COMPLEXITY CONSIDERATIONS FOR CENTRALIZED PACKAGING VS. REMOTE PACKAGING

Brian Tarbox
Motorola Mobility

Robert Mack
Motorola Mobility

Abstract

Adaptive streaming protocols will be a critical component for operators offering IP video services. One of the key functions in Adaptive streaming is a "packaging" function that creates playlists/manifests, segments the video into chunks, and "wraps" the chunks to make them suitable for one of several protocols. There is an ongoing debate as to the merits of where to perform adaptive stream packaging within a service provider's content delivery network (CDN). Various analyses have considered centralized, distributed, and edge packaging architectures. These analyses primarily considered the CDN bandwidth and storage savings that could be attributed to distributed/edge packaging architectures versus the operational complexity that would likely result. In addition, these evaluations focused more on video on demand (VOD) rather than linear content distribution.

For many Service Providers the ability to centralize all transcoding and packaging operations is appealing, particularly if they own the CDN and are therefore less concerned with the per-bit content distribution costs. For other Service Providers, particularly those that want to augment their existing service with streaming capabilities, but are sensitive to these costs and the costs associated with standing up large centralized video processing centers, the ability to customize content at the edge may make more sense. So, for example, a Tier 2 or Tier 3 operator may want to augment his offerings out of an existing Regional Headend.

In addition, edge packaging may offer options that can reduce the complexity associated with providing certain desired system functions. For example, considering regional ad insertion and blackout, edge packagers can incorporate simple functions that emulate similar legacy system capabilities which minimize the impact to a service provider's network and operations.

Finally, this paper will also explore some of the unique functional capabilities that packagers can offer in support of centralized or regionalized architectures, including intelligent access network capacity management, playlist obfuscation for ad insertion, regionalized blackout, and support for both legacy and advanced advertising in adaptive environments. This will enable operators to fully understand the trade-offs of implementing various packaging architectures and make the right choices when rolling out IP video services.

INTRODUCTION

Previous papers examining where to perform packaging focused on network bandwidth and capital expenditures as the variables to measure. This paper examines other factors that must be explored in order to enable sound decision making in systems architecture design. These factors focus on the processing complexity required to output segments and manifests under various expected conditions. These complexity considerations may make the support for certain desired capabilities problematic. In particular, this paper examines the ability of the various packager configurations to support regionalized and targeted ad-insertion as well as to support blackout processing.

BACKGROUND

Packager Locations Within the Distribution Network

Until recently most discussions about packager locations listed center and edge as the options. In 2012, however, the concept of a hybrid center/edge packager has gained traction. Each will be described briefly.

In the center packager, the packaging function configuration is embedded within the transcoder, or the packager is connected to the output of the transcoder which, in turn, is connected to the origin server. All video stream processing, including fragmentation, ad-insertion, manifest creation, etc. is performed prior to any client request for the content. All fragments and manifests that might be required or requested are deposited onto the origin server.

In the edge packager configuration, the transcoder outputs its fragments (e.g., MP4 or FMP4) and optionally a mezzanine manifest file onto the origin server. One or more packagers are located "south" of the CDN. The client connects directly to the packager or to an optional edge server which redirects to the packager. The packager then requests fragments and the optional mezzanine manifest file from the origin server, transwrapping on-the-fly into the appropriate format for the requesting clients (e.g., HLS, HDS, HSS).

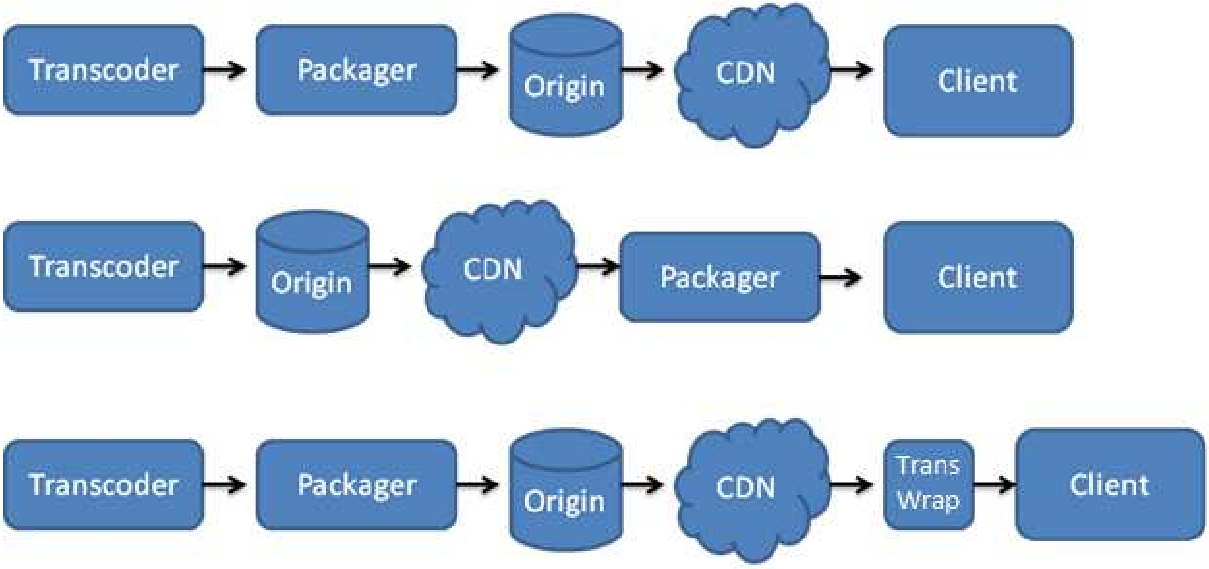

Figure 1: Three Styles of Packaging

The hybrid solution is similar to the center configuration except that it adds a transwrapper component located between the client and the CDN. This is a relatively new style of configuration and so many options are possible. The “north side” packager might transwrap to one of the four standard formats (HLS, HDS, IIS, DASH), leaving the transwrapper to re-wrap only if a given request was for a different format. The transwrapper component might also perform session or region specific operations such as ad-insertion or blackout.

Regionalization Perspective

Some service providers may want to augment their existing home cable/data service with streaming capabilities for in-home IP devices. They may not being trying to stand up an OTT type of service to off-net subscribers, rather they want to offer current subscribers the ability to use their portable devices or want to deploy IP STBs with streaming capabilities. Furthermore, they plan to perform multi-rate transcoding within their existing Central Headends and will deliver the multi-rate transcoded transport streams or mezzanie files to the Regional Headends via their IP distribution networks or potentially via CDN. Just as in their legacy systems, regional customization takes place at the Regional Headends. This is where an Edge Packager could be used to perform regional ad insertion or even blackout processing.

Tier 2 and 3 service providers may also wish to offer streaming services and are sensitive to CDN content distribution costs. Similarly, they may wish to receive multirate transcoded content directly from content providers or resellers and need to perform regional customization (e.g. regional ad insertion).

FUNCTIONAL CAPABILITIES SUPPORTED BY PACKAGERS

Any discussion of ad insertion must occur in the context that viewers do not, in general, want to watch ads. They will go to substantial lengths to avoid ads, with smartphone and tablet users running clients that have been specially designed to defeat ad presentation systems. If one builds a manifest that looks like the following, you can bet that someone will find a way to avoid watching fragment three.

```
  <fragment time=1, length=10,
uri=LOTR.mp4>
  <fragment time=11, length=20,
uri=LOTR.mp4>
  <fragment time=21, length=80,
uri=http:/www.ad-decision-system.com>
  <fragment time=81, length=10,
uri=LOTR.mp4>
```

Figure 2: Easy to Defeat Manifest

Center Packaging Ad Insertion

Central packaging implies that all of the work to create fragments and manifests is completed prior to any request for content. The transcoder outputs to the packager which outputs to the origin server and that “transaction” is complete. The transaction from the client to the origin server (through one or more CDNs) is a completely separate transaction. It may not be apparent, but this is true regardless if the content is VOD, linear, or network digital video recorder (nDVR). In both VOD and nDVR there is a gap between the recording/packaging of content and its eventual playback. Even in the linear case, however, the packager is essentially filling a bucket (the origin server) while the client is emptying that bucket. This becomes more clear if the linear case is expanded to consider StartOver TV. StartOver is basically linear with a limited ability to jump back to the start of a program. In adaptive bit rate (ABR) linear TV, there is what amounts to a 30-second jitter buffer to deal with packaging and manifest creation; StartOver just expands that buffer to 30 minutes.

This implies that all manifest information for regional or targeted advertisements must be built prior to content request. This can be accomplished in two ways. First, one could create multiple manifests, one for each ad region. The session manager then needs to point the client to the region-specific manifest. It does not appear that this approach is feasible for targeted advertizing. The second approach is to build a single manifest such that the ad decision manager (ADM) or some other component is invoked at playback time.

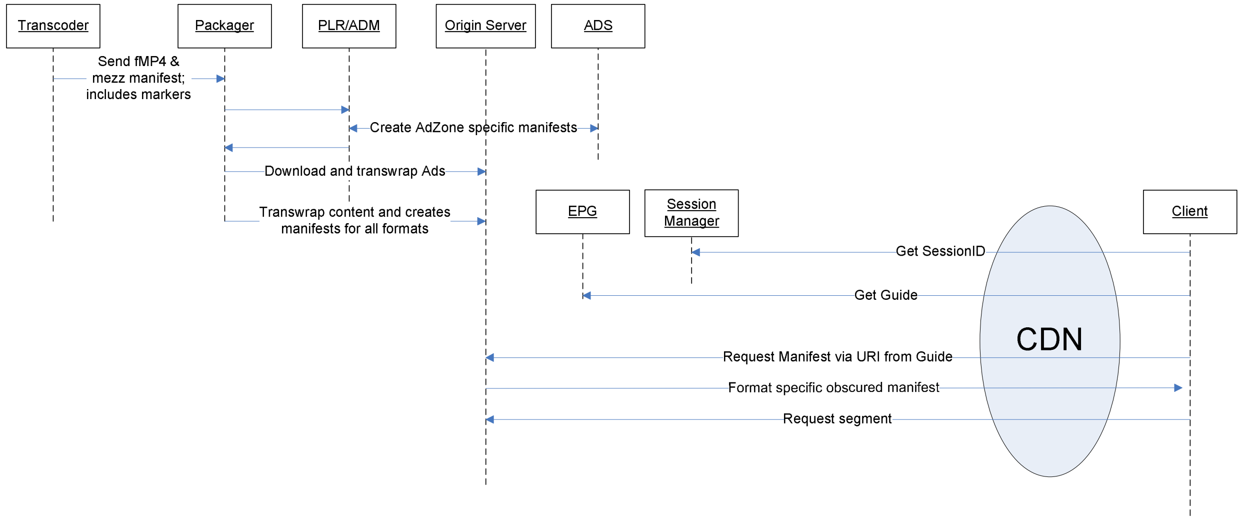
Transcoder
Packager
PLR/ADM
Origin Server
ADS
Send fMP4 & mezz manifest; includes markers
Create AdZone specific manifests
Download and transwrap Ads
Transwrap content and creates manifests for all formats
EPG
Session Manager
Client
CDN
Get SessionID
Get Guide
Request Manifest via URI from Guide
Format specific obscured manifest
Request segment

Figure 3: Center Packager for Ad Insertion

For the second approach to work, the manifest entry corresponding to an ad (or set of ad segments) must be a uniform resource identifier (URI) that can be processed by the origin server or by the ADM itself. One could imagine generating a URI pointing to the origin server that was encoded in such a way that the origin server could detect it, perform some processing on it, and issue an HTTP redirect to the ADM with some localization parameters supplied. Alternately, the URI could point to the ADM directly with some guarantee that the parameters required for localization would be directly supplied by the client. In this last case the URI cannot effectively be obscured. In addition, the client may be able to spoof the parameter(s) used to direct the system towards the targeted ads.

Edge Packaging Ad Insertion

Edge packaging defers the creation of manifests and fragments until the content is requested. This means that a session ID (targeted ads) and/or region ID (regionalized ads) are known at packaging time. Given this information, the packager can use its Play List Rebuilder (PLR) function to query the ADS during manifest creation. It can also cache those results. Since it does not have to pre-build the manifests, it can build regionalized manifests only as needed. This is a savings since it is unlikely that every program that is recorded will, in fact, be requested from all possible ad regions. Granted, manifest creation is not expensive, but managing an explosion of files that might never be used can add complexity to the overall solution.

Since manifests are only created as needed, it becomes possible to create targeted or per-user/per-session manifests. Depending on the ADM being used and the campaign that is in force, it may be desirable to create viewing experiences tailored for a single user at a particular time. While such a manifest cannot be cached since it is a onetime use artifact, at least it can be created. In the standard central packaging configuration, targeted manifests are simply not possible. It should also be observed that while the manifest for a targeted viewing cannot reasonably be cached, the content and ad fragments associated with that viewing may well be cachable.

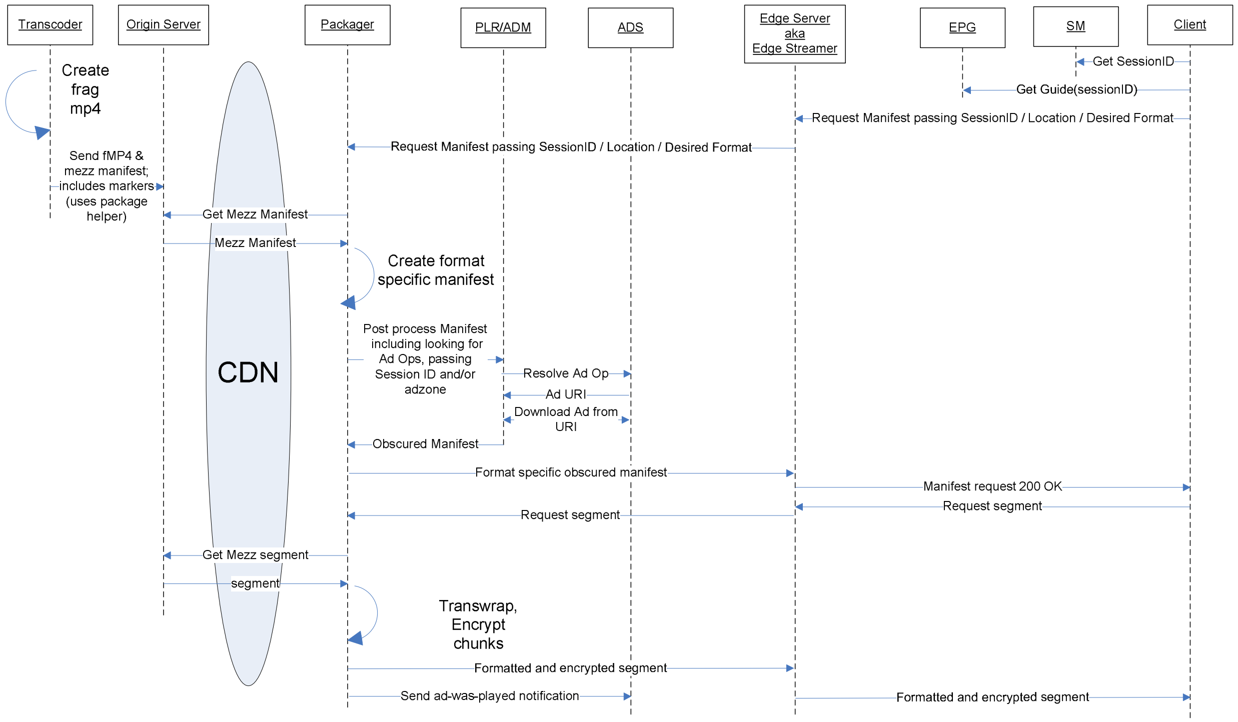

Figure 4: Edge Packaging Ad Insertion

Viewership Management

Another significant problem with Central Packaging is related to fulfillment or viewership management. As seen in Figures 3 and 5, there is no obvious way to indicate when an ad is actually played by the client. Keep in mind that the origin server is acting as a simple web server and the distinction between content fragments and ad fragments has been obscured. This appears to leave us unable to inform the ADS when the fragments corresponding to a particular ad were ever actually viewed by a client. This would seem to imply the need to put some level of intelligence in the origin server, yet the proper level of that intelligence is elusive. If a pattern to the manifest entries is declared such that the fulfillment observer could determine which fragments were actually ads, one would have to assume that the clients could detect the pattern.

Alternately one could imagine creating a back channel from the origin server back to the packager providing fragment-requested information for all fragments. The packager presumably knows which fragments are ads and could be the component to send the fulfillment message to the ADS That begins to blur the lines of function design for the packager and also makes assumptions about packager knowledge.

To avoid this one could add a component to the control plane path such that whenever a fragment was fetched by a client this new component could detect when an ad fragment was requested and inform the Ad Management Service. This control plane component would have to deal with the fact that an ad fragment might be cached by the CDN and not actually fetched from the Origin Server.

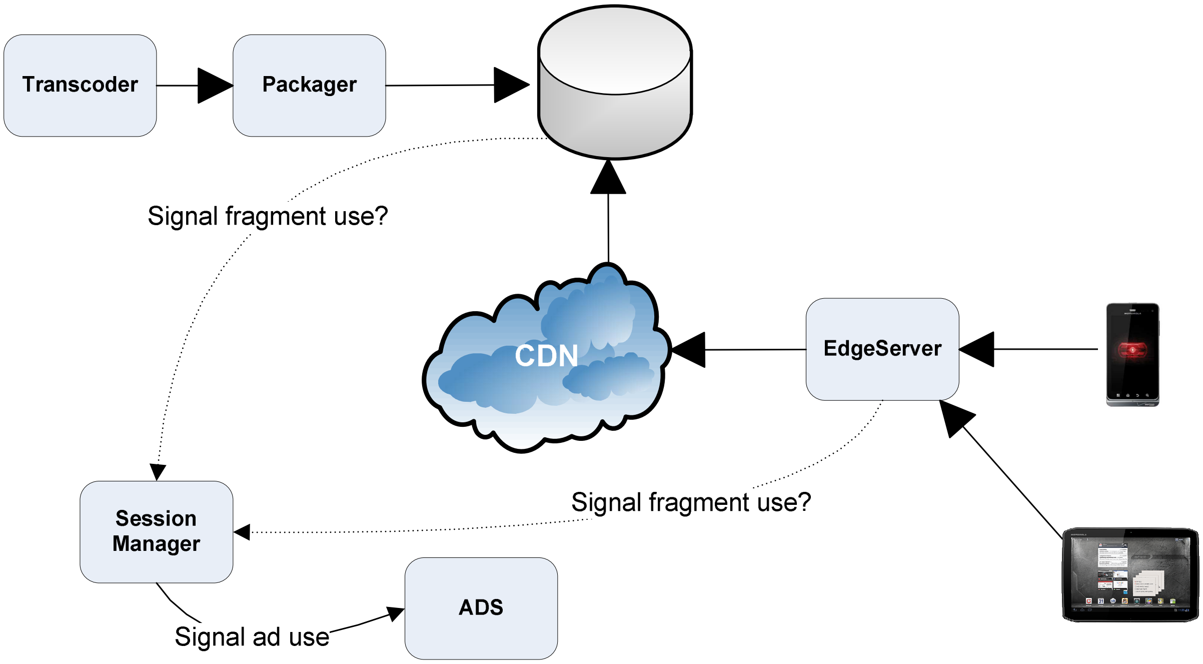

Figure 5: Center Packager Viewership Monitoring

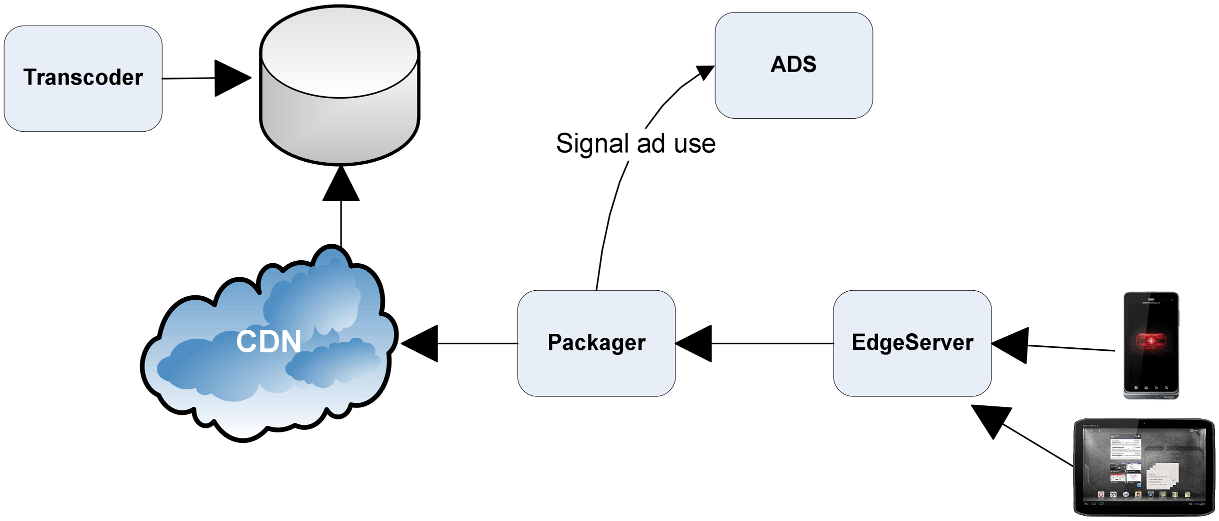

Figure 6: Edge Packaging Viewership Monitoring

Such a centralized control plane component would also seem to become a bottleneck and/or single point of failure as it would appear to need to be involved in every fragment request from every client.

Yet another approach would be to split the packager function into separate transwrapping and manifest generation components. One could then position the manifest generation component logically between the client and the origin server. This hybrid option will be discussed in more detail later

Each of these problem becomes simpler in the Edge Packaging deployment shown in Figure 6. As can be seen from the diagram the packager is involved in all fragment requests and can directly inform the ADS about fragment downloads. This is especially convenient since the Packager component is already well positioned to have knowledge of which fragments are and are not advertisements.

Ad Zone Dynamism

Late binding of manifest creation also allows for dynamism in the set of ad zones. Ad zones are only applied to manifest creation at session creation time. While this may not be a large difference for linear or nDVR applications, it can be for VOD content. VOD contents are ingested once and then may exist in the system for weeks or months. Some classic content might stay in the library for the life of the system. It is certainly imaginable that an operator might want to change the configuration of their ad zones during this time window. It's not clear that there is a way to modify ad zones in a center packaging configuration.

Time Based Ad Selection

Edge packaging also provides more options for ad selection. For example, suppose the system wants to generate a targeted ad based on the playback time of the content. An example would be a show-teaser ad suggesting that the user watch the "next" program. Except in the case of linear content, "next" will mean something different at playback time than it did at record time. If a user records a show on Monday at 7:00 PM they may receive teaser ads for the 8:00 PM Monday show on the same channel. Playing that content back on Tuesday results in a teaser ad that is, at best, useless and, at worst, defeats the viewers quality of experience. Receiving promotions for a show that you missed and cannot, in fact, watch could frustrate viewers. Therefore, the ad decision that will be reflected in the manifest should be made at playback time, not at record time.

Hybrid Packaging

A hybrid approach to packaging should be explored to round out the possibilities. As alluded to earlier, this approach involves using a central packager to perform content and ad fragment chunking. All such fragments are loaded onto the origin server, along with one or more undifferentiated manifest files.

Between the origin server and the client is another component, the transwrapper. This component or software service may be co-located with the origin server or may be a deployed on separate hardware. The intention is that the URI supplied to the client for obtaining the manifest should resolve to the transwrapper component.

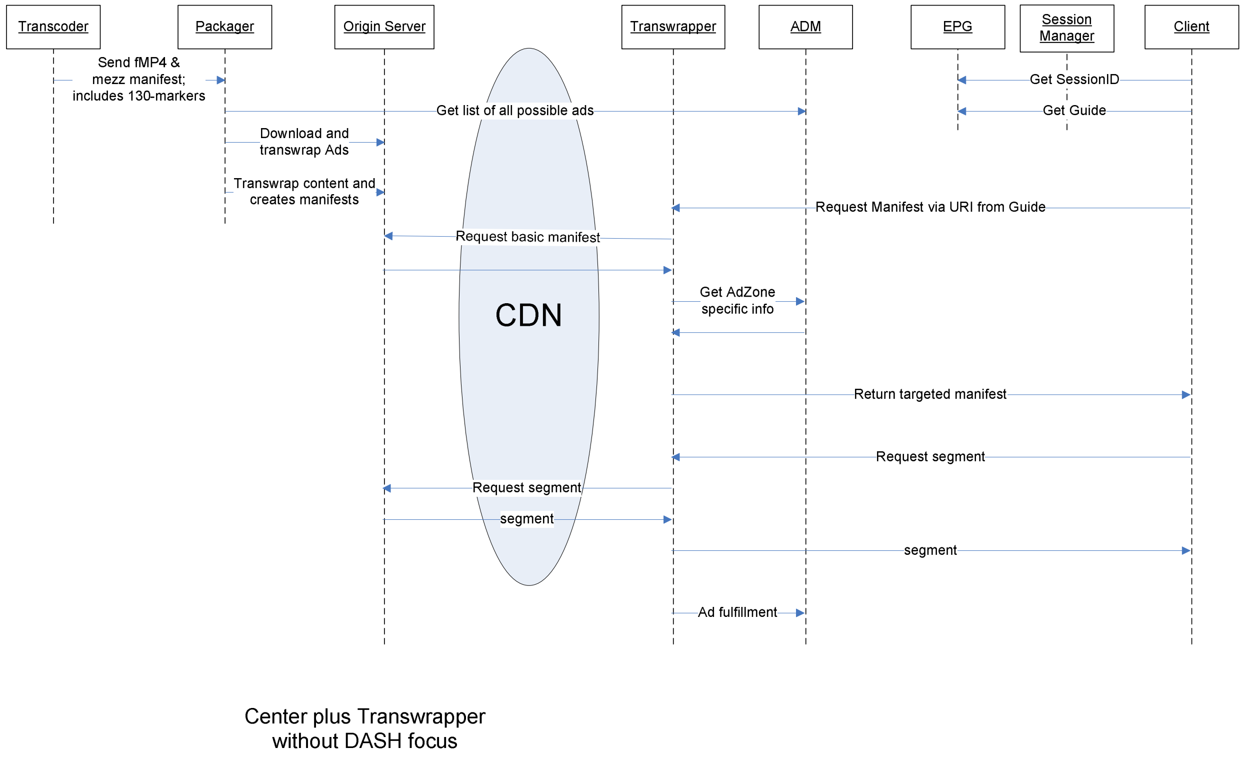

Figure 7: Hybrid Packaging

The transwrapper uses information in the client request to a) transwrap as needed to the client's requested format, and b) perform ad personalization. To do this, the client must supply a regionID for regionalized ads or a sessionID for personalized ads or equivalent data that can be resolved into an appropriate ad zone. Based on that derived ad zone, the transwrapper assembles a manifest tailored to clients within that ad zone.

At first glance, this seems like a reasonable compromise that achieves many goals. From another point of view, however, the system now has both center and edge packaging components. In other words, it's not clear that Hybrid packaging has any advantage over Edge Packaging.

CENTRAL & REGIONAL BLACKOUT APPROACHES

It is anticipated that blackout control will be a required function in multi-screen environments, wherein IP set-top boxes, in-home portable devices, and portable devices outside the home will have to be restricted from receiving content based upon their location (or the subscriber's home location) during a service substitution event. Today, blackout is enforced through the content provider's uplink control system. Normally, a retune command is inserted at the uplink and targeted to individual integrated receiver/decoders (IRDs) known to be operating within a specific region. When a specific IRD observes a retune command addressed to it, it mutes the video stream or replaces it with an alternate service for the duration of the blackout. Similar functionality can be provided in multi-screen systems by manipulating playlist/manifest files during a blackout event.

In a centralized architecture, where packaging and manifest creation is performed, a new manifest or sequence of manifest files needs to be created during the blackout event for clients within the affected region. The new sequence of manifest files will direct those affected clients to tune to alternative content for the duration of the blackout event. Through an element known as a Blackout Manager, unique regional manifest files are generated for all the blackout regions under its control. The Blackout Manager must have knowledge of the CDN topology and specifically the mapping of each edge cache to the specific geographic region it services. It is required to continuously monitor for blackout events by processing IRD retune messages for its regions. When a blackout is in effect for a given region, it requests manifest updates from the playlist rebuilding function, which are subsequently published to the CDN. The updated manifests reference new URLs that point to alternate content during a blackout for one or more affected regions.

Figure 8 illustrates the system. The blackout manager requests that the playlist rebuilder generate unique manifests for each of the three regions under its control, namely, Pittsburgh, Philadelphia, and State College. The manifest (M) is retrieved from the packager and the SportsNetwork.ServiceProvider.net/pitt, SportsNetwork.ServiceProvider.net/philly, and SportsNetwork.ServiceProvider.net/StateCollege manifests are created and published to the CDN. The content identified within the different regional manifests can be identical until such time as a blackout event is required to be enforced. At that time, the manifest file for that area is modified by replacing the blacked out content URLs with URLs for the content to be substituted.

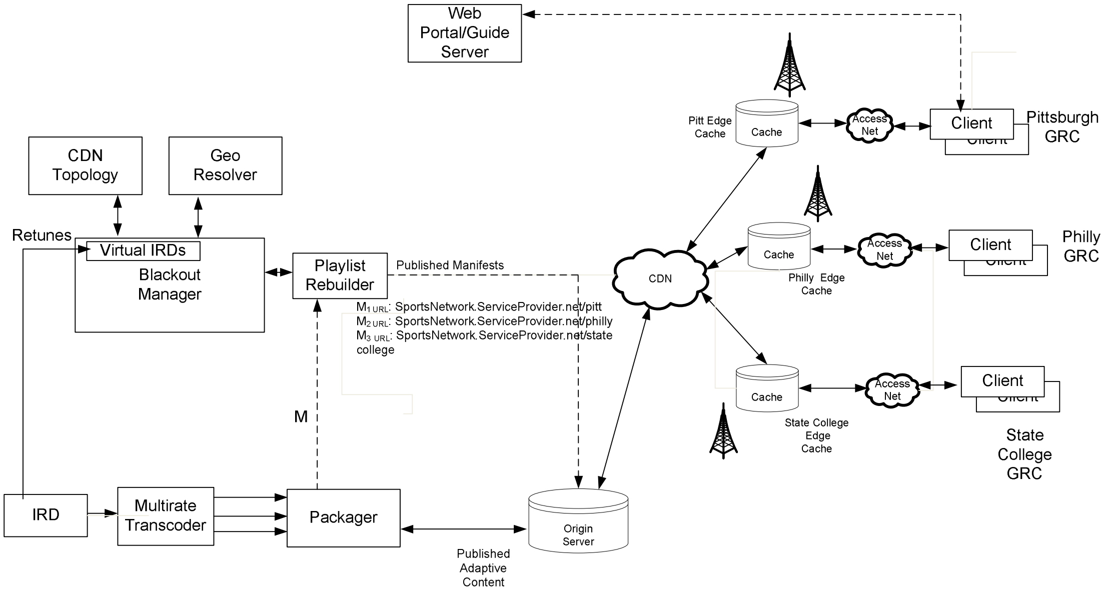

Figure 8: Centralized Blackout Management

A particular client within a given blackout region can retrieve the right manifest for that area through a number of techniques listed here:

1) Client GeoLocation: Client geolocates itself using embedded GPS technology or geo position services available on the network. The client reports its location to an upstream control plane element—for example, a session manager—which, in turn, identifies the appropriate URL for the manifest associated with that region. Alternatively, the client can construct an HTTP request that includes location metadata, which results in the return of a location-specific manifest.

2) Control Plane GeoLocation: The client is geolocated by control plane elements within the network. For example, a session manager that the client communicates with could use a geoLocation service to resolve the client's source IP to a location within the network. The session manager, in turn, identifies the appropriate URL for the manifest associated with that region to the client.

3) Edge Network GeoLocation: Edge network elements append location metadata into a client's HTTP request relying on the network's knowledge of where HTTP requests entered the network.

Each of these options has advantages and disadvantages as described below. For the first option, it is necessary to have clients that can perform geo location processing. Without this capability, they would be unable to request a manifest for their particular GRC and would likely receive the most restrictive manifest (blackout area), even if they were not located within a blacked out area. It's also not hard to imagine the development of downloadable applications that will allow clients to spoof their actual location within an HTTP request for content.

In the second option, control plane elements are in the critical path of determining the client's location at all times. This is particularly difficult if the client is mobile and is crossing different blackout zones. Each time the client enters a new zone it must be detected by the control plane elements so that a new manifest for that zone can be delivered.

The third option is the most ideal way. Here, a function exists within edge distribution network elements (e.g., CDN cache) that can append location-specific metadata into the client's HTTP request for the manifest file. For example, if Service_Provider is the subscriber's service provider, and if the subscriber is trying to acquire the SportsNetwork broadcast, the guide/navigation function would provide the SportsNetwork URL, SportsNetwork.ServiceProvider.net/index. The location- specific metadata would be inserted within the access network or at the boundary (edge cache) between the access network and CDN ingress point, so as to accurately identify the physical location of the client. This would result in a modified URL, SportsNetwork.ServiceProvider.net/pitt/index. This is a simple and reliable method that even works if mobile clients cross GRCs dynamically. As the client moves in and out of different access points, the network elements at the edges of the network add location-specific metadata that becomes part of the request, resulting in the return of the appropriate manifest. Of course this approach requires this functionality to be incorporated into access networks or CDNs in a standardized fashion.

An alternative that models today's blackout system solutions uses the packager to enforce blackouts within a given region. As illustrated in Figure 9, edge packagers are located within the different blackout regions.

Each live packager is configured to assume a virtual IRD identity. The IRD retune messages received from the satellite downlink are carried within metadata that is distributed to all the packagers in the regions. Each packager filters for retune messages destined for its VIRD identity. A live (edge) packager that observes a retune message addressed to it will update the manifest it is creating with URLs that point to alternate content during the duration of the blackout.

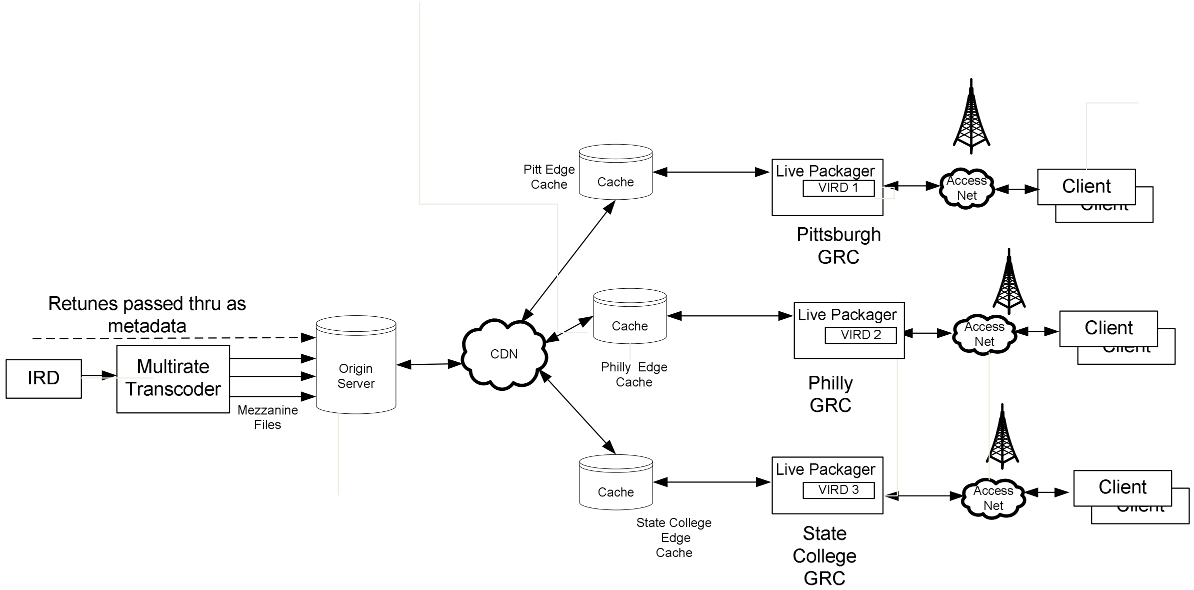

Figure 9: Edge Blackout Architecture

The following is a summary of the advantages and disadvantages provided by the centralized and edge/regional blackout solution options:

Centralized

Advantages

- All operations are managed centrally.
- All content processing equipment can be co-located.

Disadvantages

- A blackout management function is required that understands the GRCs it manages and CDN topology.
- A scalable playlist rebuilding function is required.
- Network changes may be required to append location information to HTTP requests or higher level managers may be required to point the client to the appropriate manifest URLs

Edge

Advantages

- There is no need for a centralized blackout management function.
- There is no need for a centralized, highly scalable playlist rebuilder function.

Disadvantages

- Requires deployment of edge packagers.
- During BO, content replacement has to be facilitated at the edge packagers.

CONCLUSIONS

The choice of where to locate the packaging function is a complex one with implications far beyond capital expenditure decisions such as how much hardware to purchase and where to locate it. Previous discussions of central versus edge packaging have focused on the costs of network infrastructure. There are however many other important factors that should be considered. The packager has been thought of as a simple component that provides chunking and creates manifests. As has been described in this paper the truth is that many of the high value features of the overall system are highly dependent on the packager configuration. Ad insertion and blackout control are two examples of such high value features.

As an operator, do you want to support generalized ads, regional ads, or targeted ads? How concerned are you about the race between ad providers and ad-defeating clients?

Are you focused exclusively on one of several possible subsets of the viewing experience (VOD, linear, nDVR, ABR, multi-screen)? Depending on the subset, the problem space changes. One may want to consider a phased approach or look at the entire problem when planning a deployment.

Of course, the choice may not be between center or edge packaging. It may well be between center and Hybrid packaging, and edge packaging. One could well argue that simply using edge packaging is the simpler solution.