LIVE BROADCAST DISTRIBUTION WITH REALSYSTEM SERVER 8

RealSystem iQ Whitepaper

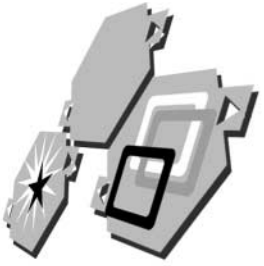
7 December 2000

RealNetworks, Inc.
2601 Elliott Avenue, Suite 1000
Seattle, WA 98121
U.S.A.

http://www.real.com
http://www.realnetworks.com

Printed in the United States of America.

CONTENTS



LIVE BROADCAST DISTRIBUTION

This whitepaper highlights the features and benefits of RealNetworks® RealSystem Server 8 (RealServer® 8) live broadcast distribution technology for both Internet and enterprise environments. Live broadcast distribution is one feature of Neuralcast Technology introduced with RealServer 8. Neuralcast Technology is a family of features that enables intelligent media delivery through distributed networks:

- NeuralCast Live Distribution

  Live distribution is the intelligent and reliable delivery of broadcasts through a network of RealServers. Current abilities include multiprotocol transmission between servers, error correcting methods for streams, and terrestrial and satellite multicast support.

- NeuralCast Communications Protocol

  This is the ability of a network of RealServers to exchange information and make decisions. Current abilities include capacity sharing and capacity fail-over in the event of a network or equipment outage.

- NeuralCast Live Redundancy

  From encoder to server or from server to server, this is the ability to send redundant streams, providing a fail-over feed in the event of a network or equipment outage.

Audience

This whitepaper is targeted at system administrators, system architects, and technology managers interested in large-scale broadcasting of live streaming media live broadcasting over the Internet or across private-enterprise WAN and LAN intranets.

This document assumes a working understanding of network architecture and IP protocols, as well as familiarity with existing streaming media technologies

developed by RealNetworks. For a better understanding of these technologies, see the links to other online resources at the end of the document.

For more information on the configuration and usage of live broadcast distribution, please refer to the *RealServer Administration Guide* and the release notes accompanying RealServer. The RealServer release notes are frequently updated. You can find URLs for these documents in the section "Additional Resources" on page 26.

Introduction to Live Broadcast Distribution

RealServer 8 introduces new technology for the distribution of live broadcasts from originating (transmitting) RealServers across both terrestrial and satellite IP networks to receiving RealServers. This technology makes possible a new paradigm for scalability and reliability in broadcasting streaming media live.

RealServer 8 live broadcast distribution replaces live splitting mechanisms that have been supported by RealServer in various forms since version 3. RealServer 8 is capable of live broadcast deployments that can be scaled to reach thousands, or even millions, of clients simultaneously.

Specifically, this newest version of RealServer offers the following advantages:

- Server-to-server multicast and unicast transmission of live broadcast streams.
- Unidirectional distribution, enabling large-scale broadcasting of live streamed media events over Internet Protocol (IP) satellite links.
- The elimination of constraints on CPU and network bandwidth that are inherent with persistent Transmission Control Protocol (TCP) connections between live broadcast receivers and transmitters that existed in earlier versions of live-splitting technology.
- Ease of transmission through firewalls, as outbound User Datagram Protocol (UDP) traffic requires no return connection to the originating transmitter.

As for reliability and fault tolerance, RealServer 8 is dynamic and flexible in several key respects:

- It enables redundant network paths to distribute mission critical broadcasts across vulnerable and lossy network segments.

- It fortifies the live distribution data channel against network packet loss by implementing configurable in-stream forward error correction (FEC).
- It enables a live broadcast to be relayed by redundant RealServers located on either the same network or different networks, circumventing single points of failure in the distribution chain.

Essentially, RealServer 8 represents a whole new architecture for live broadcasting. Multicast transmission between RealServer transmitters and receivers simplifies RealServer deployments across networks while also scaling to reach as large an audience as necessary.

RealServer 8 leverages existing network infrastructure to achieve live broadcast distribution redundancy, negating the need to purchase additional hardware or software to facilitate live broadcast distribution.

Live Broadcast Distribution Defined

RealServer 8 broadcast distribution is a new method of transmitting live broadcast data from an originating RealServer (a *transmitter*) to a receiving RealServer (a *receiver*). The transmitter acquires a live feed from RealProducer (the *encoder*) or from RealNetworks Simulated Live Transfer Agent (SLTA). The transmitter is configured to distribute content from the live source to one or more receivers, with almost no upper limit on the number of receivers.

The receiver is configured to listen for live broadcasts that originate from the transmitter. As live broadcast packets arrive at the receiver, the receiver fulfills requests from specific RealPlayer® clients. From a single distribution channel, the receiver "splits" the broadcast and serves it to the connected clients.

You can initiate live broadcast distribution *channels* either in a *push* manner, where the transmitter initiates the packet flow to the receiver, or in a *pull* manner, where a client request arriving at the client (receiver) triggers the transmitter to begin sending data packets.

Features and Benefits

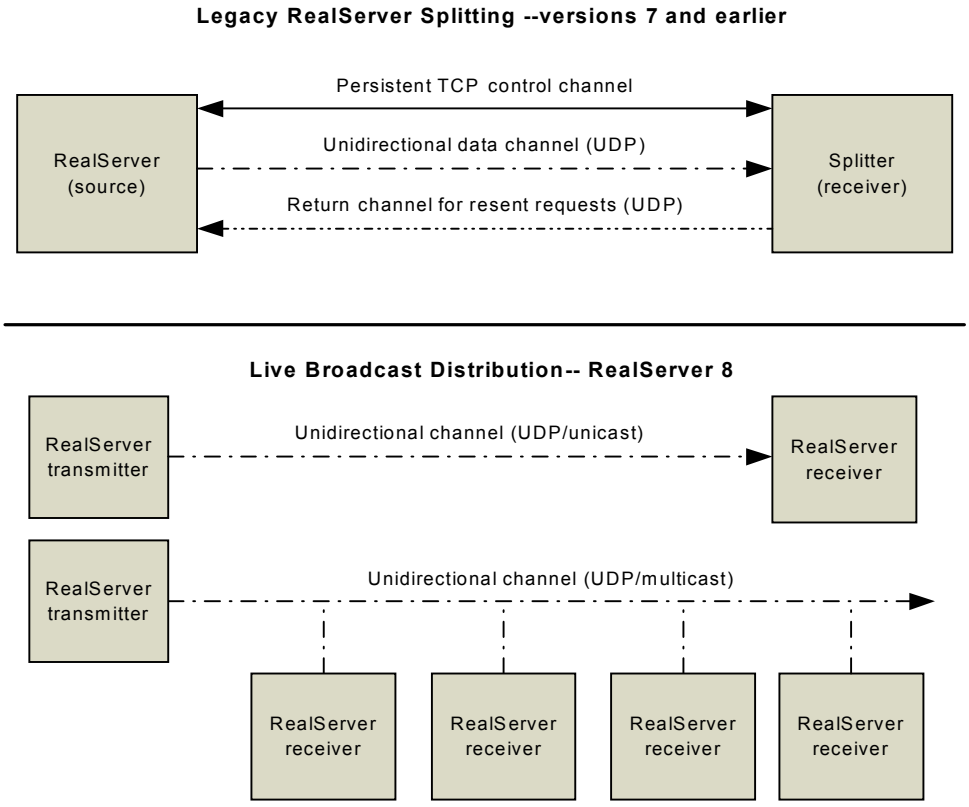
Scalability

RealNetworks' live broadcast distribution technology is the most scalable means currently available for delivering streaming media to vast Internet audiences.

Optimization for Connectionless Transmissions

Unlike earlier versions of splitting technologies, live broadcast distribution is optimized to write streaming media packets over a network in a unidirectional manner that does not require acknowledgment of arrival from the packet receiver. Previous implementations of splitting required a persistent TCP control channel between the transmitter and the receiver. By default, RealServer 8 broadcast distribution requires only a single one-way network path from the transmitter to the receiver. If connectivity between the transmitter and receiver is interrupted subsequently restored, the live distribution will continue without needing to be restarted and without any other action required on the part of either the transmitting or receiving server.

To accomplish this, the transmitter inserts all of the information the receiver will require to process the live distribution into the one-way packet flow. The necessary session description, stream data, and error correction data flow across the network to the receiver, so there is no need for an out-of-band channel.



Point-to-Point Broadcast Distribution

In nonmulticast environments, UDP transmission is made over a unidirectional connection between transmitters and receivers. This method does not incur the overhead resources involved in of maintaining a persistent two-way connection. Because the UDP unicast transmission of RealServer 8 broadcasts does not require a back channel, most firewall devices will not restrict live transmissions that originate behind a firewall and send data packets out to remote receivers.

For highly reliable LAN segments, where the overhead of a persistent two-way connection can be maintained, you can use TCP as the transport protocol. It is important to note that the inherent reliability of TCP causes a degree of latency that is not beneficial to streaming media. Broadcast distribution involves several methods, discussed in the remainder of this document, that

are more effective than TCP in overcoming packet loss under conditions in which timely delivery of streamed content is critical.

Point-to-Multiple-Point Broadcast Distribution

Multicast transport for live distribution enables one transmitter to reach a virtually limitless infinite number of receivers while using the minimum amount of bandwidth to traverse transmitter-to-receiver network segments.

Beyond offering unbounded scalability over terrestrial networks, multicast transmission makes IP broadcasting of streaming media possible over satellite networks, where a back-channel connection initiated by a receiver is not possible.

Reliability

Forward Error Correction

To fortify live broadcast packets against loss occurring because of network drop-outs, RealServer 8 employs a configurable forward error correction (FEC) method. This enables receivers to reclaim—in real time—packets lost during transmission, without having to initiate a return channel to the originating transmitter.

The percentage of FEC data embedded in the data stream is configurable, enabling system administrators to balance the amount of bandwidth used versus the reliability of the network segments that exist between the transmitter and the receiver.

Redundant Distribution

Webcasts are now capable of reaching audiences in the millions; thus, the need to ensure live broadcast reliability is the overriding requirement for content-delivery networks. By design, broadcast distribution offers system administrators many deployment options. These options enable administrators to distribute a live channel in such a way that there are no single points of failure as live packets traverse network segments between the transmitter and the edge receiver that is responsible for fulfilling client requests.

Because many of the features of broadcast distribution can be configured dynamically, live distribution streams can be added (and removed, if there is a

redundant stream) without interrupting other traffic between RealServer transmitters and receivers.

Resend Requests

When bidirectional connectivity can be established and maintained between the transmitter and receiver, you can configure the receiver to automatically request lost packets on an as-needed basis. This optional resend connection uses either UDP or TCP, regardless of which live distribution transport protocol is being used for streaming data. The resend request information is returned to the transmitter over a unicast connection, even if the live data arrives at the receiver through a multicast.

You can use FEC and resend requests in tandem to provide packet-delivery fault tolerance for any network whose topology permits two-way connectivity between broadcast nodes.

Dynamically Routing Around Failure Points

You can use network routers to establish alternate paths for packet traffic. When a router detects a disruption on the primary path, it automatically reroutes packet traffic along the alternate path. This way, live broadcasts are dynamically detoured around failure points without needlessly consuming bandwidth for redundant streams.

Note that to use this bandwidth-saving method of stream delivery, you do not need to purchase any additional streaming media products. The same routers that are used to forward all IP traffic can be employed to forward data packets around points of failure whenever necessary.

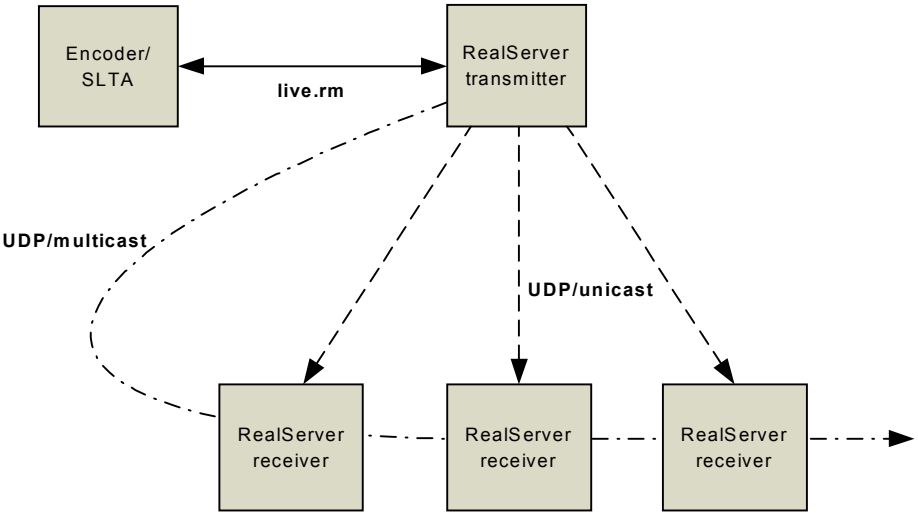
Stream Reconvergence

Whenever a live broadcast is sent redundantly over multiple networks, it can always be reconverged at the edge receiver with no adverse effect on the RealPlayers being served from the receiver. The receiver simply uses the first packet that arrives and automatically discards subsequent packets that contain replicated data. The receiver examines the data independent of its source address, the transport protocol used, or the network that it traversed.

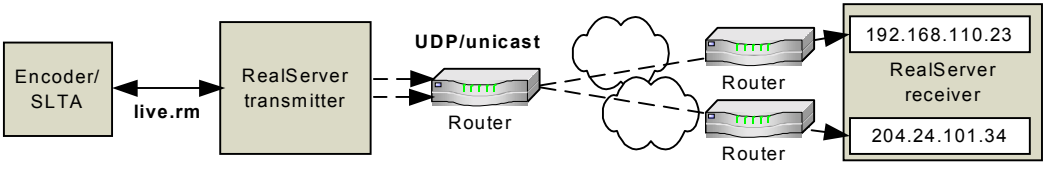
The same is true in cases where live packet traffic switches routes in mid-broadcast. When this happens, the receiver continues to serve the broadcast to clients successfully even though the packet traffic route has changed.

The rest of this section discusses just a few of the many reliable distribution deployments that you can use. The following scenarios and illustrations should prove useful:

- A multicast and unicast distribution is established from one transmitter to multiple receivers, such that a single live broadcast can be sent from the transmitter to all of the receivers concurrently, as shown in the following diagram. The use of different network paths helps ensure the successful transmission of the live packets to their destinations.

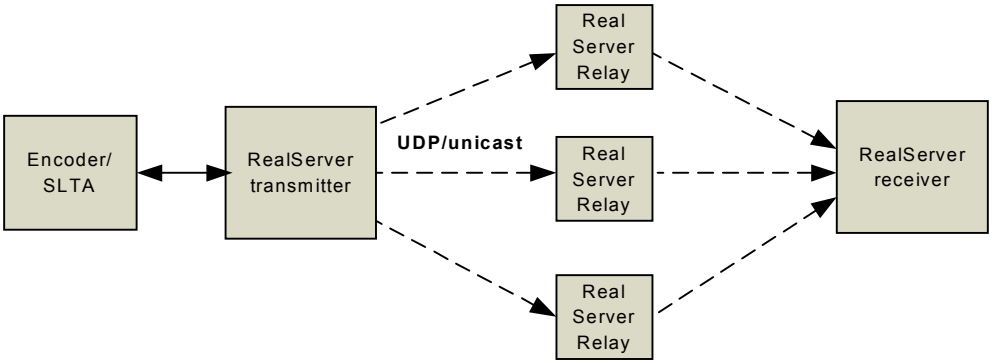


- Network routers send the same live broadcast to multiple interfaces (actual or virtual) on the host receiver, as shown in the following diagram. Note that the network route used for the first interface on the receiver is different from the route used for the second interface. You can configure routers to automatically send packet traffic by way of a secondary network path only when a disruption has occurred along the primary path.



- Another option is to use an intermediate tier of specially configured transmitters to relay the same live broadcast to an *edge receiver* (a receiver that is adjacent to the clients in a broadcast deployment configuration), as

shown in the following diagram. If one relay suffers a power outage, the live distribution is seamlessly supported by other parallel relays. Because the live broadcast is carried by parallel relays to the edge receiver, not only is there no disruption of packet delivery to the edge receiver, but all connected clients continue receiving the same quality of transmission, with no perceptible interruption or rebuffering.



Security

RealServer 8 live broadcast distribution offers an optional security feature that, when enabled, ensures that receivers are entrusted to process (and serve to clients) any live distributions that they split. This security feature does not prevent packets from being sent; rather, it inhibits rogue receivers from successfully parsing (and serving to clients) live transmissions. Like all features of broadcast distribution, this security mechanism is designed to function over a unidirectional connection, regardless of the data transport protocol being used.

The system administrator establishes a common security type and password on both the transmitter and the receiver. When the transmitter sends live packets, a security token is embedded in each packet that the receiver parses as it processes the inbound streaming data.

Deployment Options

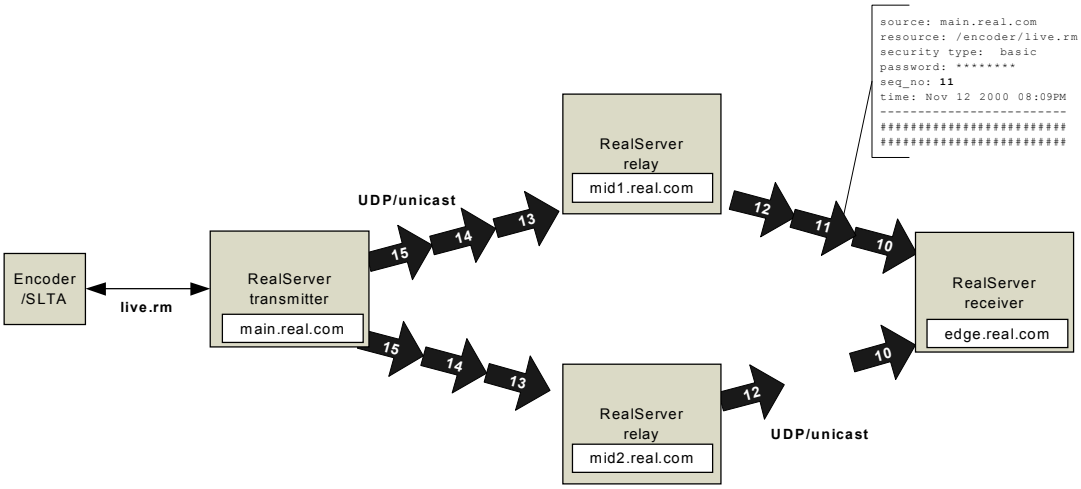
Flat Network Deployments: Using Multicast

The advent of multicast transmission of live broadcasts between RealServers may act to flatten existing splitter deployments. Multicasting offers the most

scalable means yet developed for reaching vast numbers of edge receivers while consuming the smallest possible amount of bandwidth.

Tier Network Deployments: Using Relays

In network configurations that span non-multicast-enabled segments, a deployment of relays allows for the retransmission of live streams by hosts that are "down-network" of the transmitter. As mentioned earlier in this whitepaper, relays make tiered redundancy possible by repeating the same broadcast over multiple intermediary hosts. The edge receiver simply discards any redundant stream packets regardless of where they were sent from, thus removing the single-host dependency in the middle of the network.



You can also deploy relays at a network "peering point" or at a gateway, where because of differences in network topology or administration, it is a good idea to use one or more relays to the manage the access to and egress of live distributions that cross an administrative boundary.

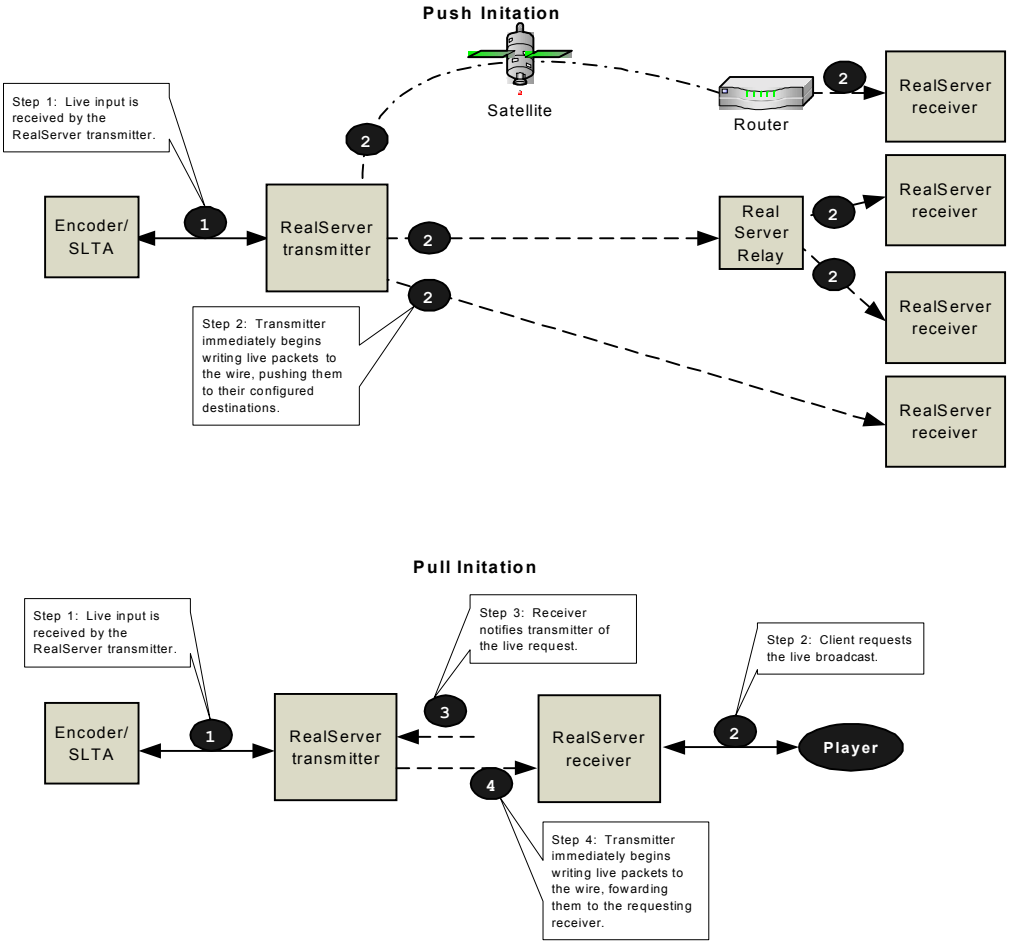
Relays provide administrators the ability to alter communications attributes while maintaining the integrity of the streaming data. A relay can retransmit the inbound source using a different transport protocol, different error correction parameters, and a different security type and password.

Having two or more relays push a live distribution broadcast to a single edge receiver removes a single point of vulnerability at a middle tier. The edge receiver simply discards any redundant packets that it receives for the same

broadcast, regardless of what relay, transport protocol, and network path were used to deliver the live broadcast.

Push and Pull Initiation

As with previous incarnations of splitting, live broadcast distribution can be initiated either by a packet push from a transmitter out to receivers (possibly through relays) or by a pull request sent from a client to a given receiver that is configured to receive live broadcasts from an originating transmitter. Both of these scenarios are illustrated in the following diagram.



Every aspect of push initiation is unidirectional, whereas pull initiation requires a one-time request made by the receiver to the transmitter to trigger

packet transmission. Because both forms of live broadcast distribution are based on the same technology, both methods employ FEC, optional security, and configurable transports, among other features. Note that each method offers its own unique advantages.

The following are some of the distinctive characteristics and benefits of push initiation:

- After a stream arrives at a receiver, there is no additional startup latency.
- It supports UDP/unicast, UDP/multicast, and TCP packet transports.
- With UDP/unicast and UPD/multicast transports, all communications are unidirectional.
- In tiered deployments, it uses relays.

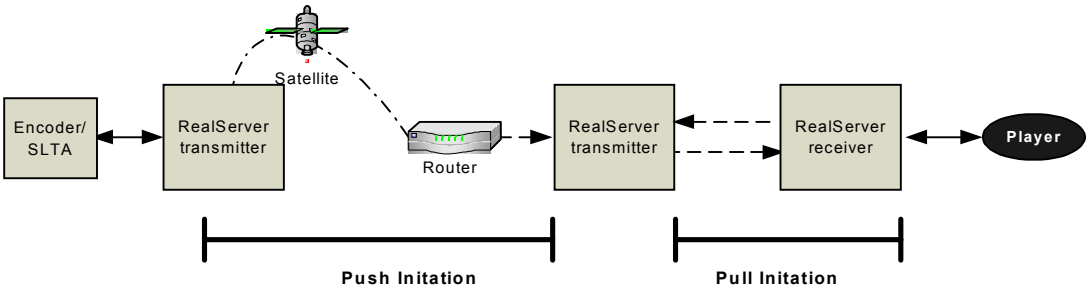
By contrast, the following characteristics are unique to pull initiation:

- It conserves bandwidth by not pulling the live stream to the receiver until the first time a client requests the stream.
- It supports both UDP/unicast and TCP packet transports.
- In tiered deployments, it uses a chain of pull-enabled transmitters.

When a relay is deployed between a transmitter and a receiver, push initiation begins the packet flow, and the relay host does not appear in the URL. Pull initiation is not used when a relay exists between the transmitter and the receiver. To produce an environment of multitiered pull splitters, you can deploy a chain of pull-enabled transmitters.

The client URL syntax used for the push method of initiating broadcast distribution is different from that used for the pull method. In the push model, the originating transmitter of the stream does not appear in the URL. In the pull model, however, the transmitting host does appears in the URL. For URL syntax definitions and specifics on how to configure push- or pull-initiated live broadcast distributions, see the *RealServer Administration Guide*.

You can use push and pull broadcast distribution initiation in conjunction with one another wherever a mid-level transmitter is configured to receive a live stream from a transmitter and is also configured to send pull requests that are requested by an edge receiver.



Latency and Bandwidth Considerations

In determining how long it will take a receiver to acquire a live split stream from a transmitter (see the following diagram), you should consider the following:

- The degree of network latency that exists between the transmitter and the receiver.
- The configurable interval used to send session announcements.
- The configurable buffer established at the receiver.

The receiver does not begin processing the incoming data until it receives the session description from the transmitter. If the session announcement-forwarding interval is set to say, 30 seconds, the receiver will require a maximum of 30 seconds before being able to begin processing live broadcast packets. This time delay does not include any network latency that may exist between the transmitter and the receiver.

As soon as the live distribution stream arrives at the receiver, the receiver constructs a local buffer before streaming data to the connected clients (you can set the duration of the buffer; 6 seconds is default setting). This introduced buffer latency occurs only if a client happens to connect to the receiver at the same time that the receiver is reading the packet containing the session announcement information.

In cases where the live distribution was pull initiated by the receiver, the receiver stores only a 3-second buffer. The receiver then requests the live distribution (and its announcement information) immediately.

Single-rate broadcasts consume bandwidth according to:

- The encoded bit rate of the live stream.

- The percentage of overhead (approximately 10 percent) for session announcement information and TCP/IP headers.
- The percentage of overhead for the configurable FEC rate.

For example, a single-rate 100-Kbps stream with a configurable FEC of 10 percent will consume approximately 120 Kbps of bandwidth.

Determining the encoding bit rates for SureStream™ broadcasts is a bit more complex. As a rule, the bandwidth consumed by a SureStream broadcast equals the combined bit rates of all the audiences *plus* the overhead percentages mentioned a few paragraphs earlier.

Feature Matrix

The following table shows how a number of associated RealSystem technologies interoperate with live broadcast distribution. For descriptions of and installation and configuration information about any of the features listed here, see the pertinent product documentation.

RealServer feature	Works with broadcast distribution?	Notes
Distributed Licensing	Yes	You can allocate steam capacity to either the transmitter or the receiver from another RealServer that is enabled to support this feature.
Broadcast Redundancy	Yes, at the transmission source	This feature works at the transmitter to alias one or more backup sources for a live broadcast.
Live Archiving	Yes, at the transmission source	You cannot archive live distribution streams at the receiver. Instead, live archiving is performed on the host transmitter, saving content from the live source.
Splitting to RealProxy	No	RealProxy does not use live broadcast distribution technology to acquire live split streams from RealServers.
Back-channel and scalable multicasting to RealPlayers	Yes	You can broadcast live distribution streams to multiple RealPlayers using any transport protocol supported by RealServer.

(Table Page 1 of 2)

RealServer feature	Works with broadcast distribution?	Notes
Data types supported		Live broadcast distribution supports all of the live data types supported by RealSystem.
• Single-rate RealAudio and RealVideo SureStream	Yes	
• RealAudio and RealVideo	Yes	
• Apple QuickTime Live	Yes	
• MP3	Yes	

(Table Page 2 of 2)

System Components

Broadcast distribution is a feature of RealServer 8. The RealServer core engine instantiates this functionality by using two shared objects (.so extension) or dynamic-link libraries (.dll extension), as in the following:

	UNIX	**Windows**
Broadcast Distribution Plug-in	`bdstplin.so.6.0`	`bdst3260.dll`
Broadcast Receiver Plug-in	`brcvplin.so.6.0`	`brcv3260.dll`

Like other features of RealServer, broadcast distribution is configured in an XML configuration file (rmserver.cfg). You can also configure the feature by using the RealSystem Administrator, a Web-based user interface that is installed with RealServer.

You enable broadcast distribution features by using the appropriate RealServer license key, which you can obtain from RealNetworks, Inc. For information on the installation, configuration, and administration aspects of using RealServer broadcast distribution, see the *RealServer Administration Guide*.

Sample RealServer 8 Deployment Configurations

There are many ways to deploy RealServer 8 for live broadcasting. Therefore, system administrators must consider various aspects of RealServer deployment, including the following.

Transport Protocol

Distribution channel transport protocols are often determined based on the characteristics of the networks that the live data will need to traverse. When setting up live broadcast distributions, network administrators should ask themselves the following questions: Which network segments are multicast-enabled? Do routers or firewall devices involve any transport protocol restrictions?

Reliability

Error-correction methods should be selected with both the network latency and the anticipated degree of network congestion in mind. If a given network is very reliable and packet loss is known to be rare, there may be little or no need for forward error correction. Also, it might be easy to maintain a back channel between the receiving RealServer and the transmitting RealServer, providing a handy alternative to writing FEC-fortified packets over a network.

If however, the live broadcast must traverse the Internet through one or more congested interconnections, a high FEC level enables the receiver to reconstruct lost packets. In general, to fortify the live transmission against loss, you should set FEC according to the amount of bandwidth the broadcaster is willing to consume.

For unicast traffic that can be sent between the receiver and the transmitter, you can use resend channels as an additional means of packet recovery in case packet loss exceeds the FEC threshold.

Failover

Any broadcast that uses a connectionless transmission (either UDP/unicast or UDP/multicast) will be routed around network outages dynamically as long as internetwork routers are properly configured. You can add even more protection against network outages by redundantly transmitting the same live broadcast over different networks and then reconverging the data streams at the edge receiver. Redundant paths for the same broadcast offer insurance against outage on any single network, however additional bandwidth consumption needs to factored when using parallel transmissions to achieve greater reliability.

Note

The sample deployments described in the remainder of this whitepaper are not the only ones possible, by any means. They are provided here merely to illustrate some of the ways in which you can configure live broadcasting distribution in RealServer 8.

Large-Scale Broadcast Distribution over the Internet

This section discusses the deployment goals, deployed technologies (products), and network topology involved in delivering live Internet broadcasts, or Webcasts, over large-scale networks using RealServer 8.

Deployment Goals

- Reach the widest possible audience while consuming the least possible amount of bandwidth.
- Serve connecting clients from disparate edge locations while acquiring live signals at a managed broadcast operation center.
- Use the Internet-wide potential of broadcast distribution to reach remote edge RealServers.
- Leverage router capabilities to tunnel IP multicast traffic over nonmulticast network segments.

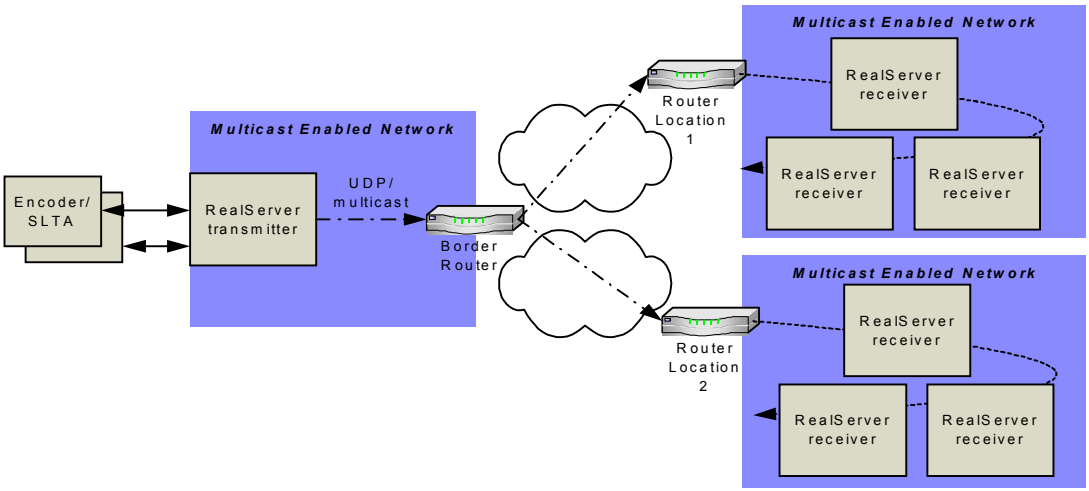
Deployed Technologies

- RealProducer 8
- RealServer 8
- Network routers (layer 3) configured to establish generic routing encapsulation (GRE) tunnels for non-multicast-enabled network segments

Network Topology

- RealServers broadcasting to clients are located at remote locations.
- Connectivity to these RealServers traverses the Internet over a combination of satellite and terrestrial networks.

- RealServers receivers are deployed at multicast-enabled locations far from the RealServer transmitters.
- RealServer operators have access to and can configure the critical internetwork routers located nearest to the RealServer transmitter and the RealServer receivers.



RealServer 8 Live Broadcast Configurations

This section presents the main live broadcast feature configurations for RealServer 8 (there are, of course, many other configuration possibilities). Included in the tables are redundancy and splitting scenarios for both transmitter and receiver hosts.

Live Broadcast Transmitter

This table describes some of the key RealServer feature configurations for transmitter hosts.

Broadcast redundancy	Configuring this is optional, but doing so ensures that live signal acquisition will continue if a failure occurs on the encoder host system or on the network to which both the encoder and the RealServer are connected.
Splitting: transmitters	A single receiver is configured to push the live source both to a specified multicast address and to a predefined range of UDP ports. The predefined transport protocol used for the broadcast is UDP/multicast. The multicast time-to-live (TTL) needs to be set to at least one "hop count" greater than the number of hops required to reach the most remote edge receiver. Because the live data will be traversing the Internet, it is recommended that FEC be set as high as possible, depending on the bandwidth constraints in each case. The Honor Resend Request option should also be enabled. This provides an additional means of packet loss recovery for receivers that are able to send unicast traffic back to the transmitter. The security type and password should be configured according to the policies and procedures of the broadcaster.

Live Broadcast Receiver

This table describes some of the key RealServer feature configurations for receiver hosts.

Splitting: receivers	A single transmitter entry is configured to receive the live source from a specified multicast address. It is also configured for the same defined range of UDP ports that was established on the transmitter. The predefined transport protocol used for the broadcast is UDP/multicast. If the receiver's network allows for resend requests to be sent back to the transmitter, make sure that the Resend Request option is enabled. The security type and password should be configured according to the settings established transmitter. By default, the receiver uses a buffer size of 6 seconds. You should increase this to give resend reports more time to initiate packet resends over Internet segments that likely have a high degree of latency.

Router Configurations

The following Internetwork operating system (IOS) excerpts— from a Cisco 3600 router—illustrate the establishment of a GRE tunnel between two routers.

```
... Begin tunnel configuration on the router nearest the receiver.
!
ip multicast-routing
!
!
!
interface Tunnel0
 ip unnumbered FastEthernet0/0
 ip pim dense-mode
 tunnel source 192.168.212.1
 tunnel destination 192.168.211.254
!
interface FastEthernet0/0
 ip address 192.168.212.1 255.255.255.0
```

```
 ip pim dense-mode
 speed 10
 full-duplex
!

... End tunnel configuration.

... PIM neighbor configuration, with the router nearest the transmitter.

router1#show ip pim neighbor
PIM Neighbor Table
Neighbor         Interface         Uptime/Expires    Ver  DR
Address                                              Priority
192.168.211.254   Tunnel0          00:07:35/00:01:39 v2    1     (BD)

... End PIM neighbor configuration.


Router configuration for the router closest to the receiver.
Add the following lines to your configuration:
[...Some lines above this.]
!
ip multicast-routing
!
!
!
interface Tunnel0
 ip unnumbered FastEthernet1/0
 ip pim dense-mode
 tunnel source 192.168.211.254
 tunnel destination 192.168.212.1
!
interface FastEthernet1/0
 ip address 192.168.211.254 255.255.255.0
 ip pim dense-mode
 speed auto
 full-duplex

PIM neighbor on the remote end, with the router nearest the receiver.

router2#show ip pim neighbor
PIM Neighbor Table
Neighbor         Interface         Uptime/Expires    Ver  DR
```

```
Address                                    Priority
192.168.212.1    Tunnel0          00:01:15/00:01:30 v2    1      (BD)

 ... End PIM neighbor Configuration.
```

Broadcast Redundancy at the Receiver

This section discusses the deployment goals, deployed technologies (products), and network topology involved in broadcast redundancy at the receiving end of live Webcasts over large-scale networks using RealServer 8.

Deployment Goals

- Serve connecting clients from disparate edge locations while acquiring live signals at a managed broadcast operation center.
- Ensure that live broadcasts arrive at the edge RealServers by forwarding redundant versions of the same broadcast across the network.
- Remove all single points of failure along the broadcast chain by establishing redundant encoding sessions, redundant transmissions, and redundant live-source aliasing at the edge RealServers.

Deployed Technologies

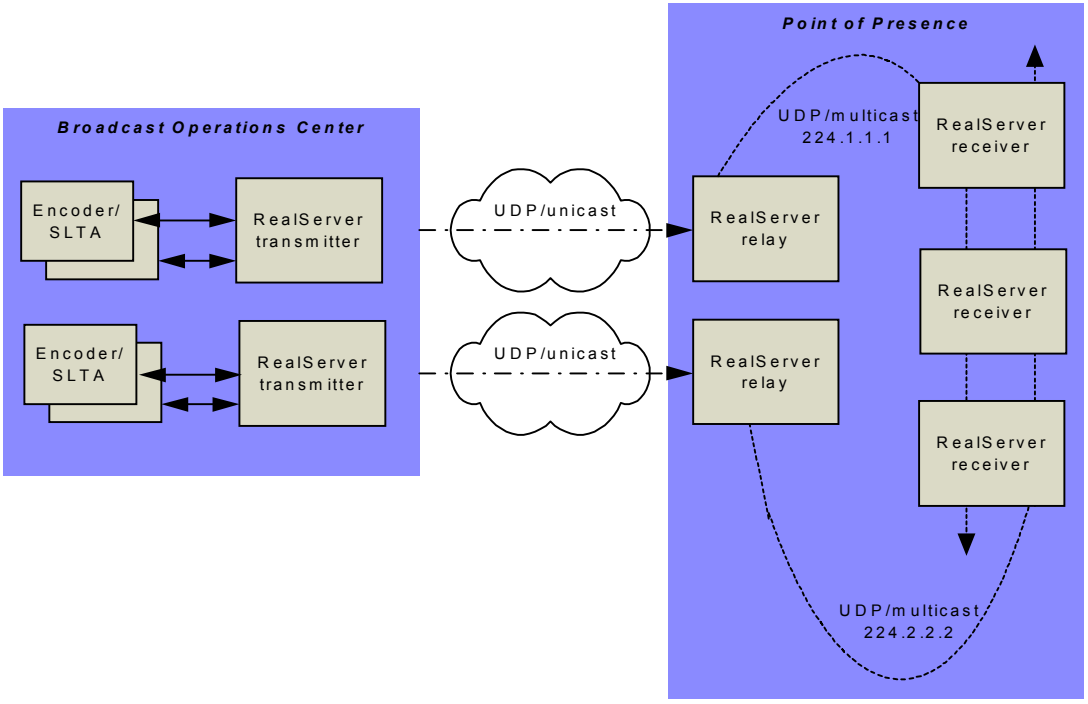
- RealProducer 8
- RealServer 8
- Network routers (layer 3) configured to establish GRE tunnels for non-multicast-enabled network segments

Network Topology

- RealServers broadcasting to clients are located at remote locations.
- Connectivity to these RealServers traverses the Internet over a combination of satellite and terrestrial networks.
- RealServers receivers are deployed at multicast-enabled locations far from the RealServer transmitters.
- RealServer operators may or may not have access to network routers. Thus, you use a RealServer configured as a relay at the remote point of

presence (POP) to rebroadcast inbound UDP unicasts as multicasts to local RealServer receivers that accept client connections.

Note that because broadcast distribution spans the Internet to reach remote edge RealServers, you should use RealServers configured to relay live broadcasts, instead of using routers, to traverse non-multicast-enabled network segments.



RealServer 8 Live Broadcast Configurations

This section presents a number of feature configurations for distributing live broadcasts with RealServer 8 by using *relays* instead of routers (there are, of course, many other configuration possibilities). Included in the tables are redundancy and splitting scenarios for transmitter, relay, and receiver hosts.

Live Broadcast Transmitter

This table describes some of the key RealServer feature configurations for transmitter hosts.

Broadcast redundancy	Configuring this is optional, but doing so ensures that live signal acquisition will continue if a failure occurs on the encoder host system or on the network to which both the encoder and the RealServer are connected.
Splitting: transmitters	Two receivers are configured to push two parallel distributions of the same live content to two unique RealServer relays on the far side of the Internet segment. The predefined transport protocol used for both of the distributions is UDP/unicast. To enable redundancy at the edge RealServer, the source name variable must be identical on both of the transmitters. Because the live data will be traversing the Internet, it is recommended that FEC be set as high as possible, depending on the bandwidth constraints in each case. The Honor Resend Request option should also be enabled. This provides an additional means of packet loss recovery for RealServer relays that are able to send unicast traffic back to the transmitter. The security type and password should be configured according to the policies and procedures of the broadcaster.

Live Broadcast Relay

This table describes some of the key RealServer feature configurations for relay hosts.

Splitting: receivers	The relay host is configured to receive the unicast from transmitter. By default, the receiver uses a buffer size of 6 seconds. You should increase this to give resend reports more time to initiate packet resends over Internet segments that likely have a high degree of latency.
Splitting: transmitters	The Relay Live Broadcasts option should be enabled on both relay hosts. Each relay host is configured to transmit the incoming source through the UDP/multicast transport protocol to a supported class D address inside the POP. The multicast time-to-live (TTL) needs to be set to at least one "hop count" greater than the number of hops required to reach all of the receivers residing inside the POP. Because the live data will be traveling only across the local network of the POP, you can use a low FEC setting. As a backup mechanism to the FEC fortification, make sure that the Honor Resend Request option is enabled, as this provides an additional means of packet loss recovery. The security type and password should be synchronized to work with the receivers residing inside the POP facility.

Live Broadcast Receiver

This table describes some of the key RealServer feature configurations for receiver hosts.

Broadcast redundancy	Enable broadcast redundancy on the receivers. Because both incoming live transmissions have the same file, path, and source name, they will be treated as redundant versions. Thus, if packet flow ceases for one transmission, the second one will take over and fulfill the receivers' need for data. As a result, live playback to clients will continue uninterrupted, and there will be no need for RealPlayers to be reconnected to the first transmitter.
Splitting: receivers	Parallel transmissions are configured to be received by all RealServers in the POP. The predefined transport protocol used for the broadcast is UDP/multicast. The system administrator should synchronize the port ranges with the ports used to forward traffic from the RealServer relay to the receivers. Because the receiver's network allows for resend requests to be sent back to the transmitter, the Resend Request option should be enabled. The security type and password must be configured according to the settings established on the RealServer relay. By default, the receiver uses a buffer size of 6 seconds.

Additional Resources

- *RealServer 8 Administration Guide*

  This guide explains how to run the standard RealServer. You will find this book at **http://service.real.com/help/library/guides/server8/realsrvr.htm**.

- RealServer 8 ReadMe

  The Readme file flags late-breaking changes and other news related to RealServer. It also outlines general setup recommendations.You can view the latest version of this file at **http://service.real.com/help/library/guides/server8/readme.html**.