CONTROLLING AN INFINITE NUMBER OF CHANNELS

Doug Makofka
Motorola, Inc. Broadband Communications Sector

Abstract

A consumer sizes a typical cable system based upon the number of channels that can be accessed. Cable systems that are capable of delivering everything 'on demand' must be sized differently. The size of an on-demand system is related to the amount of media (movies, switched broadcast, live broadcast, etc.) in the on-line library. Systems capable of delivering any content 'on demand' present the experience of having an infinite number of channels.

Channel-oriented delivery systems and media-oriented delivery systems have significant differences. Media-oriented delivery systems need to offer generic, high-level delivery and transport functionality to the service and application control subsystems that manage the media. This is in contrast to channel-based delivery systems that only need to assign a service to a channel.

This paper presents an architectural and functional introduction to media-oriented delivery systems, including the ramifications to access control, bandwidth management, network management, and media transformation subsystems.

ON-DEMAND SYSTEMS STRUCTURE

Systems that are capable of supporting a large number of on-demand services are structured differently then systems built primarily to deliver broadcast services. On-demand services are transactional by nature. Some group of equipment in the network must actively process the request for content from the customer. This is in contrast to the broadcast service, which in most cases can be represented by a fixed channel map, and does not require any processing from the network in order for the tune operation to occur. A simple 'channel up' button push suffices. OnDemand systems require content caches of some type. This is in contrast to broadcast-oriented systems which simply act as a conduit between a service originator and the client devices.

On-demand services and the content/media associated with on-demand services can be managed separately from the delivery of the service/content. For this reason, OnDemand system management cleaves nicely into two pieces: Service-Application-Content support, and Media Delivery support. **Figure 1** shows the high-level structure of the OnDemand System, which reflects this division. This paper will give an overview of each subsystem, then focus on the Delivery Network, and how it provides resources to deliver 'infinite channels'.

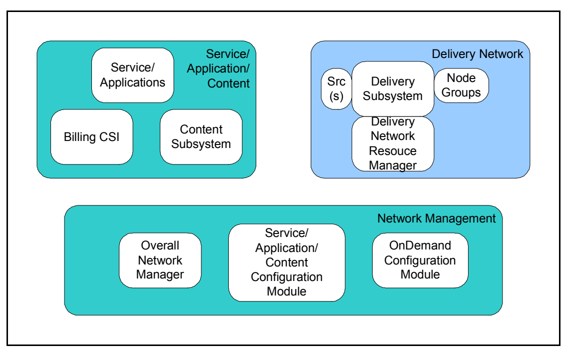

Figure 1 High-Level OnDemand System

Service/Application/Content (SAC) Subsystem

The Service/Application/Content (SAC) Subsystem has the following primary functions:

1. Determining how content is presented to the customer – host the service, access control;
2. Manage the content – including 'pitch-catch functions, and content distribution;
3. Primary billing interface;
4. Supports the automated provisioning and management of OnDemand services.

The SAC subsystem is where the majority of the added value of the current VOD systems resides. The Pegasus Interactive Services Architecture (ISA) standard is one definition of a SAC subsystem.

Access control is completely in the context of the SAC subsystem. This subsystem decides whether or not content/media will be sent to a client. The Delivery Network assumes that access control has already been applied. If it receives a request to deliver content it will. Note that encryption in the OnDemand system is client-based, not content-based. This is in contrast to broadcast systems where encryption is applied to content, and access rights are given to clients.

Delivery Network (DN) Subsystem

The Delivery Network is responsible for the delivery of content to the consumer/client device as directed by the SAC subsystem. The primary functions of the DN subsystem are:

1. Effectively manage DN resources (bandwidth, encryption, transcoding, insertion, QoS, etc);
2. Provide an open, high-level interface for requesting DN resources;
3. Manage and control devices from different vendors so that they cooperate seamlessly in the delivery of on-demand content;
4. Supports the automation of provisioning, configuration, and management of the DN.

In current VOD environments, the VOD systems are either the actual or *de facto* managers of the DN subsystem. This as been driven by the non-routable networking (ASI) between the VOD servers and the DN equipment (modulators, upconverters). With the advent of GIGE transport of content/media, the delivery of content is not necessarily determined by the output of the server. This is a strong motivation for removing the DN network management from the content environment.

Nework Management (NM) Subsystem

The Network Management (NM) subsystem, as it relates to On-Demand subsystems has the following functions:

1. Maintain a consistent view of system topology;
2. Maintain IP-level device configuration (DCHP device records, for example);
3. Maintain higher-level functional configuration for the Delivery Network;
4. Maintain and distribute higher-level resource to content source mappings between the Service-Application-Content subsystem and the Delivery Network subsystem.

The Network Management subsystem can 'see' across the two functional subsystems. It coordinates system provisioning and management activities. Even though the NM subsystem does not play an active role in the delivery of OnDemand services, it is needed to make large-scale OnDemand systems practical. Systems capable of delivering 'everything on-demand' will be quite large, and require automatic provisioning and management – to the extent that equipment can be placed in a rack, automatically determine its configuration and functional responsibilities when it powers up, without direct operator intervention. This will be discussed further in the *Automatic Configuration and Provisioning* section.

DELIVERY NETWORK (DN) STRUCTURE

The OnDemand system structure is based around the User-to-Network concept used in the DSM-CC User-to-Network Session protocol. It is natural to use this protocol as the basis of requesting and delivering resources from the DN; however, other protocols can also be accommodated – such as RTSP.

The Delivery Network (DN) offers 'resources' to the Service-Application-Content (SAC) subsystem. The SAC subsystem requests these resources from the DN as part of a session setup operation. The session is used to carry content/media associated with a service to a specific customer's client device. Examples of DN resources include bandwidth, multiplexing, encryption, transcoding, and rate shaping. A major goal of the DN subsystem is to provide these resources is a well-defined manner so that the SAC subsystem does not need to understand how these resources are mapped to specific devices in the network, or how to control those devices. This provides a basic decoupling of the DN and SAC subsystems' architecture.

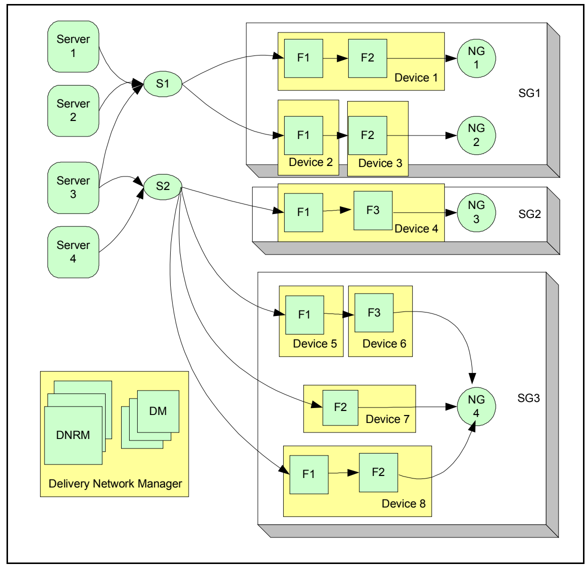

Figure 2 Delivery Network Structure

Figure 2 gives a high-level look a the structure of the DN. The arrows in the diagram show media paths. Control paths are omitted. The definitions of the elements in **Figure 2** are as follows:

Server1…n These are content/media servers. They are outside the DN, but are shown here as the connection points to the DN. The connections are likely GIGE, but ASI and other connections are also accommodated;

S1,S2 These are *Sources*, or SRC, which are the actual connection points into the DN. In the case of GIGE media transports, a Source is equivalent to an IP subnet. In the case of a ubiquitously switched system, there is only one Source;

Device 1...n These are the actual devices or products that are going to perform functions in the DN;

F1…n These are the *Functions* performed by the Devices (mux, modulate, encrypt, etc.). Each Function is described by a *Function Block*, or *F-Block* for short. F-Blocks map directly to the resources offered by the DN to the SAC subsystem;

NG1…n These are *Node Groups* – defined by the structure of the combining networks on the output of the Inband RF modulators. A Node Group represents the set of ‘outputs’ that can be seen by a particular client device. It also is a primary topological grouping of client devices;

SG1…n These are Service Groups. A Service Group is a collection of Node Groups. The motivating idea behind Service Groups is that they collect Node Groups that have the same Resources (F-Blocks). This helps to decouple NG topology from the SAC subsystem;

F-Block Chain Each path from a SRC to a NG is called an F-Block Chain. For example, there is an F-Block chain from SRC S1 through Device1, to Node Group NG1. Resources F1 and F2 are available on this chain;

DNRM The Delivery Network Resource Manager (DNRM) is the part of the Delivery Network Manager that handles the initial requests for resources from the Service\Application\Content (SAC) subsystem, allocates the resources to specific F-Blocks, then commands the Device Manager (DM) associated with the F-Blocks to do work;

DM The Device Manager (DM) translates standard F-Block behavior requests into device-specific commands.

Every device or product used to process media in the DN must be described as one or more F-Blocks. For example, referring to **Figure 2**, if F1 is a modulator, and F2 is an upconverter, then Device1 is a product that takes its input, modulates it, upconverts it, then outputs the processed RF signal. From the standpoint of the Domain Network Resource Manager (DNRM), it doesn’t matter what Device1 is. Either the device supports standard F-Block commands directly, or there is a Device Manager (DM) that translates the standard modulator, and upconverter F-block commands into device-specific commands. The following is a partial list of potential F-blocks:

1. RF Modulator (QAM64, QAM256, etc.);
2. Upconverter;
3. Encryptor;
4. Inserter;
5. Multiplexer;
6. JitterBuffer/QoS shaper;
7. Transcoder;

Delivery Network (DN) Transactions

The DN subsystem provides resources used by the SAC subsystem to deliver services. There are many potential on-demand service types. Video On Demand (VOD) and its variants, Network Personal Video Recording (NPVR), Switched Broadcast, and others. The DN is unaware of these service types; however, all of these service types are accommodated by the DN using the same set of resources.

A User-Network model is used to structure the DN transactions. The SAC subsystem, and the customer’s client devices are each Users. The DN is the Network. The primary transaction structure is one of the Users requesting sessions for media to be delivered by the DN. These sessions have resources associated with them. Bandwidth is the fundamental resource, but there are others – those defined by the F-Blocks supported by devices in the DN. The typical session setup trans action is shown in Figure 3.

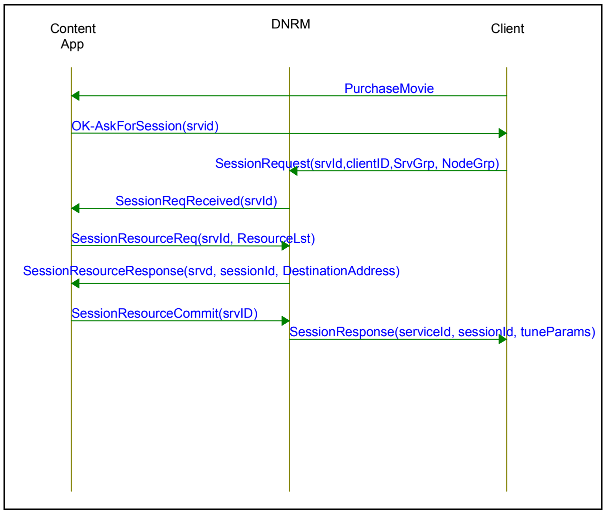

Figure 3 Session Setup Transaction

Notice that the session setup transaction is compatible with the DSM-CC U-N Session Setup Protocol. RTSP could also be accommodated by adding a few new functions. Another protocol variation is having the Content/App subsystem issue the SessionRequest command, or by just issuing a SessionResourceRequest indicating a new session is to be established. The DNRM could handle both scenarios from either protocol without any problems.

Managing the Delivery Network

Figure 4 shows the standard control flows in support of the DN. Notice that there are resource/session flows, and network management flows. Each are needed to support the OnDemand system. There are two levels of management in the DN:

1. F-Block or resource management
2. Device management.

Requests for resources are always at the F-block level. DN resources are allocated on the basis of F-Block allocation. Each F-Block type as an associated allocation scheme that allows the Delivery Network Resource Manager (DNRM) to know where unallocated resources reside in the DN. The DNRM handles the resource allocation based on the F-Block allocation scheme. The resource request is then translated to F-Block parameters which are passed to the Device Manager (DM) associated with the device that will provide the requested F-Block function.

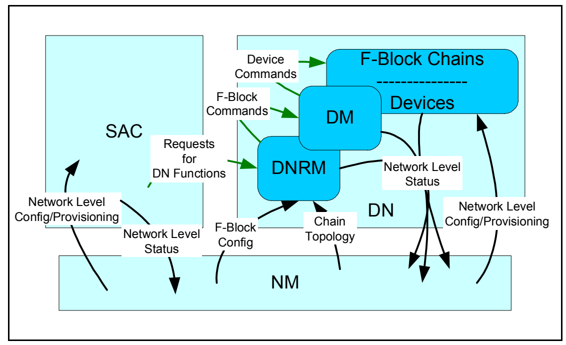

Figure 4 Management Control Flows

The Device Manager (DM) accepts the F-Block function request and issues the proper commands to the device. The DM may also request that the DNRM adjust the 'available resource pool' for the F-Block in questions. This allows device resource allocation to deviate from the standard F-Block allocation scheme.

Automatic Configuration and Provisioning

Delivering on-demand services to large subscriber populations requires networks that must support a large number of simultaneous 'channels'. These systems can be orders of magnitude bigger than broadcast-oriented system. In addition, there is are on-going control transactions between subsystems. The control transactions demand that each subsystem is 'in sync' with at least some portion of the overall topology and state of the OnDemand system. DN topology representation is an important part of automatic OnDemand system configuration.

OnDemand System Topology

The basis of DN topology is Network Topology. An example of Network Topology is given in Figure 5.

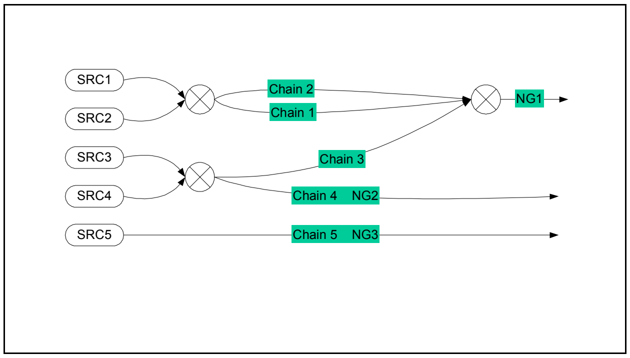

Figure 5 Network Topology

Network Topology is the framework networking that maps the DN sources into Chains, that reach Node Groups. In a GIGE-based system, Network Topology is a subset of the GIGE IP network configuration. This shows the close connection between the DN configuration and provisioning and the underlying network configuration – in this case IP.

Once Network Topology is defined, devices must be placed in the Chains between the Sources and Node Groups. This representation of topology is called Chain Topology. An example of Chain Topology is given in Figure 6. Chain Topology is the configuration information used by the Delivery Network Resource Manager (DNRM) to turn requests for resources into F-Block allocations.

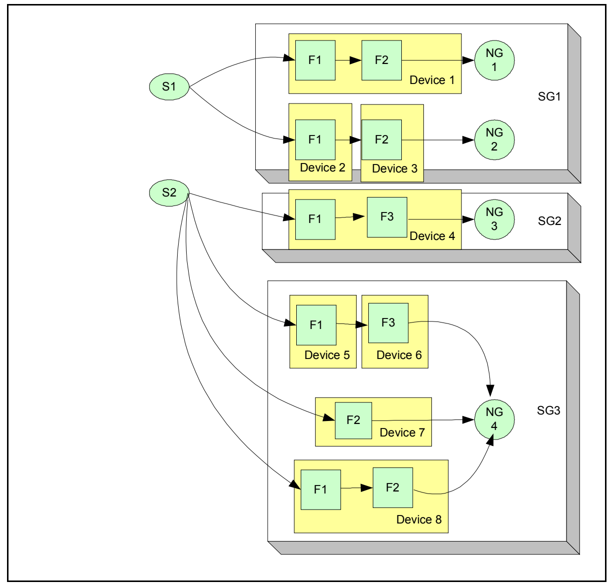

Figure 6 Chain Topology

The Service/Application/Content (SAC) subsystem may need some DN configuration information in order to efficiently distribute content to servers. Source Topology meets this need. Figure 7 shows the picture of the DN conveyed by Source Topology.

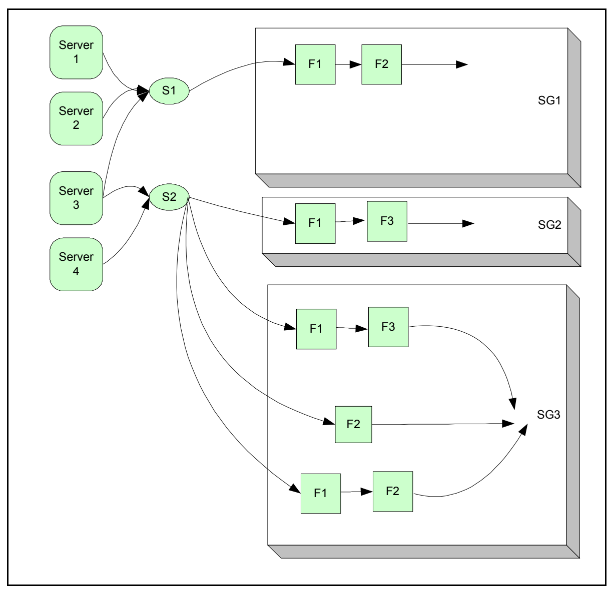

Figure 7 Source Topology

The SAC subsystem can use the DN without Source Topology. The content/media used by the services supported by the SAC will require specific resources of the DN. The

SAC could just request those resources without any notion of Source Topology; however, some optimizations in the SAC environment are possible with a knowledge of Source Topology. One example is that content/media requiring a resource not available in a specific Service Group could be hidden from customers in that service group. This would prevent requests for content it is impossible for the DN to deliver.

<u>Delivery Network Configuration Module</u>

The Delivery Network relies on topology, F-Block, and Device information. This information must also be used to drive or modify IP-level configuration information, such as DHCP records. This information cannot be hand-crafted. It must be formed from higher-level configuration operations. The Delivery Network Configuration Module (DNCM) automatically generates the topological and configuration information. Graphic interfaces are used to 'stage' the OnDemand system. The graphic interfaces allow manipulation of the topological diagrams shown in this paper, as well as detailed F-Block and device information.

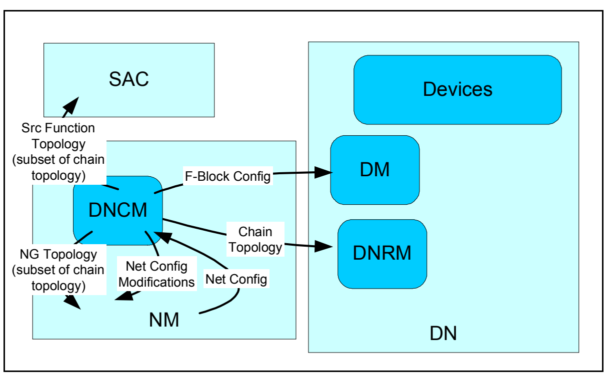

Figure 8 Delivery Network Configuration Module (DNCM) Environment

Figure 8 shows the environment in which the DNCM functions. It also shows the major information flows from between the DNCM and the other OnDemand System entities.

Among the DNCM's major operations are the following:

1. Load/Modify/Delete Network Layer Topology;
2. Load/Modify/Delete F-Block Defs;
3. Load/Modify/Delete Device Defs (each device participating in the DN will supply a device definition package);
4. Load/Modify/Delete NG Defs;
5. Configure Chain Topology;
6. Add /Remove Device to/from DN;
7. Enable/Disable Device in DN;
8. Associate Network Layer With Device Layer (effect any coordination between the subsystem managing DHCP);
9. Build Configuration Script;
10. Execute Configuration Script;
11. Configure DNRM and DMs (these managers need their own configuration based on system size, dedunancy strategy, etc.).

Ideally, an MSO can have individual systems configured off line by a knowledgeable systems engineer. These configurations can be sent to the locations where the DN exists. Technicians at the DN sites can then 'rack and stack' the devices needed in the DN. Using the DNCM, the technicians at the DN site can apply the configuration to the newly-installed devices without any additional configuration operations. Ongoing configuration and device changes can be handled by the local technicians at a high level. The DNCM will be able to d sanity checking on these changes. The DNCM can then make sure the changes are applied in a controlled, consistent fashion across the entire OnDemand System.

CONCLUSION

OnDemand systems are different the broadcast-oriented systems, hence, require a different structure and management strategy. Yet, it is possible to manage and control large OnDemand systems. Managing Delivery Networks is more than just assigning bandwidth. All Delivery Network functions must be available to the users of the Delivery Network via high-level functions that are well-defined, open, and allow competition among device vendors supplying products that provide Delivery Network Functions. Creating a Delivery Network management subsystem, that operates independently from the Service/Application/Content subsystems, will make delivering everything on-demand a technical and practical reality.