physically located. Each VDS operates with a local ISA Content Store, which is extended to manage both centrally and locally stored content.

Real-time asset (RTA) content is not centralized, and is stored on local Vaults in each headend. The VDS determines if content should be ingested centrally or locally based upon on the type of content (VOD or RTA) that is being requested.

Note If the local Vaults are not available because they are down or have lost connectivity, then the master Streamer in the headend automatically takes over as the Ingest Driver client. If this occurs, when the local Vaults have been recovered and regained connectivity, the Ingest Driver client must be migrated back to the local Vaults before RTA ingests can be restored.

To move the Ingest Driver client from the master Streamer back to the local Vaults, stop and restart the statsd process on the master Streamer by entering the following commands:

```
pkill statsd
/home/stats/statsd -i <server_mgmt_IP_addr> -s <subnet mask> -d eth0
```

Ingest Driver

The ISA Regionalization feature introduces the Ingest Driver, which has a server-side and a client-side. The Ingest Driver server is located at the central location, on the master Vault, and is responsible for managing the content ingestion and deletion requests from the Ingest Driver clients located at the remote sites.

Ingest Driver Server

The centralized Vaults run an internal Naming Service, Notification Service, and Content Store. This Content Store is not associated with a remote BMS, and acts independently of all remote sites. The Ingest Driver gets the Content Store factory from the internal Naming Service, ingests content using the createServant and provision methods, and deletes content using destroy and removeServant.

The Ingest Driver server is started and stopped on the master Vault and is automatically restarted like other ISA processes. When the server is started, it binds to a TCP socket and waits for requests. To handle the requests quickly, there are several threads created to parse the requests and fulfill them. When the server processes the request for each content, only one request is handled; that is, other simultaneous requests for the same content are blocked.

The Ingest Driver server reads the isa.cfg file and incorporates the following Ingest Driver configuration parameters set on the CDSM GUI:

- IngestDriverEnabled=1
- IngestDriverRole=1 (for server)
- IngestDriverHost
- IngestDriverPort
- IngestDriverNoOfThreads

The Ingest Driver server logs events to the IngestDriver.log file located in the /arroyo/log directory.

Ingest Driver Client

The Ingest Driver client is used by the local Content Store to send requests to the Ingest Driver server and receive responses from the server. When a provision call from the local Content Store is received from the backoffice, the Ingest Driver client establishes a TCP connection with the Ingest Driver server, sends the request, and closes the connection once the response is received.

The local Content Store reads the isa.cfg file and incorporates the following Ingest Driver configuration parameters for the Ingest Driver client set on the CDSM GUI:

- IngestDriverEnabled=1

- IngestDriverRole=0 (for client)
- IngestDriverHost
- IngestDriverPOrt
- IngestDriverTimeout
- MarketId

Ingest Driver Content Management

The local Content Stores at the remote sites perform content management of the content at the central facility by interfacing with the Ingest Driver. The Ingest Driver compares the requested content identifier of each content ingestion and deletion request to the VDS repository to determine if the content exists. If the content does not exist, it is ingested using FTP and the FTP URL provided by the remote site. If the content already exists, the repository is updated to maintain the reference between the requesting site and the content. The Ingest Driver returns the VDS internal representation of the content bundle and associated content information, such as file size and bit rate.

When the Ingest Driver receives a deletion request, it determines if the request is for the last reference to the content. If it is the last reference, the Ingest Driver requests that the central Content Store delete the content and associated MPEG files. If it is not the last reference for the content, the Ingest Driver just removes the reference of the requesting site for that content in the repository.

Remote Ingests

At each headend, the external ISA interfaces to the backoffice do not change, and call flows remain the same. The remote CDSM is extended to identify a site as part of a regionalization grouping, and specify the communication information of the Ingest Driver. Internally, the local Content Store application is modified to check for this setting. If regionalization is turned on, the local Content Store application directs VOD (provision) requests to the Ingest Driver and RTA (provisionForPush) requests are directed to the local Vaults.

The local Content Store performs the createServant call locally, thus ensuring that each remote site has its own IOR representing the content object. If the request is distributed, the local repository is updated with the content-specific information, such as the content bundle, file size, and bit rate returned by the Ingest Driver. This allows the remote site to have local representation of centrally stored content.

For RTA content, the process is same as it has always been for the VDS. The local Content Store processes the provision call (provisionForPush for RTA) and directs the local Vaults to perform the ingest of the content.

Remote Streaming

Local streaming is accomplished by way of the Cache Control Protocol (CCP) locate capability. Each remote site is configured to communicate to a specific set of storage devices which could include local Vaults, central Vaults, and Caching Nodes. The locate feature broadcasts a request for a specific content, and the system performs a cost analysis to determine which storage device can best provide service. For VOD content, if the content is not already cached on the local Streamers, it is acquired from either the central Vaults or Caching Nodes. For RTA content, if the content is not cached on the Streamers, it is acquired from the local Vaults. However, knowledge of the content type is not required as the locate capability is able to determine its location.

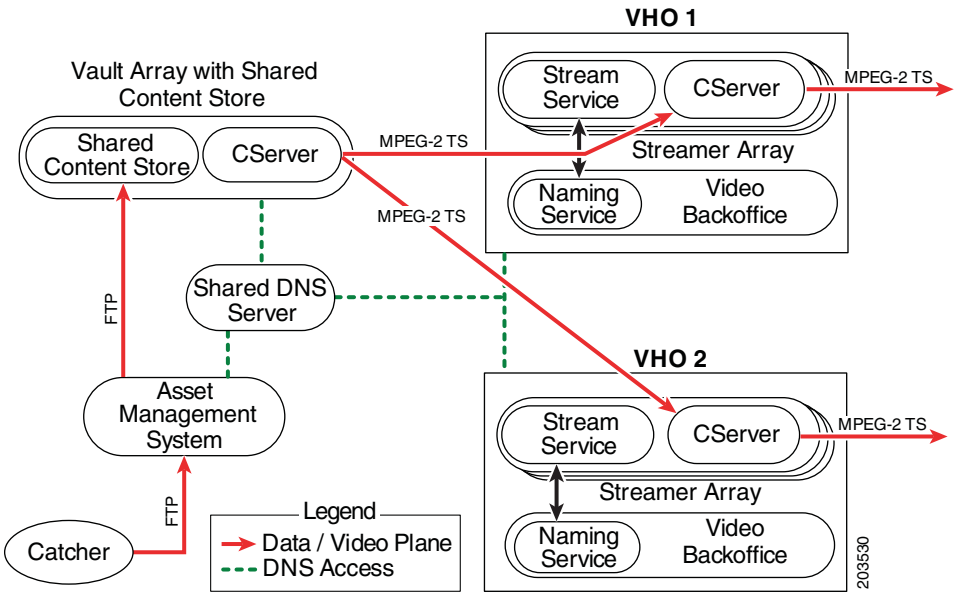
Shared Content Store

Shared Content Storage, also known as Shared Content Store (SCS), works with a single, centralized AMS and catcher, through which all initiation for content ingest and content deletion is sent. The SCS handles ingest and deletion requests from multiple backoffices by way of the central AMS. The scenario of backoffices independently ingesting and deleting content through their local AMS is not supported.

Note The Content Storage feature requires the Virtual Video Infrastructure feature with Caching Nodes.

Figure 2-8 shows a high-level view of the SCS and a single, centralized AMS for multiple video hub offices (VHOs). A VHO is a local deployment that includes the video backoffice, Streamers, application servers, QAM devices, and other headend equipment.

Figure 2-8 Shared Content Store For Multiple Video Headends



The DNS server typically runs on the BMS server. The Naming Service is part of the video backoffice (VBO). All CORBA components, including the AMS, Stream Service, and Content Store, need to register with the Naming Service. The catcher receives or "catches" new content assets from an external communication device such as a satellite transmission or FTP server. After the package is received completely by the catcher, it sends the package by way of FTP to the AMS. The package consists of video and image content assets, as well as ADI metadata.

Following are the requirements for the SCS feature:

- Single, shared DNS server with all devices registering their hostnames to it. A central, shared DNS is required to resolve multiple Naming Services belonging to the different VHOs.
- Hostnames must be unique for all devices. This is required for the Naming Service discovery.
- Each VHO has its own Naming Service to which the ISA components of the VHO register.
- AMS controls the ingest and deletion of content.
- The Vault array has one SCS.
- SCS registers with each Naming Service.

A VVI with SCS must be initialized in the following order:

1. The shared DNS server must be up and running before starting up the shared AMS, SCS, and VHO devices.
2. SCS successfully registers with the Naming Service for each VBO.
3. Each VHO Stream Service registers with its respective Naming Service.

Ingesting Content with the Shared Content Store

Upon receiving the content package, the AMS schedules it for ingest by informing the Package Factory in each participating VBO of the content package, and passing the pertinent information (the ADI metadata, the URL where the content package can be accessed in the AMS, and the verb *ingest*).

The SCS creates one interoperable object reference (IOR) for each content package. The IOR is returned to all VBO Package Factories that request it, including any that requested it at the time the IOR was being created.

Deleting Content with the Shared Content Store

To delete content that was ingested for more than one VBO, the AMS is used to send the *export package delete* request to each VBO. The content is deleted from the Vault array only when all VBOs have requested the deletion. If one or more VBOs have not requested that the content be deleted, the content remains in the Vault array.

Virtual Content Store

The Virtual Content Store feature in an ISA environment replaces the Shared Content Store feature. The Shared Content Store (SCS) feature is the ability of several local sites (video hub offices [VHOs]) to ingest content at a central location and share that content with the other VHOs. The SCS feature eliminated ingesting multiple copies of the same content.

Vault Virtualization replaces the SCS with the Virtual Content Store (VCS). No content is ingested at the local VHO. All ingests and deletions of content occur at the central location, and both ingests and deletions are initiated by the local BMS at each local VHO, just as they were in the SCS. However, the VHOs do not need to communicate with the super headend (SHE) as they did with the SCS feature. With VCS, communication of ingestions and deletions is handled by the Ingest Driver client residing on a Streamer in each VHO and the Ingest Driver server residing on the master Vault in the SHE. Vault Virtualization requires that Vault Groups be disabled.

The Virtual Content Store (VCS) component runs on a Streamer in the Stream Group, and if a failover occurs, the VCS fails over to another Streamer in the Stream Group.

Only one copy of the centrally located asset is ingested and shared by the system, and the asset is only deleted when all VHOs have requested the deletion.

For information on configuring Virtual Content Store, see the "Virtual Content Store Configuration Workflow" section on page 3-9.

BMS Considerations for ISA Environments

The TV VDS integrates with Interactive Services Architecture (ISA) used in business management systems (BMSs) such as the Tandberg OpenStream and the RTSP used in BMSs such as ARRIS nABLE, as well as in environments that are a combination of both ISA and RTSP. The BMS determines the roles and responsibilities of the TV VDS.

OpenStream ISA Integration

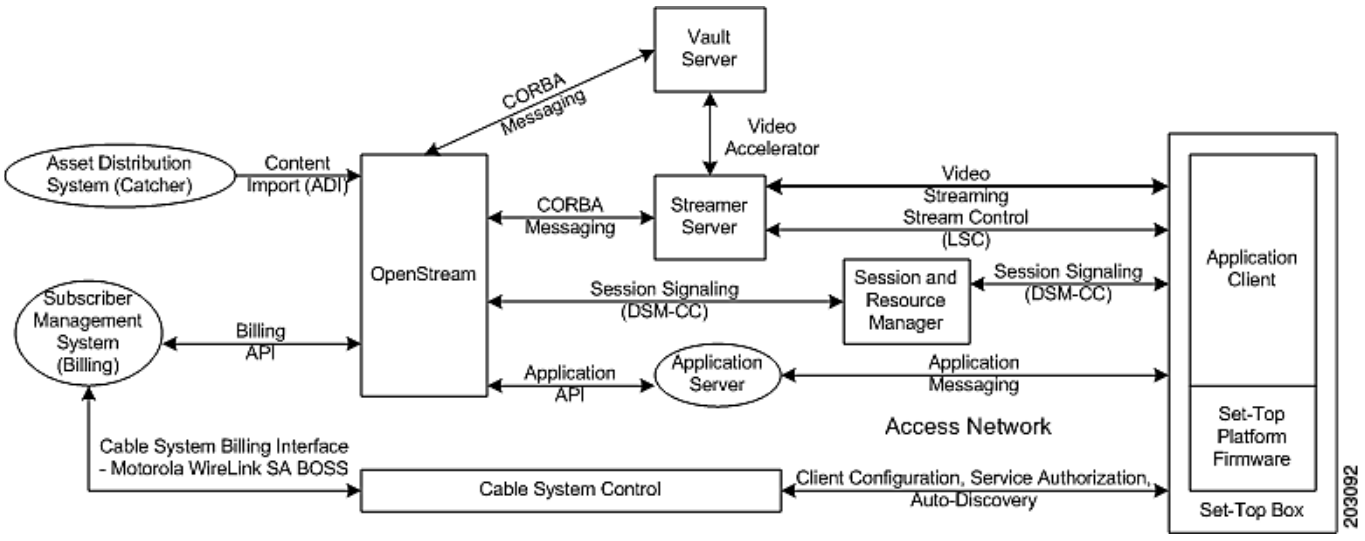
The OpenStream BMS is built on Common Object Request Broker Architecture (CORBA) and provides naming and notification services. The Naming Service allows the TV VDS to locate objects in the system such as content, equipment, assets, and so on. The Notification Service allows the TV VDS to listen for important events in the system as well as to send events to the OpenStream BMS and other components in the system.

Note Dual conditional access systems (CAS) for ISA environments, Cisco/Scientific Atlanta Power Key Encryption System (PKES) and the Motorola OffLine Encryption Station (OLES), is supported. A field on the Monitor Completed Ingests page indicates whether the ingested content is encrypted or not. Both clear and encrypted content can be ingested.

For more information on the configuration parameters required to facilitate communication between the OpenStream BMS and the TV VDS, see Appendix C, "BMS Communication."

Figure 2-9 illustrates how the TV VDS integrates with the OpenStream BMS.

Figure 2-9 TV VDS Integration into the OpenStream BMS



Streaming Mode

OpenStream uses a session-based approach to handle resource requirements and allocation. In the course of setting up a session, a QAM device is specified that has available capacity and connectivity to the Streamer and the STB requesting the service. Typically, the Session and Resource Manager (SRM) is responsible for the allocation of network resources. OpenStream uses the Digital Storage Media-Command and Control (DSM-CC) session management protocol to request resources from the SRM.

When using gigabit Ethernet for streaming, OpenStream communicates with the SRM to negotiate network resources and allocation for sessions.

When using Asynchronous Serial Interface (ASI) for streaming, the Streamer performs the role of the SRM by managing and allocating the access network resources and providing this information to the OpenStream BMS.

Steering Ingests

The Ingest Steering feature offers the ability to have one BMS send ingest information to the master Vault, and depending on the product ID in the content name, the content is either ingested by one of the Vaults in the national Vault Groups, or it is ingested by a specific local Vault Group.

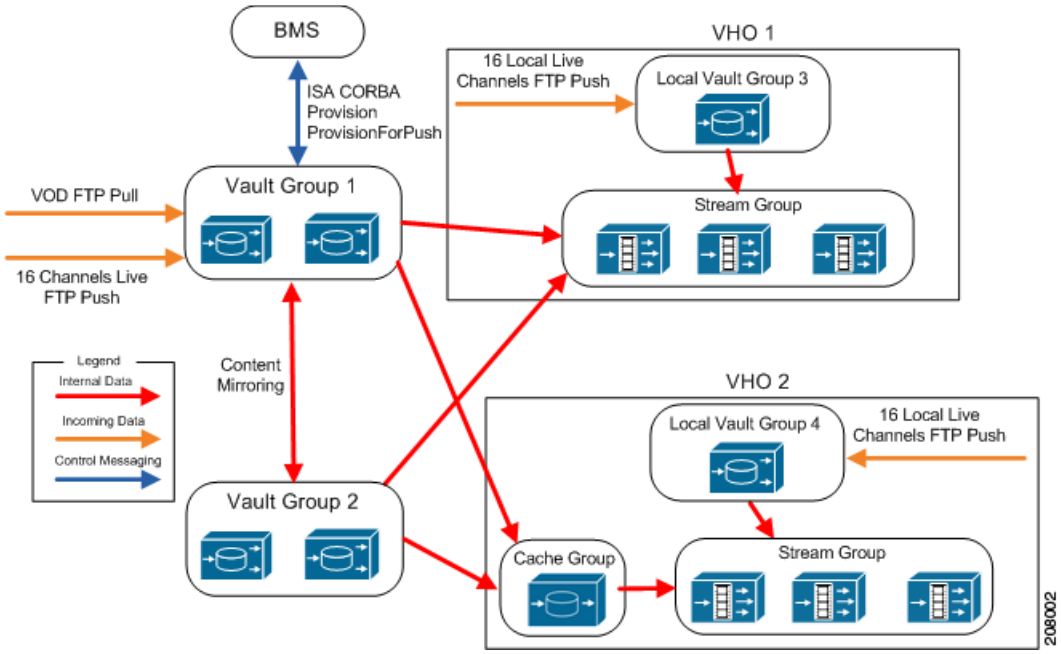
Note For the Ingest Steering to function correctly, the content name must be in the following format: ProviderId::AssetId::contentName.

The Ingest Steering feature requires that VVI with central management and Vault Groups be enabled.

Figure 2-10 shows a high-level view of Ingest Steering for a single, centralized BMS and multiple VHOs. Each VHO has a local Vault Group through which all local live content is ingested. Each Stream Group streams local live content as well as national live and VOD content. The BMS sends messages to the master Vault Group (Vault Group 1), and depending on the product ID and the ingest steering configured, the content is ingested by either the local Vault Group or the national Vault Group.

Content objects on the national Vault Groups are mirrored among each other, while the content on the local Vault Groups are copied to separate hard drives on each Vault.

Figure 2-10 Ingest Steering



Network Connections

The network connections for a TV VDS with Vaults and Streamers, a TV VDS with ISVs, and a TV VVI with Caching Nodes all have different network connections. Table 2-2 lists the different required interfaces for each VDS server. The interfaces are described in the following sections. Figure 2-11 illustrates a TV VDS with Vaults and Streamers. Figure 2-12 illustrates a TV VDS with ISVs. Figure 2-13 illustrates a TV VVI with Caching Nodes.

Table 2-2 VDS Required Interfaces

Interface	Vault	Streamer	ISV	Caching Node
Management	1	1	1	1
Ingest	1	—	1	—
Cache	1 to 8	1 to 13	1 to 4[1]	1 to 12
Stream	—	1 to 13	1 to 4	—

1. The cache interfaces on an ISV are used for content mirroring among ISVs.

Note Table 2-2 lists the mandatory interfaces for each VDS server. If HTTP Streamers are used in a VVI, each Caching Node must have one interface designated as the Locate interface. Stream Control is an optional interface function. For more information, see the "Configuring the Interfaces" section on page 4-63.

Figure 2-11 shows the different logical networks of a VDS consisting of Vaults and Streamers. The ingest network receives content from the content source by way of an FTP staging server or FTP catcher and the content is ingested by the Vaults. The management network consists of communication between the CDSM and the BMS, as well as communication with the Vaults, Streamers QAM devices, and STBs. The cache network consists of Vaults and Streamers.

Figure 2-11 Vault and Streamer Network Connections

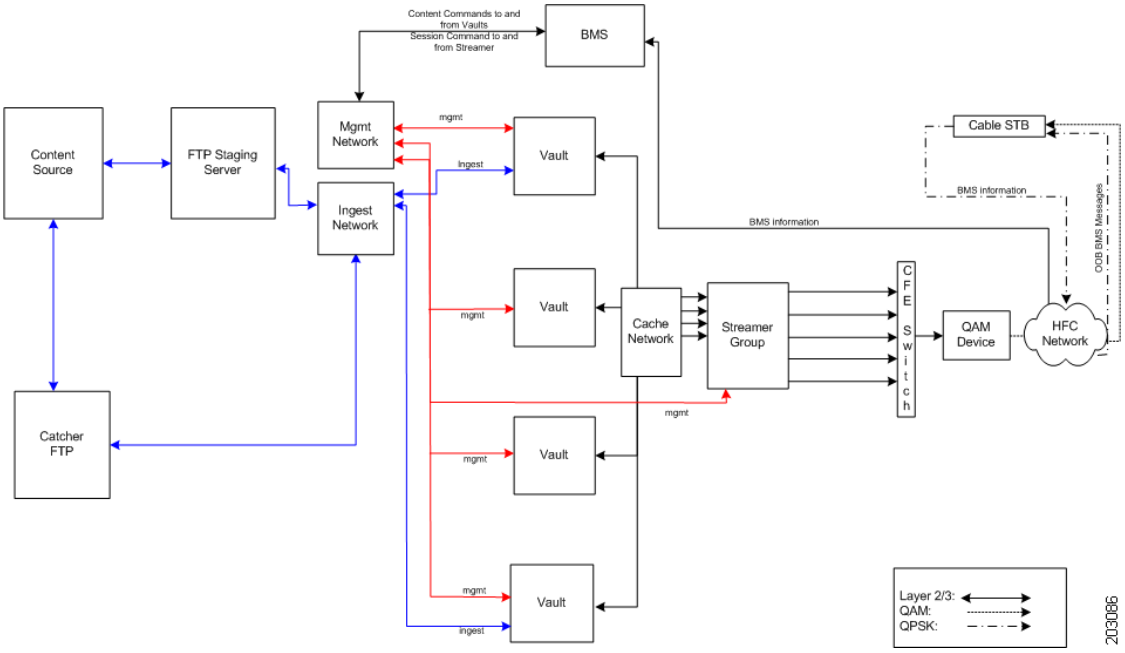


Figure 2-12 shows the different logical networks of a VDS consisting of ISVs. The ingest network receives content from the content source by way of an FTP staging server or FTP catcher and the content is ingested by the ISVs. The management network consists of communication between the CDSM and BMS, as well as communication with the ISVs, QAM devices, and STBs.

Figure 2-12 ISV Network Connections

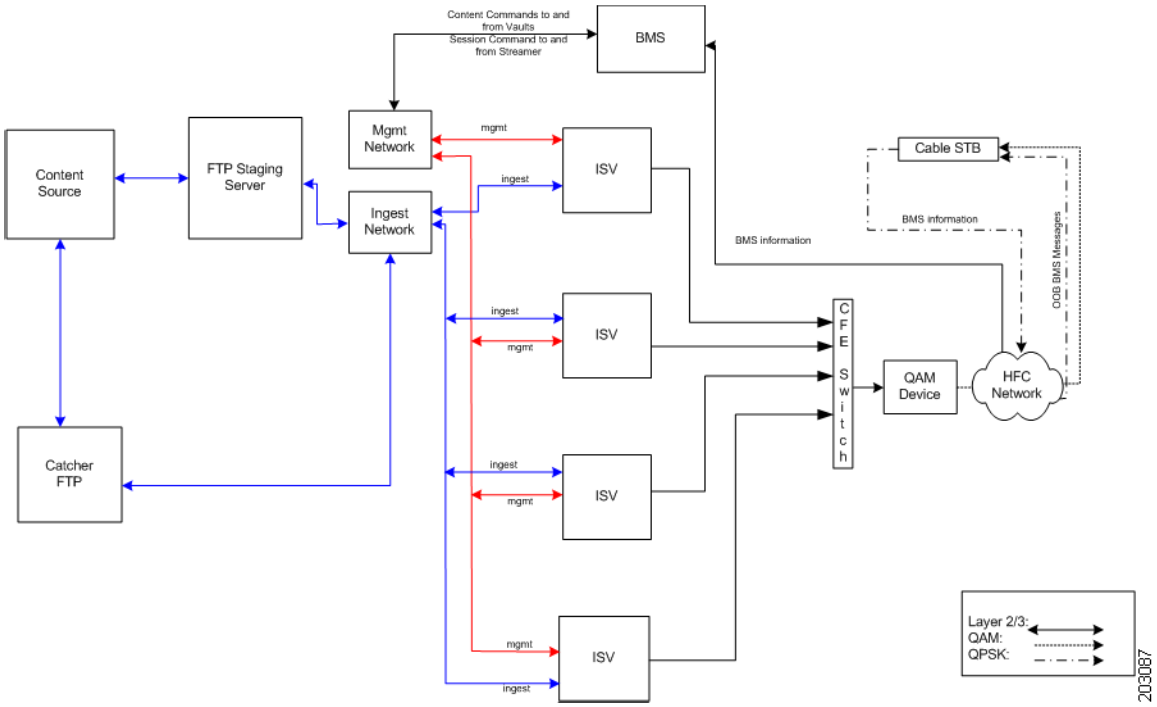
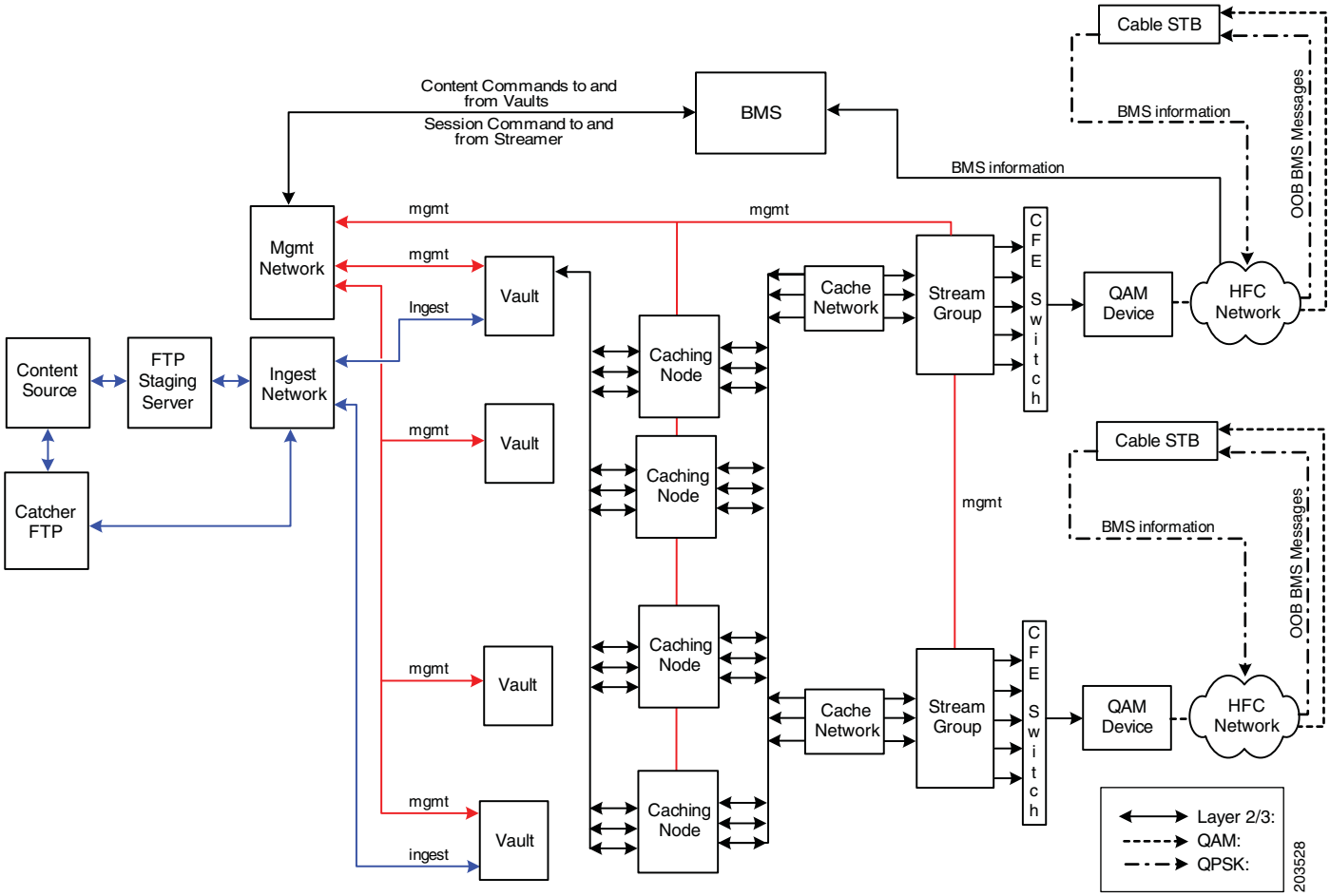


Figure 2-13 shows the different logical networks of a VVI. The ingest network receives content from the content source by way of an FTP staging server or FTP catcher where it is ingested by the Vaults. The management network consists of communication between the CDSM and BMS, as well as communication with the Vaults, Streamers, Caching Nodes, QAM devices, and STBs.

Figure 2-13 VVI Network Connections



Ingest Interface

The ingest interface takes in FTP traffic from the content provider at a maximum rate of one gigabit per second. After the Vault server receives URL information about the content from the BMS by using the management interface, the ingest interface either (1) receives FTP traffic by acting as an FTP client, or (2) receives live data upon receiving a request to act as the FTP server.

When using Layer 2 packet forwarding, to segregate all ingest traffic through the switching fabric, we recommend the use of a port-based VLAN.

Management Interface

The management interface communicates with the network management system (NMS) by way of SNMP, the BMS by way of ISA commands and also RTSP, and with all Vaults, Caching Nodes, and Streamers in the same array. Information shared among servers in the same array includes the following:

- Host service information
- Domain Name System (DNS) service information
- QAM gateway information
- All ISA information

Management traffic is low volume; however, when using Layer 2 packet forwarding, we recommend using a port-based VLAN to ensure delivery of critical management communications.

Cache Interfaces

The CCP uses the cache interfaces on the Vaults, Caching Nodes, and Streamers to send the following data to the servers in the same array:

- Content sent to the Streamers
- Content mirrored among the Vaults
- Messages containing information used for performance optimization exchanged among all the VDS servers

Note All Cisco VDS servers are connected through a switch fabric. Because all Vaults, Caching Nodes, and Streamers in the same array exchange heartbeat messages through the cache interfaces, it is important to ensure there is enough bandwidth among switches involved in delivering cache traffic and to support the same aggregated amount of traffic on all cache interfaces.

When using Layer 2 packet forwarding for cache traffic, we recommend the use of a port-based VLAN.

Cache/Stream Interfaces

The cache/stream interfaces on the Streamer server can be used for both cache and streaming traffic. The number of interfaces designated for each traffic type is configurable. If an interface is configured for both cache and streaming traffic, priority is given to the higher-bandwidth stream traffic, provided cache traffic is able to transmit on other interfaces.

When using Layer 2 packet forwarding for cache and stream traffic, we recommend the use of a port-based VLAN.

Streaming Interface

The streaming interface delivers streaming traffic consisting of MPEG-2 transport streams to STBs by way of QAM devices.

If an interface is configured for both stream and cache traffic, and the jumbo frames feature is not enabled for stream traffic while jumbo frames is enabled for cache traffic, stream traffic uses 1500-byte packets while cache traffic uses jumbo frames.