TANDBERG

Infrastructure Deployment Guide

TANDBERG

D50362, Rev 2.0

Table of Contents

1. Abstract

This document is meant as a technical guide for developing and deploying an H.323 network using the TANDBERG infrastructure, including TANDBERG Gatekeepers and Border Controllers. It is not meant to serve as a complete comprehensive guide to network deployments, but rather as a guideline only. If more comprehensive information is required, please review section 5 and consult your TANDBERG representative.

This version of the document is written towards the specifications in the software release N5 for the TANDBERG Gatekeeper and Q5 for the TANDBERG Border Controller.

2. Deploying an H.323 Network

The N4/Q3 and previous software releases for the TANDBERG Gatekeeper and Border Controller include many different features that will allow a network engineer to design and implement a network that will meet all of their specific needs. From authentication to network modeling to centralized traversal deployment, these releases will allow the video deployment to adapt itself to the network rather than forcing a modification to the network to incorporate video.

The goal of a video deployment is to incorporate a new tool that will aid in conducting business but will not hinder any other business from being pursued. In order to accomplish this task, different aspects of the deployment will need to be considered, including security, bandwidth management and redundancy. Each one of these aspects will play its own role in ensuring that the video network is both successful and does not interfere with any other activities within the organization. The following sections will describe each one of these aspects, why they might be needed and how to utilize them in the proper form.

2.1 Design Considerations

It is important to understand all of the network requirements prior to implementing any of the infrastructure devices on the network directly as the specific requirements may change how the network devices themselves are going to be modified. Once the end goal is understood, at least on a high level, the basic network design can then begin.

2.2 Deploying a Gatekeeper

A gatekeeper should be considered a requirement within all H.323 network installations as it provides a significant impact to the success and ease of use to the network itself.

The primary purpose of a gatekeeper is address translation, translating IP addresses of endpoints into user-friendly E.164 aliases and H.323 IDs. These aliases allow for an endpoint to be called through the use of names and numbers rather than tough to remember IP addresses. While dialing by IP address is still a practiced function in the

H.323 world under the IPv4 protocol, it will not be possible for an end user to dial the IP address of an IPv6 system as they are much more complicated and tough to remember. Aliasing is a common practice in IP communications – for example, when one references an email address of a remote person, they reference it using the DNS name of the corporation they are communication with (e.g. john.doe@company.com); they do not reference it via the IP address of the SMTP mail server of the organization (e.g. john.doe@1.2.3.4). In the mail example, a DNS server will serve as the go-between of the two organizations; in the H.323 world, the gatekeeper provides that same functionality.

In a DHCP environment, translation of IP addresses is an absolute requirement as addresses can change at any time. In order to then ensure continued success of videoconferencing in this type of an environment, a gatekeeper is required to perform the automatic translation of the endpoint addresses.

If an MCU or gateway is installed on the network, the deployment of a gatekeeper is required to take advantage of all resources within these systems. For example, to dial from an H.323 system out to an H.320 system through a gateway, the H.323 system will dial a gateway prefix + the ISDN number of the H.320 system. The gatekeeper will then resolve the prefix dialed by the H.323 system to the gateway and route the call accordingly.

In addition to the translation of addressing, the gatekeeper will serve as a centralized management point of an H.323 network. Providing benefits such as aiding in firewall traversal, bandwidth limitations, authentication and authorization, the gatekeeper will provide management of all endpoints within an H.323 network from a centralized source, restricting the ability for endpoints to cause issues within the IP infrastructure and ensuring the success of the H.323 deployment. These features of the gatekeeper will be discussed in depth later in this document.

2.3 Deploying a Border Controller

A TANDBERG Border Controller serves as the traversal server in the TANDBERG Expressway™ firewall traversal solution. In order to ensure that any and all firewalls do not interfere with the progress of the video calls connected through the Border Controller, it is highly recommended that the box is placed completely outside the firewall. By placing the traversal server outside the firewall, all communication will utilize the Expressway technology in order to pass through the firewall and will then flow from the Border Controller out to the internet. However, if the box is placed on the DMZ of the firewall, the traffic will then flow through the firewall again after it leaves the Border Controller towards the public internet, thereby giving the firewall an opportunity to interrupt the call flow and cause issues with connectivity. The box itself is hardened and designed to be placed outside the firewall.

No matter where the system is placed on the firewall configuration, it is *required* that the Border Controller is directly configured with a publicly routable IP address. Due to the nature of address signaling within an H.323 connection, the Border Controller will signal contact information to both endpoints involved in the traversal call, instructing them what address and ports to connect in order to complete the call. Thus,

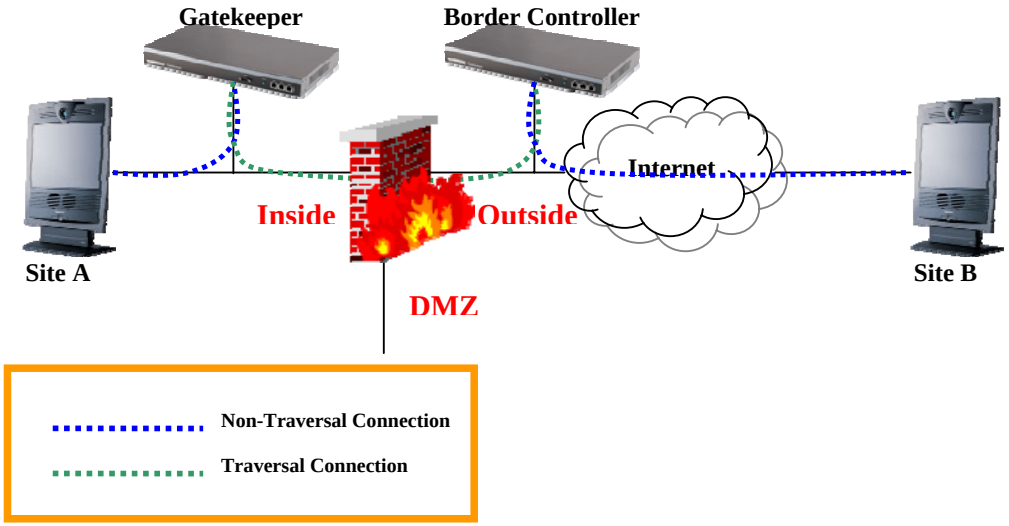
the Border Controller is required to be directly configured with a public address such that this signaling can take place and the success of a traversal call can be guaranteed.

The Border Controller can also provide gatekeeper functionality to all of the endpoints registered directly, thereby ensuring that any endpoint registered can serve as a fully functioning entity of the H.323 network.

2.3.1 Deploying a Border Controller Outside the Firewall

As stated previously, the Border Controller is designed to be placed completely outside the firewall. In this scenario, all traffic from within the network will traverse through the firewall on the reserved Expressway ports, and then will progress to the public internet unimpeded.

For the initial design, we will consider deploying a TANDBERG Gatekeeper in conjunction with the Border Controller to serve as the traversal client. In subsequent sections of this document, we will discuss different options regarding the gatekeeper, including deploying traversal without a gatekeeper.

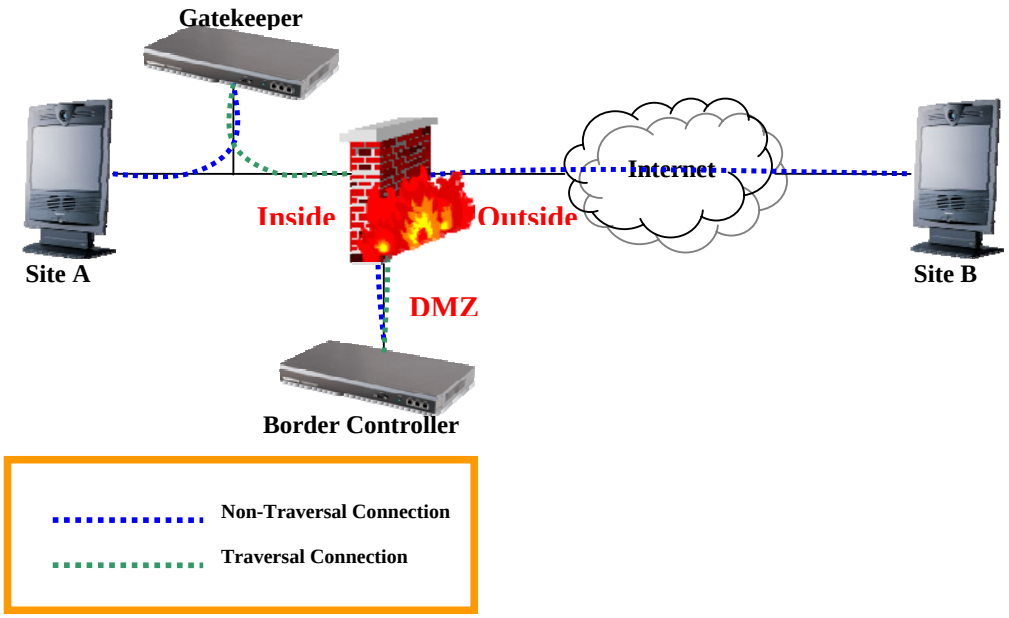


In the scenario illustrated above, Site B is an unregistered endpoint on the public Internet with no firewall traversal issues at all. Because of its position within the network diagram, traversal technology will not be used to connect to this endpoint, thereby using the normal H.323 connection procedures and layer 4 ports of both the endpoint and the Border Controller. Because Site B is now an unknown endpoint and all of the layer 4 ports will vary depending on the manufacturer, the firewall rules would be complex in order to allow the traffic to flow freely in both directions – a requirement for a successful call.

2.3.2 Deploying a Border Controller in the DMZ (Single Firewall)

If concerns about placing the Border Controller completely outside the firewall exist, the system can be placed within a controlled domain portion of the firewall, commonly known as the DMZ or Demilitarized Zone. This portion of the firewall exists on a

third interface and is neither inside nor completely outside the firewall, but rather a public interface that allows an administrator to restrict access to specific resources on that interface. For example, hosts on the DMZ portion of the firewall may be readily accessible on well known videoconferencing ports, but all management ports such as HTTP, HTTPS, telnet and others are access restricted from the public internet.



When a Border Controller is installed on the DMZ area of the firewall, traffic will now flow through the firewall twice during a call in the above scenario. Just as previous, the traffic on the traversal link from the gatekeeper to the Border Controller will flow through the firewall and be subject to the firewall rules and awareness. However, once the traffic then flows from the Border Controller out to Site B over the non-traversal connection, the call will then be subjected to the firewall rules and awareness a second time, therefore introducing another variable into the call connectivity.

However, there are benefits to placing the Border Controller in the DMZ section of the firewall. By placing the system into a portion of the network that is controlled by the administrator, specific restrictions can be inserted in order to prevent unauthorized access to the system. In this case, the Border Controller will be placed on this controlled domain and the rules on the firewall will require manipulation in order to allow for H.323 traffic to flow unimpeded to the far end.

In order to ensure a successful deployment in this scenario, it is absolutely *required* that the Border Controller is configured with a publicly routable IP address; NAT or Network Address Translation is not supported in the Border Controller. Additionally, it is highly recommended that all H.323 awareness on both the RAS and the H.225 channels is disabled and all traffic from ports 1024 to 65535 both TCP and UDP are allowed for inbound and outbound communication from the public Internet to the specific IP address of the Border Controller located on the DMZ. These requirements are structured to mimic the placement of the Border Controller on the outside of the firewall, while maintaining access restrictions on all of the management ports of the

box, thereby providing both a secure and flexible solution to prevent any problems from occurring.

If further lockdown of the ports for the Border Controller is required, access to the public Internet can be restricted to the TCP and UDP ports required by the Border Controller only, however, this will not guarantee the fully flexible solution, regardless of the type of endpoint located at Site B.

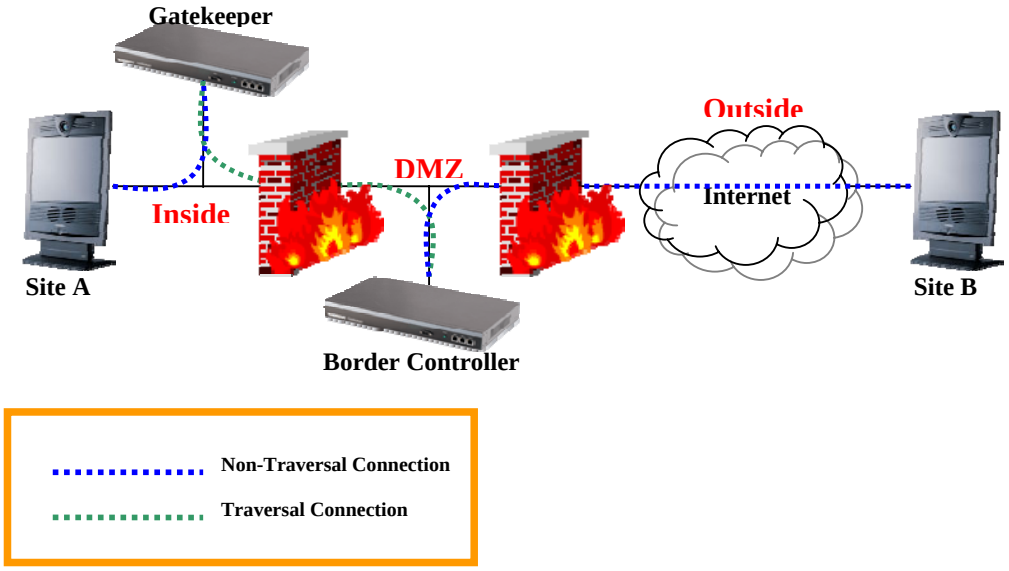
Parameter	Configuration	Optional/Required
IP address	Public	Required
H.323 Awareness	Disabled	Required
TCP Ports 15000:15100	↔	Required (H.225)
TCP Ports 19000:19100	↔	Required (H.245)
UDP Ports 50000:51600	↔	Required (Media)
TCU/UDP Ports 1024:65535	↔	Recommended

↔ Bi-directional communication

All of the recommendations that are made above are done so in order to promote flexibility of the deployment. In order to ensure that the Border Controller is deployed in such a way that it will not limit connectivity of any calls to and from various endpoints throughout the network, all of the recommendations above should be followed.

2.3.3 Deploying a Border Controller in the DMZ (Two Firewalls)

Certain scenarios exist where the Border Controller is required to be installed behind a specific Internet-facing firewall to further decrease the exposure of the Border Controller into this type of environment. In this scenario, the common deployment looks similar to the figure below.

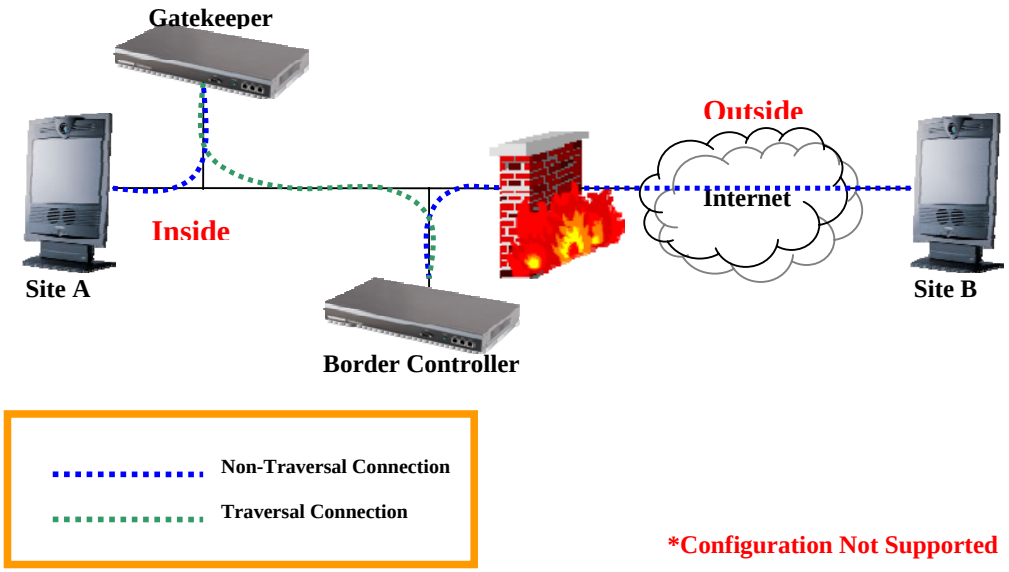


In this scenario, two specific firewalls are used in order to provide the DMZ functionality instead of a single firewall as previously discussed. In this type of a deployment, the same rules would apply to the second firewall as applied to the DMZ port of the firewall. One major change, however, will exist in the second firewall – this firewall must now allow bi-directional communication between the "inside"

interface (this is the DMZ interface of the network, in actuality) and the "outside" interface of the firewall. Because this is the pathway between the Border Controller and all external endpoints, connections will be sent unsolicited in from the outside of the network to the Border Controller located in this DMZ.

2.3.4 Deploying a Border Controller Behind a NAT Environment

Because the Border Controller is a traversal server to assist H.323 traffic through a firewall (going from public to private access), the system cannot be placed behind the firewall as it will not be able to function as normal. The intended purpose of the system is to be placed on a public-facing side of the firewall. The system is required to be configured with a publicly routable address which will be used to signal to the far end system in order to complete all traversal calls. If the system is configured with a private address, the Border Controller will then signal the private address to the far end, thereby preventing calls to connect.



If placing the Border Controller outside the firewall is not a possibility, implement one of the DMZ solutions discussed above.

2.4 Deploying Firewall Traversal

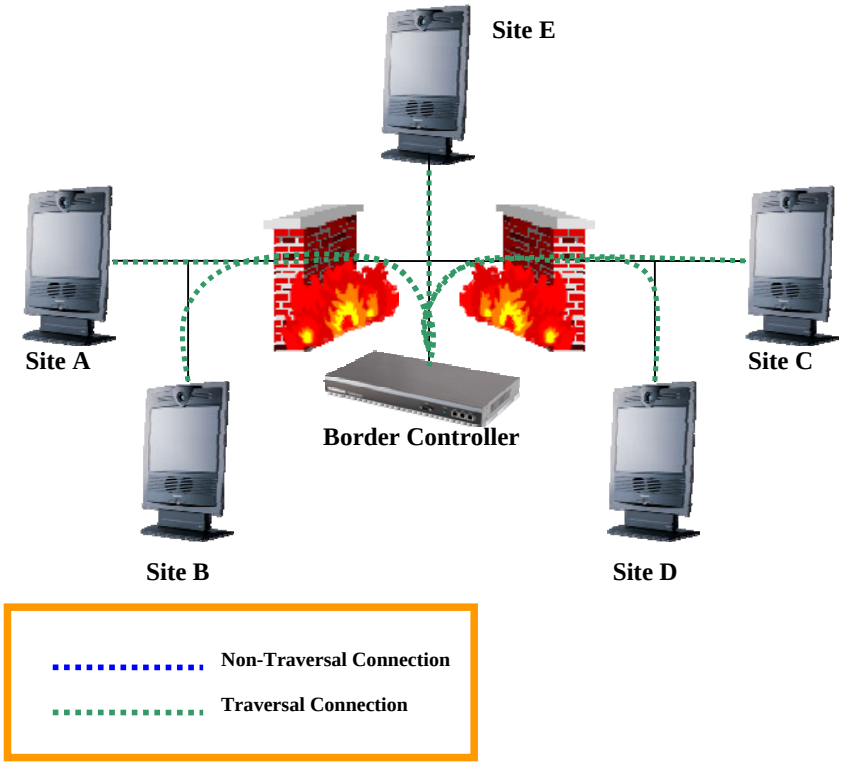
In order to be deployed properly, firewall traversal requires at least two of the three elements involved in the solution:

- Traversal server (e.g. TANDBERG Border Controller)
- Dedicated traversal client (e.g. TANDBERG Gatekeeper)
- Endpoint with traversal client built-in (e.g. TANDBERG MXP Endpoint or any endpoint that supports H.460.18/.19).

Any of these deployments will function without any issues on the connectivity side, but might not be the best deployable solution. A solution must meet the requirements of the end user as far as usability and scalability are concerned. Many different deployments will be similar as far as configuration is concerned, but will differ on the details of the deployment. Overall, they will follow one of the following deployments.

2.4.1 Single Traversal Server

The traversal server can be deployed solo as a "minimalistic" approach to the network design. In this design, depicted in the diagram below, all endpoints will register directly to the Border Controller as a central point of the network, thereby removing firewall barriers between any and all endpoints.



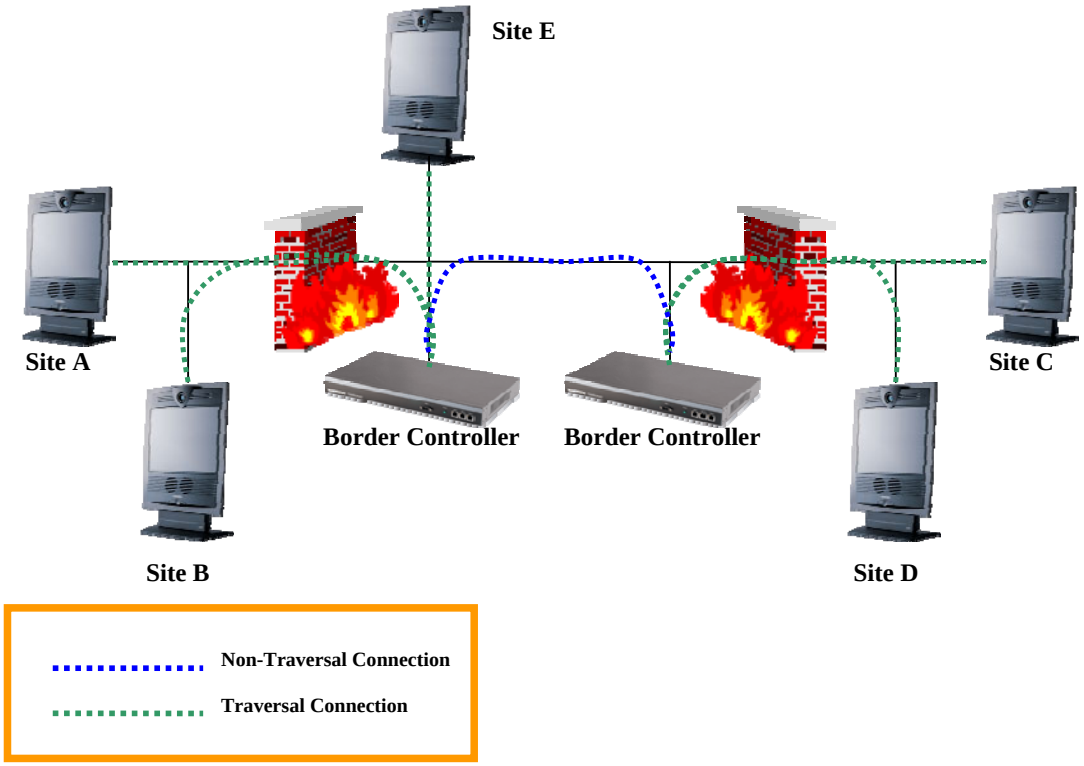
Within this deployment, however, all calls between all endpoints count as traversal calls on the Border Controller licensing. For example, if Site A calls Site C, the Border Controller will license that as a traversal call. If Site B calls Site A, the Border Controller will also license that as a traversal call. For a larger deployment of endpoints, licensing may present issues if this deployment is considered.

Additionally, all of the bandwidth of all of the calls will flow through the Border Controller. If Site B calls Site A, the call bandwidth (both call control and media) will flow from Site B out of the firewall to the Border Controller, then traverse back inbound to Site A. If the Border Controller is then installed at an off-site location, this connection will then tie up a significant amount of bandwidth on the local internet connection as the network will see the connection between Site B and the Border Controller as well as Site A and the Border Controller, thus creating virtually 2 calls from the network point of view. For example, if Site B called Site A at 384 kbps, the network would see two active 384 kbps streams between the endpoints and the Border Controller, realistically occupying 768 kbps of bandwidth.

Finally, this approach will not be able to support any endpoints that do not support a type of traversal technology, such as H.460.18/.19.

2.4.2 Multiple Traversal Servers

In order to ease the restrictions on both the bandwidth and the options of the above design, multiple traversal servers can be deployed throughout a network, each serving the local endpoints.



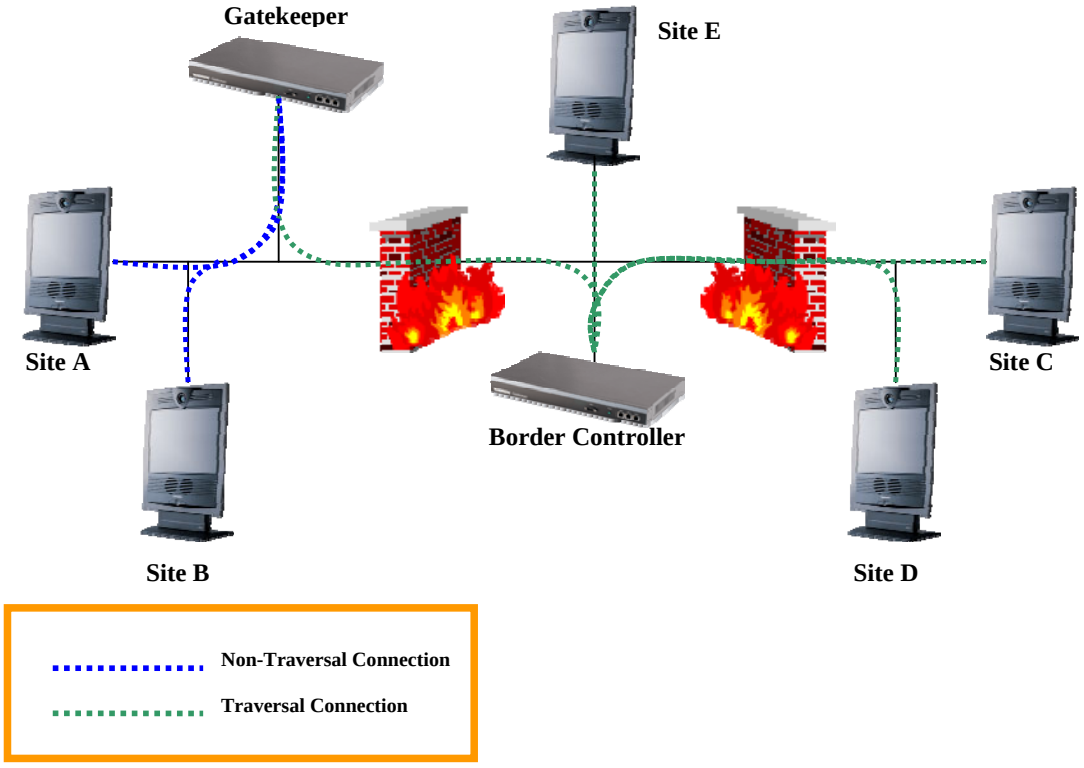
This network design continues to serve the "minimalist" approach to the layout as it will implement few boxes to achieve large scalability. In this particular situation, there will be less of a strain on options and bandwidth as the Border Controllers are deployed locally (reducing the need for calls between local endpoints to truly leave the network) and deploying more systems to share the option load.

In this case, if Site B calls Site A, the call will still traverse through the firewall from Site B and return through the firewall to Site A, but will not tie up any WAN bandwidth due to the local installation of the Border Controller. However, the traversal options of the local Border Controller will still be used and the traffic will be required to flow through the firewall for both connections, increasing the work load of the firewall for a local connection.

Additionally, this network layout will not support the ability for systems that do not support any type of traversal technology, such as H.460.18/.19.

2.4.3 Single Traversal Server, Single Internal Gatekeeper

In order to promote flexibility of the network design, an administrator could deploy a network with a single traversal pair of systems within a major office and have remote endpoints then register directly into the major office.



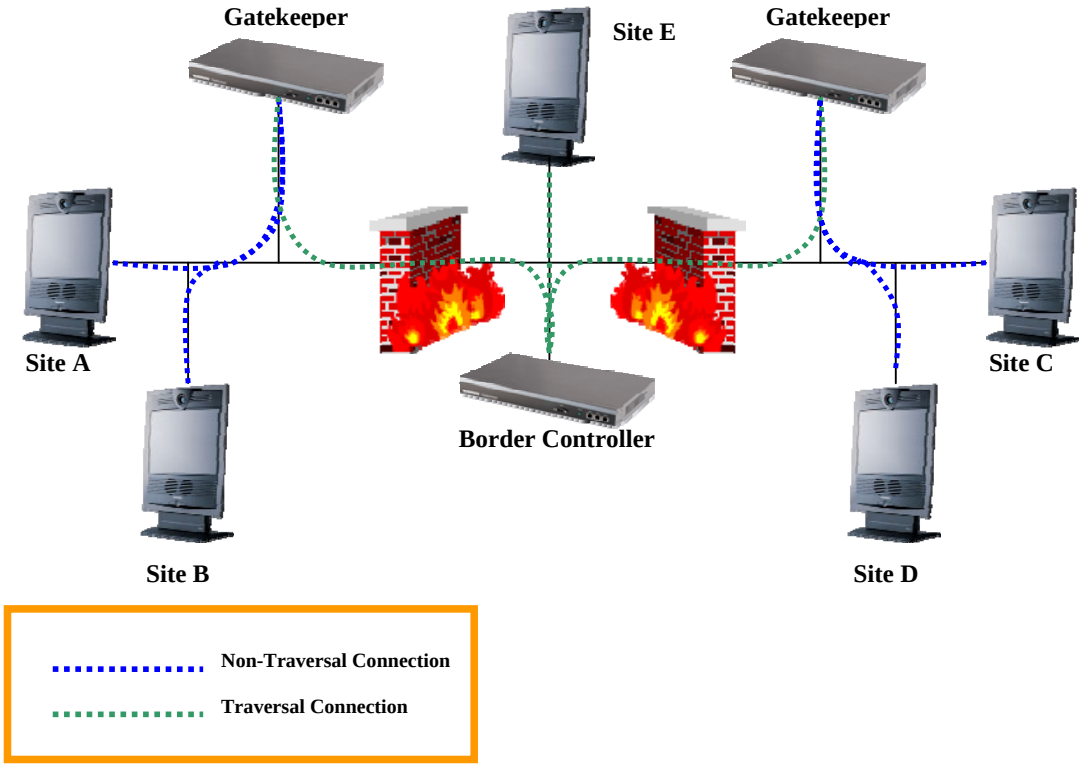
This design continues to minimize the amount of equipment installed on the network while providing some extra functionality to the systems installed locally to the gatekeeper. These systems, when registered directly to the local gatekeeper will not utilize any traversal options, WAN bandwidth or firewall resources when connecting to other systems on within the same gatekeeper zone. This solution works on the premise that most H.323 systems are local to a single office and a majority of the H.323 call traffic is local as all other systems located at remote offices or in a small office, home office (SOHO) environment will be required to register directly to the local Border Controller. This solution will also allow non-traversal capable endpoints on the local network and registered to the gatekeeper to take advantage of the traversal technology as the gatekeeper will be able to serve as the traversal client for these systems.

This solution, while improving the versatility of the network deployment, will continue to suffer from some of the challenges of previous designs as all endpoints located outside the network will tie up traversal resources for all calls, even those that will take place between endpoints located on the same LAN in remote offices (e.g. between Sites C and D). There will also be a large utilization of bandwidth of the centralized office when remote endpoints are connected to endpoints that are not located on that centralized LAN; remote offices that implement multiple video systems will also require a large amount of bandwidth when connecting calls between endpoints within that office.

This solution will also not support endpoints that do not support traversal technology and are not located within the centralized office.

2.4.4 Single Traversal Server, Multiple Internal Gatekeepers

To continue to improve the versatility of the H.323 network, multiple centralized traversal clients (e.g. TANDBERG gatekeepers) can be configured to a single traversal server. In this scenario, each office with multiple endpoints can deploy a gatekeeper that will then register to a centralized traversal server.



When a gatekeeper is deployed locally at the remote offices, the traversal resources and WAN bandwidth will only be utilized when connecting to systems that are not located on the same LAN. For example, when Site C calls Site D, the traffic and resources will remain within the LAN and will allow the firewall resources and bandwidth to be better utilized by other applications. This network design will also provide an additional point of management within the network administration, thereby allowing an administrator to monitor and restrict access to network resources as required (these type of restrictions will be discussed later in this document).

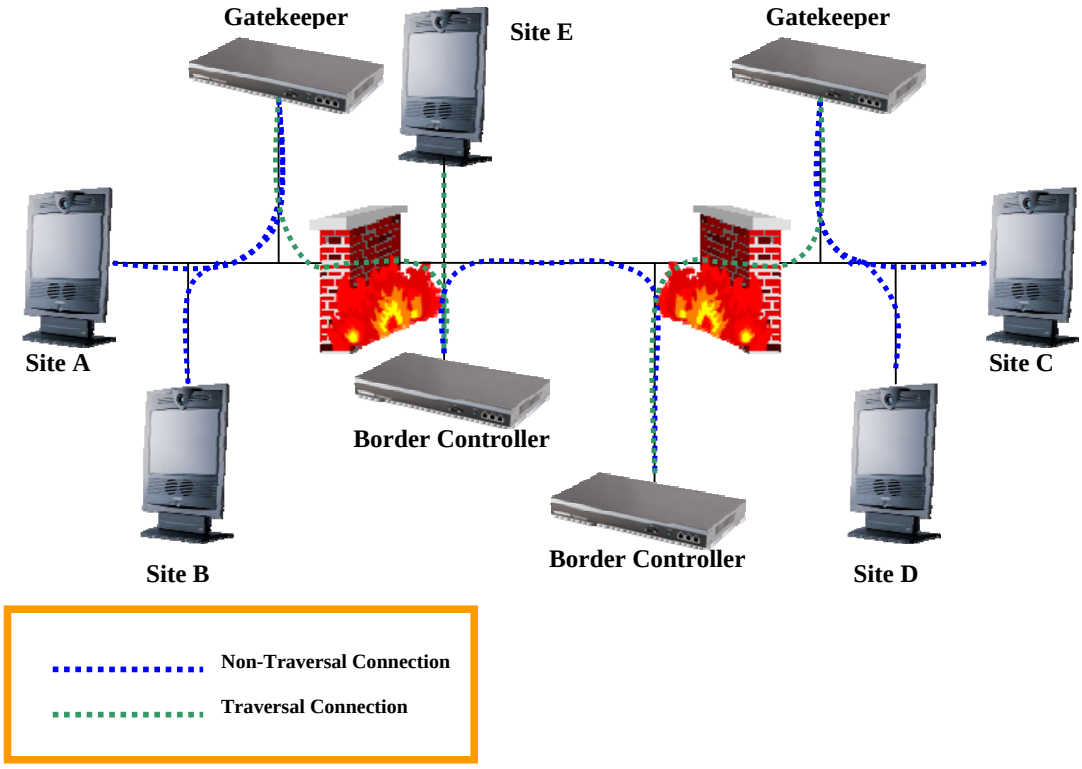
Any single or small concentration of endpoints located in a SOHO-type of environment will then register into the solution through the centralized Border Controller in order to

This design, however, does have a few restrictions that need to be considered. First of all, the deployment of a centralized Border Controller will continue to have a large bandwidth requirement when any site at a remote location is connecting to another site at different remote location as both segments of the call will flow through the centralized Border Controller.

This solution will also not support the ability for systems that do not support any type of traversal technology to register in unless they are located in an office that has a localized gatekeeper to act as the traversal client on behalf of the endpoint.

2.4.5 Multiple Traversal Server, Multiple Internal Gatekeepers

The most flexible network design will implement multiple traversal servers throughout the network in order to de-centralize the bandwidth and option requirements.

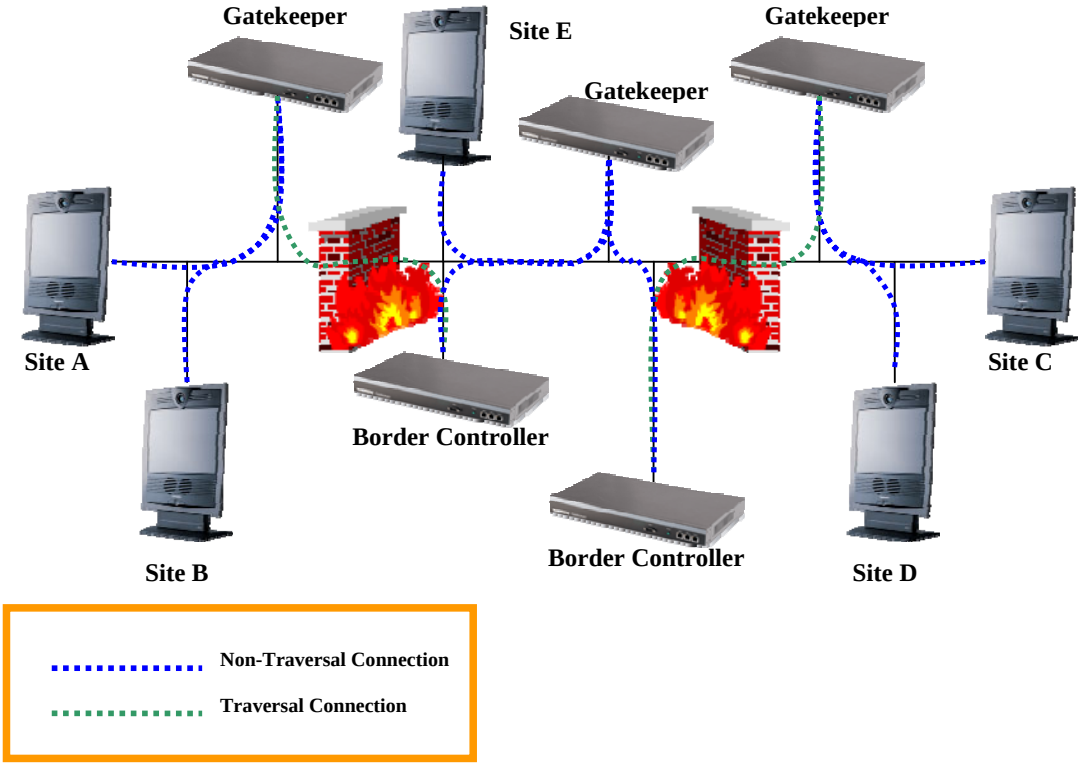


This deployment will offer the most flexibility and should be used if centralizing network resources is not an option. A distributed model will allow for systems to be deployed and give the least amount of bandwidth requirements to a single site while continuing to provide full traversal capability to all endpoints involved. Each gatekeeper-Border Controller pair located at a remote office will serve as a self-standing network allowing endpoints within that network to connect to all others, no matter where they are located on the network.

This solution will not, however, solve any type of a traversal requirement for endpoints that do not have a localized gatekeeper that will serve as the traversal client on their behalf.

2.4.6 Centralized Public Gatekeeper

If endpoints located in a SOHO environment require the ability to register into the network but do not support any type of traversal technology and do not warrant a localized gatekeeper to be installed, a centralized gatekeeper can be installed on the public side of the network to allow for any endpoints in this situation to register.



The external gatekeeper will allow endpoints not able to take advantage of the traversal technology directly to register into and participate within the H.323 network. This solution, however, will not be able to solve any firewall challenges for the remote endpoints that are registered directly.

Deployment of an external gatekeeper can be combined with any of the network layouts described above to achieve this functionality.

2.4.7 Customized Deployment

All of the deployment scenarios discussed above are not full deployment solutions, but express fundamentals about how the deployments are implemented and any limitations of which one would need to be aware when making the decision on deployment. Many networks that implement firewall traversal do not fit in any one of the above designs, but rather a hybrid of two or more to serve the purpose of the network.

3. H.323 Network Security

When deploying an H.323 network, especially one that will provide the ability to connect with systems not locally controlled, the ability to maintain security of the network may become a focus on specific deployments. The TANDBERG solution will allow an administrator to maintain and perhaps strengthen the security of the H.323 network through the implementation of some the features that are described in the following sections.

3.1 Restricting Registrations

Registrations within the H.323 network can be restricted at the gatekeeper or Border Controller level through the implementation of one or both of the following methods. Without an active registration to the network, the system is thereby restricted from accessing network resources.

3.1.1 Allow/Deny Lists

Endpoint registrations to a gatekeeper or Border Controller on a network can be restricted by H.323 ID or E.164 alias of the registering endpoint. When configured for an allow/deny list, the gatekeeper or Border Controller will monitor the list prior to confirming the registration to ensure the registration is permitted within the rules.

3.1.1.1 Allow List

When configured for an allow list, the gatekeeper or Border Controller will only allow E.164 aliases or H.323 IDs on the list to register. If an endpoint tries to register to the gatekeeper and the alias or ID is not present on the list, the registration will be rejected with the reject reason of '*securityDenial*.' If, upon registration, the E.164 alias or H.323 ID of the system is present on the list, the gatekeeper will allow the registration to continue.

3.1.1.2 Deny List

When configured for a deny list, the gatekeeper or Border Controller will permit all endpoints that attempt to register unless the endpoint is on the list. The deny list should be setup to prevent specific aliases to register to the gatekeeper as any endpoint not explicitly on the list will receive full access to the network resources.

3.1.2 Authentication

Authentication can also restrict access to an H.323 network through the use of a username/password credential combination in order for the gatekeeper/Border Controller on the network to verify the identity of the endpoint registering. Once authenticated into the network, the endpoint will then supply the same authentication credentials to the gatekeeper for all RAS messaging signaling, including all call requests and responses, thereby ensuring that all messaging is coming from a trusted source.

The password information within an authenticated RAS message is not sent in clear text, but rather in a hashed mode along with other information in order to increase the security to ensure that no third party can obtain the password through any type of sniffing or decoding means. The hash serves as a challenge response to the gatekeeper to show that the endpoint does, in fact, know the password.

Another requirement for the H.235 authentication will be communication with either a public or private NTP server. This time synchronization, also used within the hash mechanism, is to ensure time relevancy to both the gatekeeper and the endpoints, thereby ensuring that all communication was sent promptly to the gatekeeper from the endpoints. By using these timestamps within the messaging, the gatekeeper prevents "replay attacks" by ensuring that all communications are occurring real-time.

How does the signaling work? Before attempting to register to the gatekeeper, the endpoints (Gateway included) now send a Gatekeeper Request (GRQ) message directly to the gatekeeper it has been programmed to register to. Within the GRQ message, the endpoint identifies that it supports authentication. When the gatekeeper then responds with the Gatekeeper Confirm (GCF) message, the system then responds to the endpoint to identify that it also supports authentication. From this moment on, all RAS messaging from the endpoints to the gatekeepers will include the authentication credentials.

If an endpoint does not support authentication and tries to register without the authentication credentials, the gatekeeper will simply reject the registration as unauthorized. This same rejection will be sent to the endpoint upon transmission of invalid credentials.

When the gatekeeper or Border Controller sends out Location Request (LRQ) messages to other neighbored gatekeepers, it will now send its own credential information (e.g. username and hashed password challenge) to the far end gatekeeper, provided that the gatekeeper can reach its own authentication record. If the gatekeeper is configured for Local Database Authentication, one of the local database entries must be the gatekeepers system name (acts as the username) and password. If the authentication method chosen is to reference an LDAP server, the LDAP server must have an entry for the gatekeeper system name and password. Upon LRQ processing, the gatekeeper will lookup its own credentials and forward them onto the remote gatekeeper, if found.

Upon receipt of an LRQ from another source, the first task the gatekeeper or Border Controller will complete is to verify the username and password, if configured for authentication; if authentication is turned off, however, the gatekeeper will consider the incoming LRQ as it did in previous software versions. If the username and password sent within the incoming LRQ match the credentials stored in the receiving gatekeeper, the LRQ will be considered from a trusted source and will be passed as so to the CPL engine (reference section 3.2.3 of this document for more information). If the incoming LRQ credentials do not match what the gatekeeper has stored as the credentials or there are no credentials present within the incoming LRQ, the incoming information present within the LRQ will be passed to the CPL engine as from an un-trusted source and handled as such.

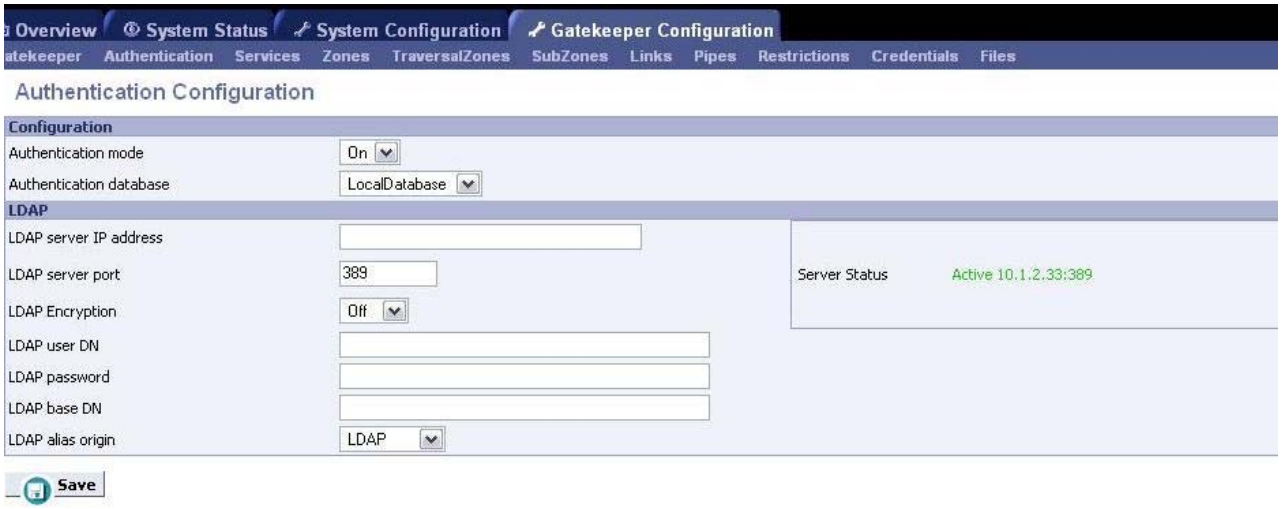
For storage of the authentication credentials, the gatekeeper and Border Controller support two different methods, Local Database Authentication and Lightweight Directory Access Protocol (LDAP) Authentication.

3.1.2.1 Local Database Authentication

The gatekeeper supports an internal database for storing the Authentication credentials. The internal database can store up to 1000 entries and will store only the username and password. This method of authentication will not verify the E.164 alias and H.323 ID of the endpoint directly (can be combined with the Allow/Deny list method discussed above); as such, local database authentication is intended for small

scale deployments and can provide a good starting point for the implementation of authentication within the network.

See below for a screenshot of configuring the gatekeeper for Local Database Authentication.



3.1.2.2 LDAP Authentication

In addition to the Local Database, the gatekeeper will also support authentication with an LDAP database that is stored elsewhere on the network (for details on which LDAP servers are supported and how the communication works between the gatekeeper/Border Controller and the LDAP server, please see the TANDBERG Gatekeeper User Manual or the TANDBERG Border Controller User Manual).

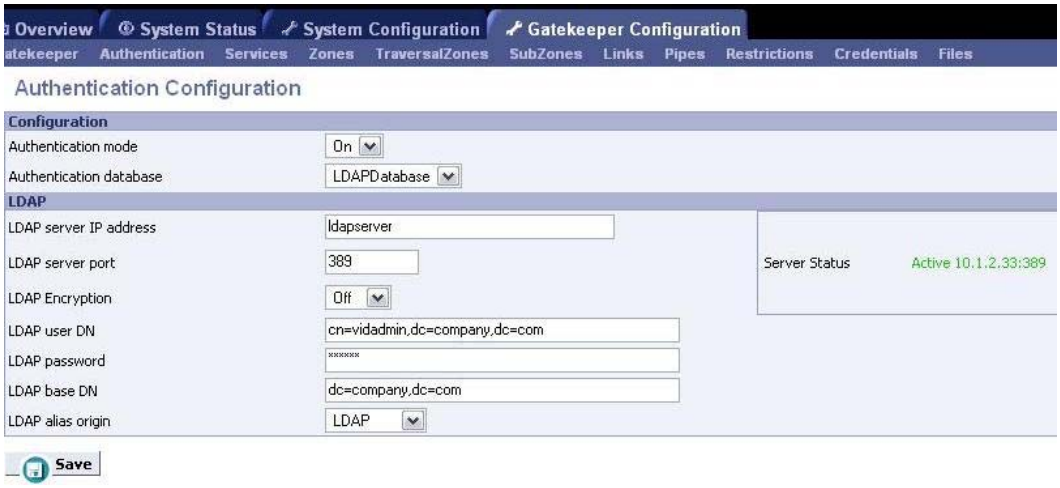
When LDAP Authentication is enabled within the gatekeeper or Border Controller, upon receipt of any RAS traffic, the gatekeeper will validate the credentials within that RAS traffic with the LDAP server itself. Once the credentials are confirmed, the verification goes one step further – the gatekeeper/Border Controller will also validate the endpoint's E.164 alias as well as the H.323 ID with what is stored within the account on the LDAP server[1] (*Note:* It is required to store the E.164 alias within the LDAP user account, however the H.323 ID is optional - if the E.164 alias is not stored in the account, the RAS message will be rejected). This is done in order to further validate the endpoint in the network and ensure that the endpoint is what it says it is, beyond the username and password. This increased validation becomes very important when authorizing the system for call permissions and increases the security and administrative control of the video network.

The LDAP Authentication is designed for larger deployments than the local database. Because of the validation of E.164 aliases and H.323 IDs, this authentication method

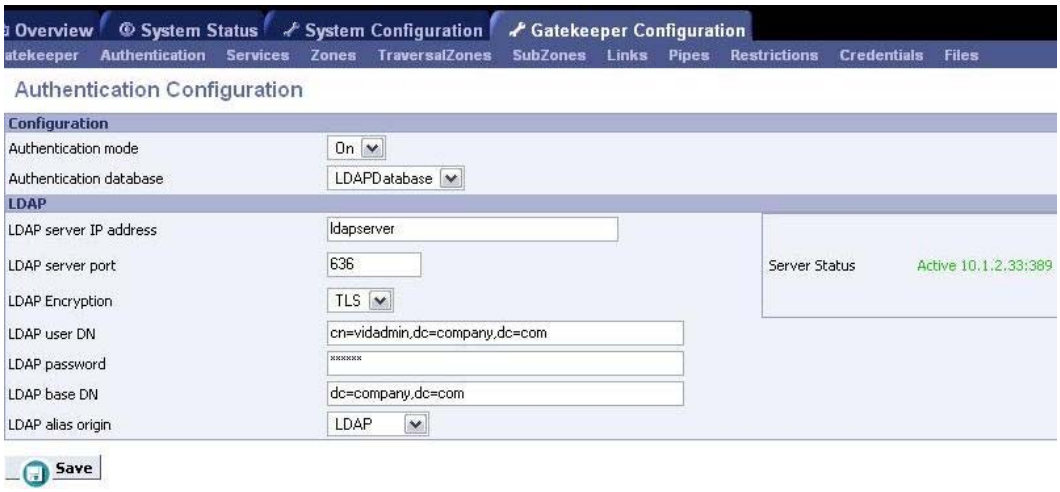
[1] It is required to store the E.164 alias within the LDAP user account, however the H.323 ID is optional. If the E.164 alias is not stored, the RAS request will be denied for a security violation. If the H.323 ID is not present, it will not be verified. If this ID is present, it will be verified and, if it does not match, the Gatekeeper will reject the RAS request for a security violation as well.

may also be desired in deployments that want to ensure complete validation of the endpoint registered to the network.

See below for a screenshot of the gatekeeper/Border Controller configured for unsecured LDAP Authentication. For information on configuring secured LDAP Authentication using either TLS or LDAPS, please see the TANDBERG Gatekeeper User Manual or TANDBERG Border Controller User Manual.



LDAP Authentication – Non-encrypted



LDAP Authentication - Encrypted

3.1.2.3 Authentication Signaling and Details

When an endpoint registers to a gatekeeper, it will begin the registration process by sending a Gatekeeper Request (GRQ) to the gatekeeper stating it supports authentication. Upon receipt of the GRQ, the gatekeeper will then confirm back (GCF) stating that it also supports authentication, if in fact authentication is enabled within the gatekeeper. If the gatekeeper responds with a GCF and the authentication support is missing in the message, the endpoint is then aware that it does not need to send authentication credentials to the gatekeeper for registration.

Once an endpoint registers with authentication enabled, all future RAS traffic must also include the authentication credentials. If a message is received by the gatekeeper that does not contain these credentials, it will then be rejected with the reasoning of *security denial*.

- Gatekeeper Request from endpoint

In the GRQ below, the endpoint will signal that it supports authentication (offset by bold type). If the gatekeeper responds that it does as well, the endpoint will submit the authentication credentials within the RRQ.

Note: This text was taken from a syslog directly off of the gatekeeper.

```
value RasMessage ::= gatekeeperRequest :
{
 requestSeqNum 2496,
 protocolIdentifier { 0 0 8 2250 0 5 },
 rasAddress ipAddress :
 {
  ip '0A0106D2'H,
  port 1719
 },
 endpointType
 {
  vendor
  {
   vendor
   {
    t35CountryCode 130,
    t35Extension 1,
    manufacturerCode 256
   },
   productId '54616E6462657267'H,
   versionId '3435'H
  },
  terminal
  {
   nonStandardData
   {
    nonStandardIdentifier h221NonStandard :
    {
     t35CountryCode 130,
     t35Extension 1,
     manufacturerCode 256
    },
    data '54616E6462657267'H
   }
  },
  mc TRUE,
  undefinedNode TRUE
 },
 endpointAlias
 {
  h323-ID : "System.Unit",
  dialedDigits : "7035551234"
 },
 authenticationCapability
```

```
 {
  pwdHash : NULL
 },
 algorithmOIDs
 {
  { 1 2 840 113549 2 5 }
 },
 supportsAltGK NULL,
 genericData
 {

  {
   id nonStandard : '20DF8903596F45199F2773C0A59274AF'H,
   parameters
   {

    {
     id nonStandard : '20DF8903596F45199F2773C0A59274AF'H,
     content raw :
'3C617373656E743E3C617373656E745F747970653E636C69656E743C2F617373656E74
...'H
    }
   }
  }
 }
}
238428604 [GKRAS-0] GRQ ASSENT info:
Assent Config :
 Version : 1
 Assent Type: Client
```

- Gatekeeper Confirm sent from gatekeeper

In this GCF, the gatekeeper will signal that it also support authentication and will expect all future RAS traffic to contain the authentication credentials (offset by bold type).

```
238428604 [GKRAS-0] [10.1.6.210:1719] Sent GCF

238428604 [GKRAS-0] ASN.1 PDU:

value RasMessage ::= gatekeeperConfirm :
{
 requestSeqNum 2496,
 protocolIdentifier { 0 0 8 2250 0 5 },
 rasAddress ipAddress :
 {
  ip '0A010225'H,
  port 1719
 },
 alternateGatekeeper
 {

  {
   rasAddress ipAddress :
   {
    ip '0A010226'H,
    port 1719
   },
   gatekeeperIdentifier "Alternate 1",
   needToRegister TRUE,
   priority 1
  }
 },
 authenticationMode pwdHash : NULL,
 algorithmOID { 1 2 840 113549 2 5 },
 genericData
 {

  {
   id nonStandard : '20DF8903596F45199F2773C0A59274AF'H,
   parameters
   {

    {
     id nonStandard : '20DF8903596F45199F2773C0A59274AF'H,
     content raw :
'3C617373656E743E3C617373656E745F747970653E7365727665723C2F617373656E74
...'H
    }
   }
  }
 }
}
```

Once an endpoint has confirmed that the gatekeeper supports authentication, it will then provide its username, password and timestamp within the Registration Request (RRQ). The password is not sent in clear text, but rather using a secure MD5 hash in order to provide secure transmission of the credentials to the gatekeeper.

- Registration Request sent from endpoint

When an endpoint registers to an authenticated gatekeeper, it will signal the authentication credentials within the RRQ (these credentials are offset by bold type).

```
value RasMessage ::= registrationRequest :
{
 requestSeqNum 2239,
 protocolIdentifier { 0 0 8 2250 0 5 },
 discoveryComplete TRUE,
 callSignalAddress
 {
  ipAddress :
  {
   ip '0A0106D2'H,
   port 1720
  }
 },
 rasAddress
 {
  ipAddress :
  {
   ip '0A0106D2'H,
   port 1719
  }
 },
 terminalType
 {
  vendor
  {
   vendor
   {
    t35CountryCode 130,
    t35Extension 1,
    manufacturerCode 256
   },
   productId '54616E6462657267'H,
   versionId '3435'H
  },
  terminal
  {
   nonStandardData
   {
    nonStandardIdentifier h221NonStandard :
    {
     t35CountryCode 130,
     t35Extension 1,
     manufacturerCode 256
    },
    data '54616E6462657267'H
   }
  },
```

```
  mc FALSE,
  undefinedNode FALSE
 },
 terminalAlias
 {
  h323-ID : "System.Unit",
  dialedDigits : "7035551234"
 },
 endpointVendor
 {
  vendor
  {
   t35CountryCode 130,
   t35Extension 1,
   manufacturerCode 256
  },
  productId '54616E6462657267'H,
  versionId '3435'H
 },
 cryptoTokens
 {
  cryptoEPPwdHash :
  {
   alias h323-ID : {"username", {0, 0, 0, 0}},
   timeStamp 1124718941,
   token
   {
    algorithmOID { 1 2 840 113549 2 5 },
    paramS
    {
    },
    hash '00010010 10111111 00000011 01110110 01010010 00110001 00010011
10110101 000100 ...'B
   }
  }
 },
 keepAlive FALSE,
 willSupplyUUIEs FALSE,
 maintainConnection FALSE,
 supportsAltGK NULL,
 genericData
 {

  {
   id nonStandard : '20DF8903596F45199F2773C0A59274AF'H,
   parameters
   {

    {
     id nonStandard : '20DF8903596F45199F2773C0A59274AF'H,
     content raw :
'3C617373656E743E3C617373656E745F747970653E636C69656E743C2F617373656E74
...'H
    }
   }
  }
 }
}
```

- Registration Confirm sent from gatekeeper

At the point where the gatekeeper receives the registration request with the endpoint credentials, it will then confirm those credentials against what it has stored within its authentication database (either locally or through the LDAP server, depending on how it is configured). Once the credentials are then verified, the gatekeeper will confirm back all of the details within the registration.

```
238705830 [GKRAS-0] [10.1.6.210:1719] Sent RCF

238705830 [GKRAS-0] ASN.1 PDU:

value RasMessage ::= registrationConfirm :
{
 requestSeqNum 2239,
 protocolIdentifier { 0 0 8 2250 0 5 },
 callSignalAddress
 {
  ipAddress :
  {
   ip '0A010225'H,
   port 1720
  }
 },
 terminalAlias
 {
  h323-ID : "system.unit",
  dialedDigits : "7035551234",
  h323-ID : "system.unit@company.com",
  h323-ID : "7035551234@company.com"
 },
 gatekeeperIdentifier "GK A",
 endpointIdentifier "EP_b8c61c680000026f0e3a5ca6",
 alternateGatekeeper
 {

  {
   rasAddress ipAddress :
   {
    ip '0A010226'H,
    port 1719
   },
   gatekeeperIdentifier "Alternate 1",
   needToRegister TRUE,
   priority 1
  }
 },
 timeToLive 60,
 willRespondToIRR FALSE,
 maintainConnection FALSE,
 genericData
 {

  {
   id nonStandard : '20DF8903596F45199F2773C0A59274AF'H,
   parameters
   {

    {
```

```
        id nonStandard : '20DF8903596F45199F2773C0A59274AF'H,
        content raw :
'3C617373656E743E3C617373656E745F747970653E7365727665723C2F617373656E74
...'H
      }
    }
  }
 }
}
```

- Registration Reject sent from gatekeeper

If an endpoint provides incorrect credentials, the gatekeeper will send the following Registration Reject (RRJ) to the endpoint.

```
value RasMessage ::= registrationReject :
{
 requestSeqNum 1928,
 protocolIdentifier { 0 0 8 2250 0 5 },
 rejectReason securityDenial : NULL
}
```

3.2 Restricting Network Access

Once an endpoint is registered within an H.323 network, it will then gain access to many different network resources, including firewall traversal, gateways, MCUs, etc. It may be required, however, to restrict registered endpoints from accessing these network resources once registered into the network.

3.2.1 Bandwidth Management

In addition to securing an IP network, an administrator must also consider the ability to model the bandwidth available within that IP network. No two IP networks are exactly the same, therefore the ability to utilize network infrastructure to truly model the existing network and restrict bandwidth to work within that network is vital to the success of IP video calls.

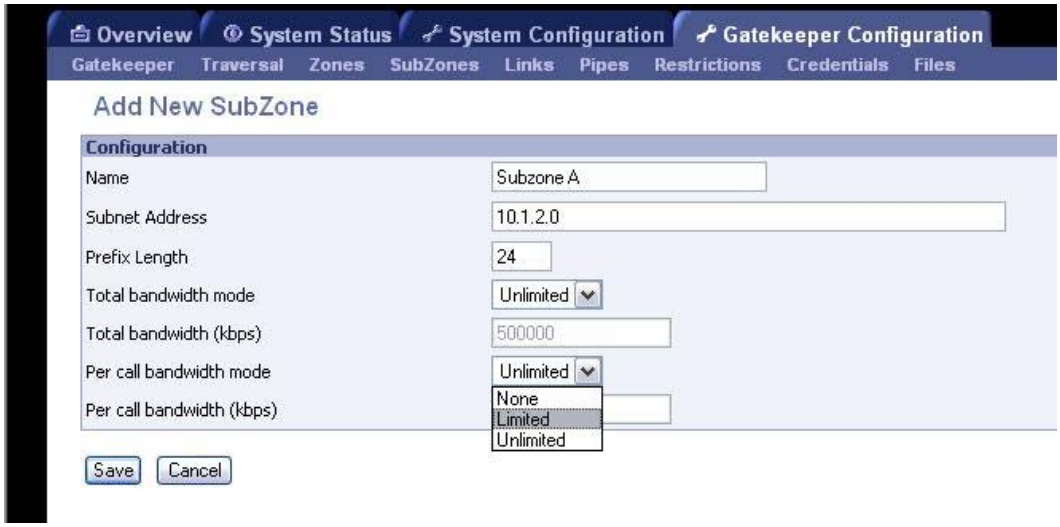
Through the use of subzones, links and pipes, the gatekeeper and Border Controller can be configured to truly mimic that particular network setup by restricting bandwidth to specific network entities as well as the outside world. The network modeling feature within the gatekeeper allows you to control the bandwidth on both a call-by-call basis as well as the total bandwidth funneled through the gatekeeper for all calls.

3.2.1.1 Subzones

The first entity within the gatekeeper that allows the administrator to regulate bandwidth usage are subzones. Within the TANDBERG Gatekeeper, subzones are implemented to act as areas within the gatekeeper that are subject to individual bandwidth restrictions and measurements. Subzones are defined by determining the network address and subnet mask of the endpoints that are going to register within that specific subzone. When an endpoint registers to the gatekeeper, the gatekeeper compares the specific IP address of that endpoint to the IP subnets that are associated with the specific subzones within that gatekeeper and, if the IP address of the endpoint

falls within the IP subnet of a specific subzone on the gatekeeper, the endpoint is then placed into the subzone (*Note:* an endpoint can only register to one single subzone – if an IP address falls within two different subzones (as dictated by the subnet masks configured within each), the endpoint will be placed into the subzone with the more specific subnet mask for that registration; e.g. a subnet mask of 255.255.255.255 will be used instead of another subnet mask of 255.255.255.0). If no subzone is applicable to the specific IP address of the endpoint, the endpoint will be registered into the Default Subzone.

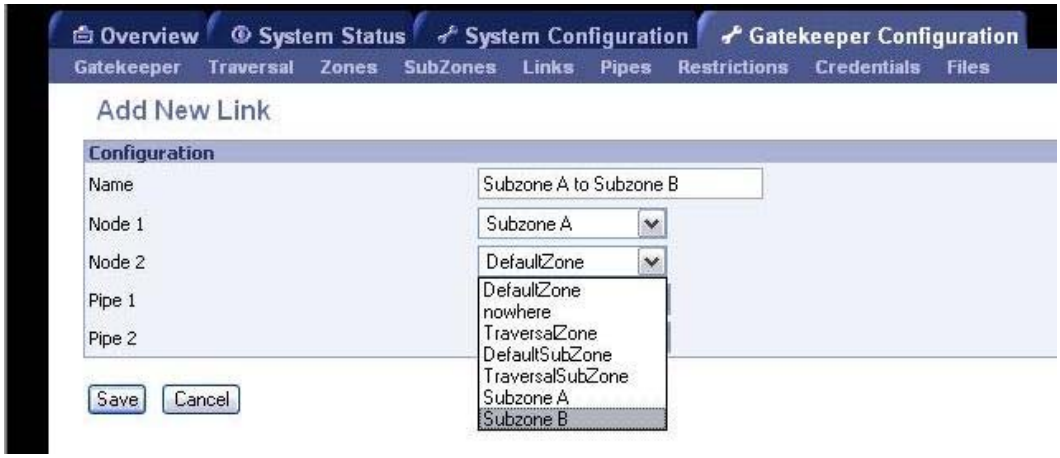
Note: subzones are not individual zones within the same physical gatekeeper – they are only used for bandwidth management.



3.2.1.2 Links

If the goal of the subzones within the gatekeeper is to create areas of different bandwidth characteristics, the goal of the links is to connect those different areas together for call routing. Links create a call-route path within the gatekeeper between two different subzones or between a subzone and an external zone (e.g. a neighboring gatekeeper). If a subzone within the gatekeeper is completely isolated from the rest of the subzones and zones of the gatekeeper (e.g. no links connect that particular subzone to the other areas), no calls will be successful from any endpoint registered to that subzone to any other endpoint registered to another subzone or zone.

Whenever a subzone is added to the gatekeeper, the system will automatically create a link between that subzone and the default subzone. The reason for this link creation is to ensure that, by default, calls will connect from all subzones added to the configuration. These links can be removed, if desired in order to create custom routes between the different subzones.



3.2.1.3 Pipes

Using pipes, an administrator is able to impose bandwidth restrictions to specific links on both a per-call and total call basis. Once created, pipes are then assigned to specific links in order to impose the bandwidth restrictions to specific call instances.

3.2.2 Network Design

By combining all of the principles as discussed above, an administrator can truly create a dynamic network that will limit the bandwidth of the calls in order to coincide with the bandwidth limitations that exist throughout the network.

When adding subzones to the gatekeeper or Border Controller, the system will automatically create links from those subzones to the Default Subzone in order to promote call connectivity during the design phase of the network. While these links can be removed in order to enhance the call control, they do not have to be modified in order for calls to simply connect. In addition, when a link is created, it will not have any pipes applied by default. When no pipes are applied to a link, the gatekeeper does not apply any bandwidth limitation to that specific link, meaning that all calls will not be restricted based on bandwidth and will be allowed to connect at the bandwidth they request.

Depending on how the network is to be designed, this may or may not be desirable. The following examples of network layouts are meant to show how networks can be set up and what the reason for the setup is. These are not meant to be copied verbatim for deployment as they do not take all network contingencies into account.

3.2.2.1 All Calls Route Through Main Office

When calls route through the main office, an administrator will need to create multiple links and subzones for all remote systems. In addition, it will become a requirement to limit calls on a per-call and total call basis in order to prevent oversubscription of the bandwidth to all networks involved.

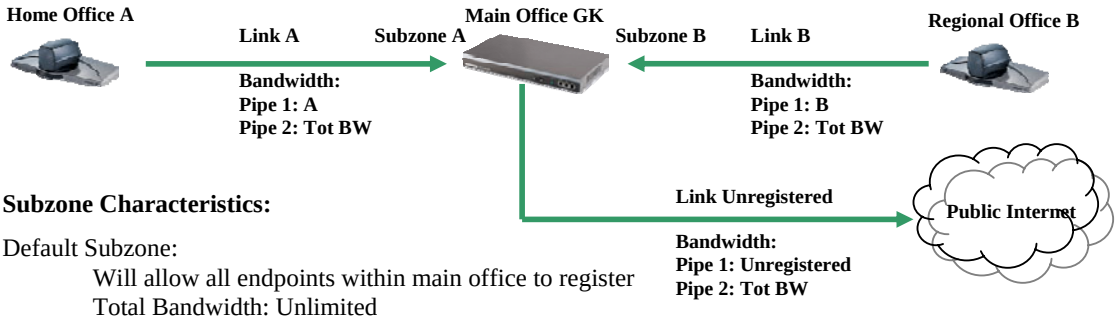
In this example, all networks do not want to restrict the total call or per-call bandwidth of any internal calls. However, the Home Office A only has a total bandwidth out to the public Internet of 512k, meaning we will need to restrict the total and per-call bandwidth of this site whenever it calls out to any public systems.

Regional Office B has a relatively large bandwidth to the public internet in 1.544Mbps, but that bandwidth is actually in the form of a dedicated T1 back to the main office, meaning that all bandwidth destined to the public internet will go through the Main Office network. In order to maintain functionality of this pipe, we will need to restrict all calls to only 384k per call to ensure that one call does not tie up the entire bandwidth.

In order to restrict the total calls of the network, we need to ensure that the entire network is limited to 4 Mbps as that is the maximum amount of bandwidth in the main office. Since all outbound calls will go to the Main Office GK, we need to restrict all calls that go through this gatekeeper to 4096k.

In order to limit the call bandwidth of all calls to unregistered systems, we will restrict the link from the DefaultSubzone to the DefaultZone. Because all calls from the Regional Offices route through the Main Office in the first place, no extra configuration will be needed.

Network Diagram;



Subzone Characteristics:

Default Subzone:
- Will allow all endpoints within main office to register
- Total Bandwidth: Unlimited
- Per-Call Bandwidth: Unlimited

Subzone A:
- Will allow all endpoints from Home Office A to register (all IPs start with 125.164.5.129-255)
- Total Bandwidth: Unlimited
- Per-Call Bandwidth: Unlimited
- Subnet Address: 125.164.5.128
- Subnet Mask: 255.255.255.128

Subzone B:
- Will allow all endpoints from Regional Office B to register (all IPs start with 12.48.62.0-255)
- Total Bandwidth: Unlimited
- Per-Call Bandwidth: Unlimited
- Subnet Address: 12.48.62.0
- Subnet Mask: 255.255.255.0

Link Characteristics:

Link A:
- Node 1: Subzone A
- Node 2: Default Subzone
- Pipe 1: Pipe A
- Pipe 2: TotBW

Link B:
- Node 1: Subzone B
- Node 2: Default Subzone
- Pipe 1: Pipe B
- Pipe 2: TotBW

Link Unregistered:
- Node 1: Default Subzone
- Node 2: Default Zone
- Pipe 1: Unregistered
- Pipe 2: TotBW

Pipe Characteristics:

Pipe A:
- Per-Call Bandwidth: 256k
- Total Call Bandwidth: 512k

Pipe B:
- Per-Call Bandwidth: 384k
- Total Call Bandwidth: 1544k

TotBW:
- Per-Call Bandwidth: Unlimited
- Total Call Bandwidth: 4096k

Pipe Unregistered:
- Per-Call Bandwidth: 384k
- Total Call Bandwidth: Unlimited (because call is routing through Main Office)

3.2.2.2 Mesh Network of Both Main and Regional Offices

In a mesh network, all calls will route directly from the source office to the destination office, not necessarily going through the main office. In this instance, it is important to create links and pipes that will be specific to the offices in which they connect in order to prevent unnecessary bandwidth limiting on any of the calls.

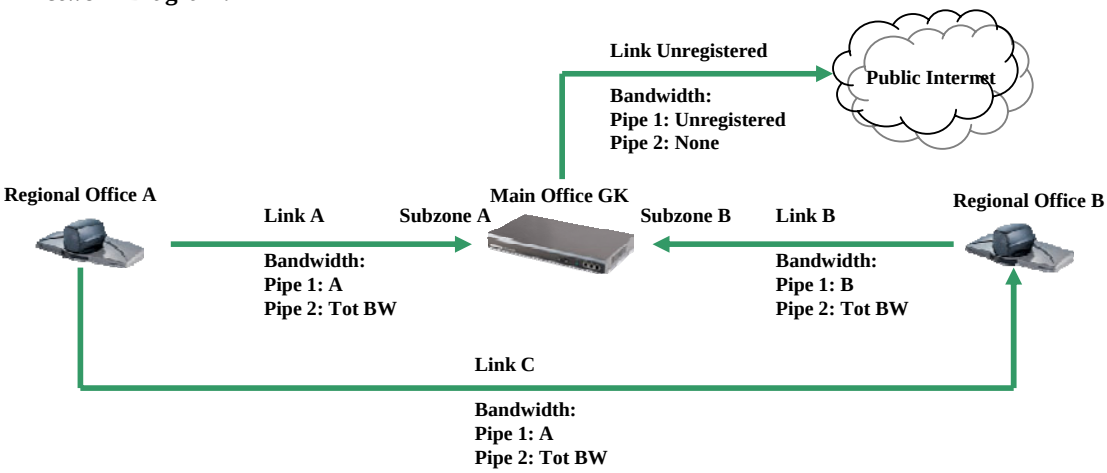
In this example, all networks do not want to restrict the total call or per-call bandwidth of any internal calls. However, the Regional Office A only has a total bandwidth out to the public Internet of 2Mbps, meaning we will need to restrict the total and per-call bandwidth of this site whenever it calls out to any public systems. We will restrict all calls outbound from this office to a total of 384k in order to prevent a single call from exceeding the bandwidth limitation of the entire office and allow multiple endpoints to be connected at the same time.

Regional Office B has a relatively large bandwidth to the public internet in 4Mbps. In order to maintain functionality of this pipe, we will need to restrict all calls to only 512k per call to ensure that one call does not tie up the entire bandwidth.

While the main office will not handle all of the call bandwidth of these remote offices, we need to ensure, however, that it will impose restrictions that it will need to ensure that its network is not over-subscribed at any time. Since the main office has the most bandwidth of over 8Mbps, we will impose a per-call restriction of 768 as the number speed of all calls is not as crucial to the entire network.

In order to limit the call bandwidth of all calls to unregistered systems, we will restrict the link from the DefaultSubzone to the DefaultZone. Because all calls from the Regional Offices will route through the Default Subzone to get to the DefaultZone, this bandwidth restriction will be in effect for all endpoints inside the office network.

Network Diagram:



Subzone Characteristics:

Default Subzone:
- Will allow all endpoints within main office to register
- Total Bandwidth: Unlimited
- Per-Call Bandwidth: Unlimited

Subzone A:
- Will allow all endpoints from Regional Office A to register (all IPs start with 12.16.5.0-63)
- Total Bandwidth: Unlimited
- Per-Call Bandwidth: Unlimited
- Subnet Address: 12.16.5.0
- Subnet Mask: 255.255.255.192

Subzone B:
- Will allow all endpoints from Regional Office B to register (all IPs start with 64.15.3.0-127)
- Total Bandwidth: Unlimited
- Per-Call Bandwidth: Unlimited
- Subnet Address: 64.15.3.0
- Subnet Mask: 255.255.255.128

Link Characteristics:

Link A:
- Node 1: Subzone A
- Node 2: Default Subzone
- Pipe 1: MainOffice
- Pipe 2: RegOfficeA

Link B:
- Node 1: Subzone B
- Node 2: Default Subzone
- Pipe 1: MainOffice
- Pipe 2: RegOfficeB

Link C:
- Node 1: Subzone A
- Node 2: Subzone B
- Pipe 1: RegOfficeA
- Pipe 2: RegOfficeB

Link Unregistered:
- Node 1: Default Subzone
- Node 2: Default Zone
- Pipe 1: Unregistered
- Pipe 2: None

Pipe Characteristics:

Pipe MainOffice:
- Per-Call Bandwidth: 768k
- Total Call Bandwidth: 8192k

Pipe RegOfficeA:
- Per-Call Bandwidth: 384k
- Total Call Bandwidth: 2048k

Pipe RegOfficeB:
- Per-Call Bandwidth: 512k
- Total Call Bandwidth: 4096k

Pipe Unregistered:
- Per-Call Bandwidth: 384k
- Total Call Bandwidth: 8192k (because call is routing through Main Office)

3.2.3 Authorization

In addition to Authentication, the TANDBERG Gatekeeper and Border Controller also supports the ability to determine which endpoints can access which resources within a network. For example, an administrator may want to prevent certain end users from accessing some of the more "expensive" resources on the network, such as gateways and/or MCUs.

The TANDBERG Gatekeeper and Border Controller support Call Processing Language (CPL) as defined within RFC 3880 as defined by the IETF[2]. This RFC allows for the development and deployment of an XML file that will dictate how the gatekeeper/Border Controller handles calls as they are received by the device. While the entire RFC has not been implemented as of the N3/Q2 release, basic functionality has been implemented in order to allow for the acceptance, denial and forwarding of calls.

CPL works within the gatekeeper upon determination of the call connectivity. Once a gatekeeper receives the call request (either an ARQ from a registered endpoint or LRQ from a neighboring gatekeeper) and can process the call (e.g. the destination endpoint is registered locally to that particular gatekeeper), the source and destination call information will be passed into the CPL engine (*Note:* if authentication is turned on within the gatekeeper and the call is passed in from an unauthenticated neighboring gatekeeper, the source information of the call will not be passed to signal that the call is from an unauthenticated entity). Upon receipt of the source and destination information, the gatekeeper will search the CPL code for the first matching criteria. Once found, the gatekeeper will process the CPL logic, exit the CPL engine and execute the call based on the modifications that were made to the call, if any.

Within the N3/Q2 software version of the TANDBERG Gatekeeper and Border Controller, the CPL engine supports the ability to connect a call, reject a call or forward the call to a different destination. *Note:* CPL scripts are not case sensitive as far as source and destination naming.

3.2.3.1 Initial CPL Script

All CPL scripts must have an XML heading that signifies that the following code is both XML and that it follows the CPL standard, as defined within RFC 3880. This code will not vary within the different CPL scripts, as such it can be copied and pasted into each one of the code sets without any modification at all.

```
<?xml version="1.0" encoding="UTF-8"?>
    <cpl xmlns="urn:ietf:params:xml:ns:cpl"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="urn:ietf:params:xml:ns:cpl cpl.xsd ">
```

3.2.3.2 Connecting a Call

Within the CPL engine of the TANDBERG Gatekeeper and Border Controller, you can choose to tell the system to explicitly connect a call or, if no action is performed, the CPL engine will allow the call to go through untouched, thereby connecting the

[2] For specific information on the RFC 3880, please reference the RFC itself. This can be found at: http://www.apps.ietf.org/rfc/rfc3880.html.

call as dialed. The following sample of XML code tells the CPL engine to allow all calls to system *Darryl* to connect as originally dialed:

```
<incoming>
   <address-switch field="destination">
     <address is="darryl">
       <proxy/>
     </address>
   </address-switch>
</incoming>
```

3.2.3.3 Rejecting a Call

The CPL engine also has the ability to reject calls from being connected based on either the source or destination information of the call. Within the CPL script, the command *<reject/>* will tell the CPL engine to completely reject the call. The following sample XML code shows a rejection of all calls from system *Barney* to system *Fred*:

```
<incoming>
   <address-switch field="destination">
     <address is="fred">
       <address-switch field="origin">
         <address is="barney">
           <reject/>
         </address>
       </address-switch>
     </address>
   </address-switch>
</incoming>
```

In order to reject the call, the gatekeeper will then respond with either an LRJ or ARJ message (depending on the incoming message) to the source system and the call will fail.

3.2.3.4 Call Proxy/Forward

In order to forward a call to a different destination, the policy script must be configured to change the call source information to a different address (*Note:* the call forwarding feature is an always on function; in other words, all calls will be forwarded upon receipt. As of this release, there is no support for "forward on busy" or "forward on no answer" functionality). This is done similar to the following snippet of code taken from an actual policy file. This code will forward a call that was originally dialed to connect to the endpoint *Barney* to another system named *Receptionist*:

```
<incoming>
   <address-switch field="destination">
     <address is="barney">
       <location clear="yes" url="h323:receptionist">
         <proxy/>
       </location>
     <address>
   </address-switch>
</incoming>
```

3.2.3.4 Call Processing

It is important to note that the gatekeeper will only run through the CPL engine one time. Once the first pass is completed, the gatekeeper will exit the CPL engine and process the call as a normal H.323 call. For example, if CPL is configured to forward

all calls from *Brian* to *John*, CPL will change the destination address to *John*, exit and process the call – it will not re-enter the CPL engine and re-process the call for *John* once the destination address has been changed.

Another consideration that needs to be taken into account during the design of CPL scripts is that the CPL engine executes the script from top to bottom. Once the engine finds the first match for either the source or destination of the call, it will then execute that command and exit – it will not try and find a "better" match further down in the script. When writing the script, it is necessary to place the "more important" matches at the top of the script and the ones that are not as important further down. For example, if it is more important to forward any inbound calls to *Brian* than it is to deny any outbound calls from *John*, then the script will need to be written such that the destination of the calls are checked before the source in order to ensure that all calls to *Brian* are forwarded immediately.

Using a combination of all of the methods as outlined above, an administrator can truly control their network by only allowing specific calls to connect, forward calls to other destinations and deny access to resources and endpoints based on source or destination addressing.

For more information on XML and how to use it within the CPL standard, please reference document number D50383 – XML and TANDBERG CPL.

4. Redundancy

As video becomes more prevalent within an organizations day-to-day business practices, ensuring the success of those video calls becomes as crucial. In order to guarantee this success, considerations must be taken in the event that there is a loss of service on the IP network side.

While redundancy on the IP infrastructure has been utilized for quite some time, it is only relatively recently that these considerations have continued into the H.323 world. The primary method that redundancy has been implemented within H.323 has been through the alternate gatekeeper protocol. Alternate gatekeepers provide a method of being able to duplicate the H.323 infrastructure. This duplicate infrastructure would then assume the role of the primary infrastructure in the event of a catastrophic failure.

Within the N3 and Q2 release of software for the TANDBERG Gatekeeper and Border Controller, respectively, TANDBERG has implemented the ability to create a truly redundant H.323 infrastructure.

4.1 Alternate Gatekeepers

Alternate gatekeepers are used within the H.323 infrastructure to ensure continued connectivity in the event that the primary gatekeeper suffers a loss of communication (e.g. power loss, network outage, system failure).

When an endpoint registers to a gatekeeper, it will receive a listing of alternate gatekeepers back within the registration confirm message (RCF). These alternate addresses are then stored within the endpoint for use in case communication with the primary gatekeeper fails – no action is taken immediately.

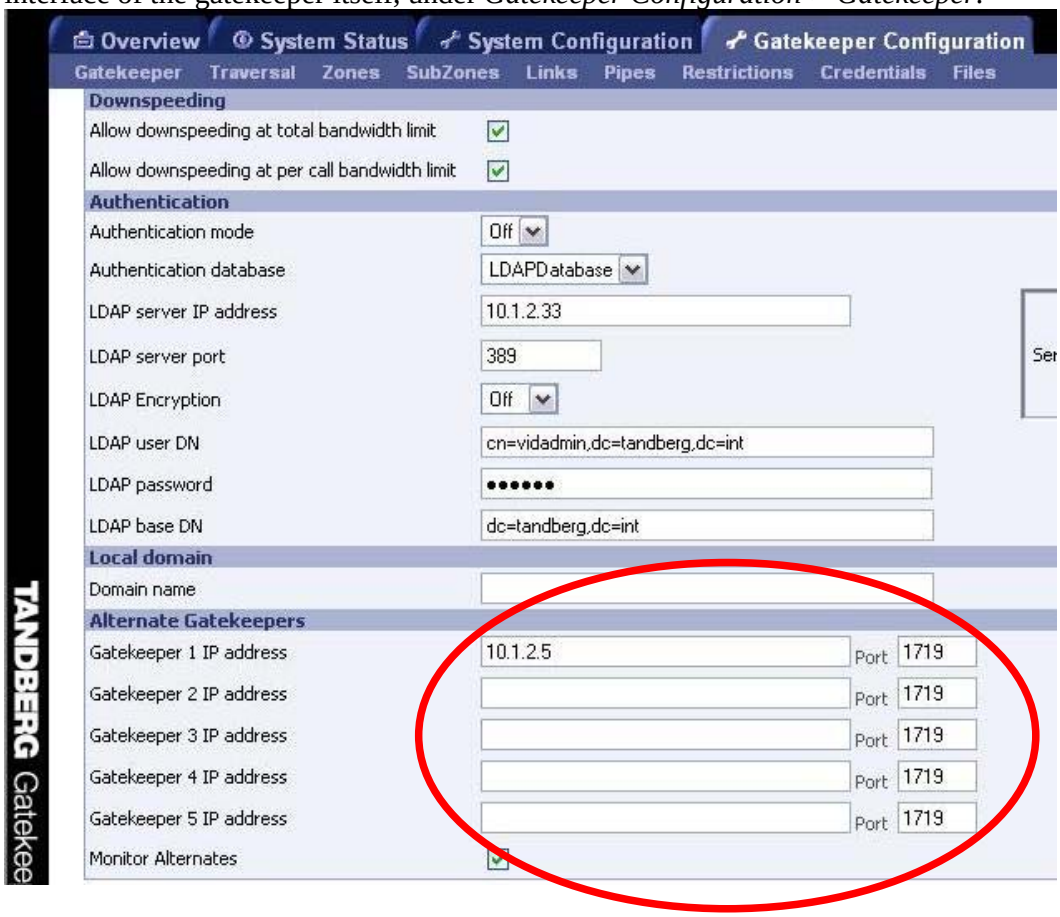
Upon detection of a failed gatekeeper, the endpoint will then attempt to register to the alternate gatekeepers it received in the RCF from the primary gatekeeper. The endpoint will begin with the first address in the list and will proceed down the list until it receives a response from one of the gatekeepers. Once the endpoint receives a response from a gatekeeper, it will begin the registration process with that gatekeeper.

Note: once an endpoint finds an active gatekeeper within the alternate list, and tries to register to that gatekeeper, it will no longer search for another active gatekeeper – it will try to register to that active gatekeeper. If that particular registration fails, the endpoint will not then proceed to the next gatekeeper – it will continue to sit in an unregistered state and not be an active member in the H.323 network. In order to prevent this from occurring, an administrator must configure all of the gatekeepers identically such that an endpoint that is permitted to register to one of the gatekeepers can register to any of the others if necessary within a fail-over scenario.

Once the endpoint registers into one of the alternate gatekeepers, it will remain registered to that gatekeeper until communication is lost with that gatekeeper or it is manually moved to another gatekeeper by the administrator. The endpoint will not attempt to re-register to the primary gatekeeper if that gatekeeper returns to active status.

Upon receipt of the alternate IP addresses from the gatekeeper, the endpoint will store them and take no further action. However, if the endpoint itself loses power or is recycled, those stored alternate IPs are lost. Upon bootup of the endpoint, it will attempt to register to the primary gatekeeper. If that gatekeeper is not active at the time of registration attempt, the endpoint will sit in an idle and unregistered state and will not register to any of the alternate gatekeepers. The endpoint will continue to attempt to register to the primary gatekeeper every few seconds until it is successful.

The TANDBERG Gatekeeper can support up to five (5) alternate gatekeepers within its configuration. These alternate IP addresses can be configured within the web interface of the gatekeeper itself, under *Gatekeeper Configuration* > *Gatekeeper*.



4.2 Alternate Border Controller

In addition to alternate gatekeepers, TANDBERG also has developed a solution to provide alternate Border Controllers to the Expressway™ solution.

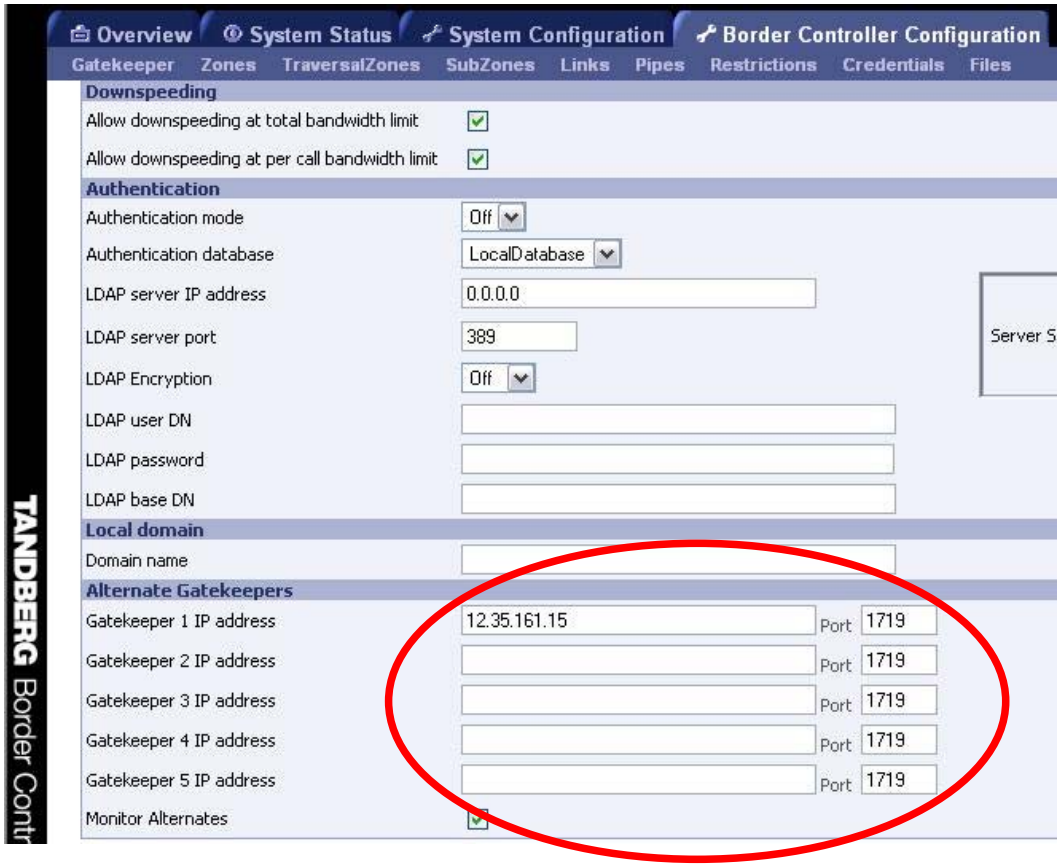
In regards to an Expressway-enabled endpoint, the fail-over process of an alternate border controller works identically to that of the alternate gatekeeper. Please see section 4.1 for more information on how this process functions.

In the firewall traversal scenario, however, the gatekeeper will take a slightly different action when registering to a Border Controller that has alternates configured. Similar to the endpoint, the gatekeeper will register to the Border Controller and receive a list of up to 5 alternate Border Controllers. However, once the gatekeeper receives those alternates within the RCF message, it will immediately go out and attempt to register with all of the alternate Border Controllers. In doing so, the gatekeeper can become registered to up to six different Border Controllers at a single time (the primary and five alternates).

If a gatekeeper is registered to more than one Border Controller, calls will only traverse outbound through what the gatekeeper determines as the *primary* link. In an initial configuration (e.g. the gatekeeper has never lost communication to the primary Border Controller), the primary Border Controller is the Border Controller to which the gatekeeper originally registered.

In the event that the primary Border Controller fails, the gatekeeper will immediately begin sending outbound traversal calls to the secondary gatekeeper (the first gatekeeper listed in the alternate list). That Border Controller will then remain the primary traversal link until either it fails or the traversal configuration is reset on the gatekeeper.

Unlike the endpoints, the gatekeeper stores the alternate addresses within persistent memory, meaning that if the gatekeeper were to recycle, upon boot-up, the gatekeeper will attempt to register to the primary Border Controller. If this registration fails, the gatekeeper will then proceed and register with the alternate Border Controllers it received upon initial registration to the primary Border Controller.


Overview
System Status
System Configuration
Border Controller Configuration
Gatekeeper Zones TraversalZones SubZones Links Pipes Restrictions Credentials Files
Downspeeding
Allow downspeeding at total bandwidth limit
Allow downspeeding at per call bandwidth limit
Authentication
Authentication mode
Off
Authentication database
LocalDatabase
LDAP server IP address
0.0.0.0
LDAP server port
389
LDAP Encryption
Off
LDAP user DN
LDAP password
LDAP base DN
Local domain
Domain name
Alternate Gatekeepers
Gatekeeper 1 IP address
12.35.161.15
Port
1719
Gatekeeper 2 IP address
Port
1719
Gatekeeper 3 IP address
Port
1719
Gatekeeper 4 IP address
Port
1719
Gatekeeper 5 IP address
Port
1719
Monitor Alternates
TANDBERG Border Contr

4.3 Alternate URI

In order to complete the redundant solution for the TANDBERG infrastructure, the ability to support multiple addresses per SRV record was implemented within the N3/Q2 software version.

As it stood previously, the gatekeeper or Border Controller that would perform the URI lookup would receive an IP address for that domain name and would LRQ out to that IP address in order to locate the far end system. Whether or not that public system would receive a response, it would consider its task completed.

With the latest version of software, the gatekeeper and Border Controller now support receiving up to five (5) different IP addresses within the SRV record lookup on the DNS server. Upon receipt of these IP addresses, the public gatekeeper/Border Controller will LRQ out to the first address. If it does not receive a response, the system will then LRQ out to the second address, and so on until it receives a response.

Note: a response consists of either an LCF or an LRJ. Once the public system receives this response, it has completed its lookup and will proceed accordingly (e.g. if it receives an LCF, connect the call; if it receives an LRJ, reject the call).

4.4 Call Routing

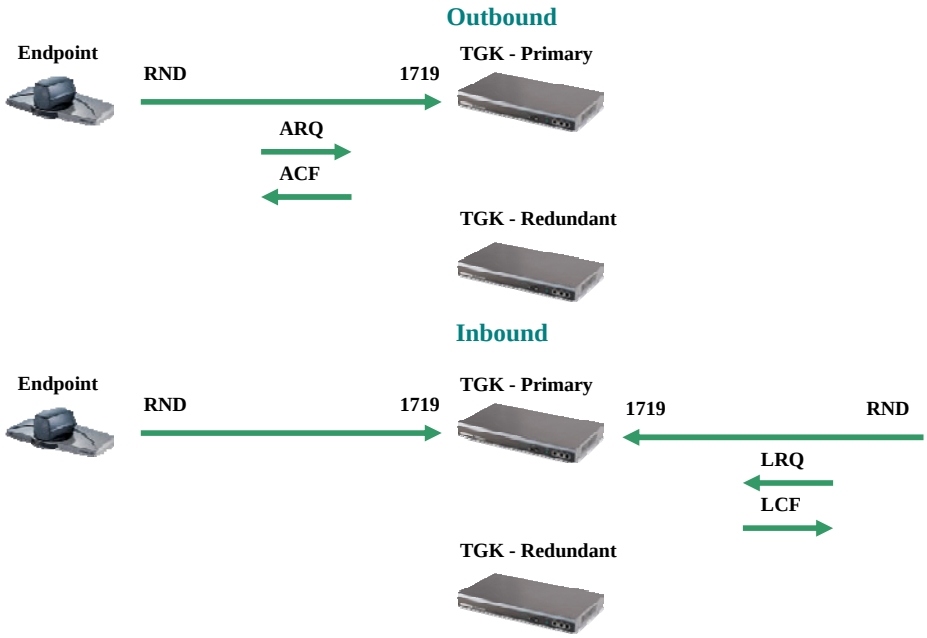
This section will go over how call routing will proceed as the TANDBERG infrastructure proceeds into and out of redundant mode.

Note: for the following sections on call routing and call flow, we will discuss the scenarios with only one redundant box. In actuality, up to five alternates could be defined, but the principles behind the inner workings of one redundant pair versus five are identical.

4.4.1 Alternate Gatekeeper (No Traversal)

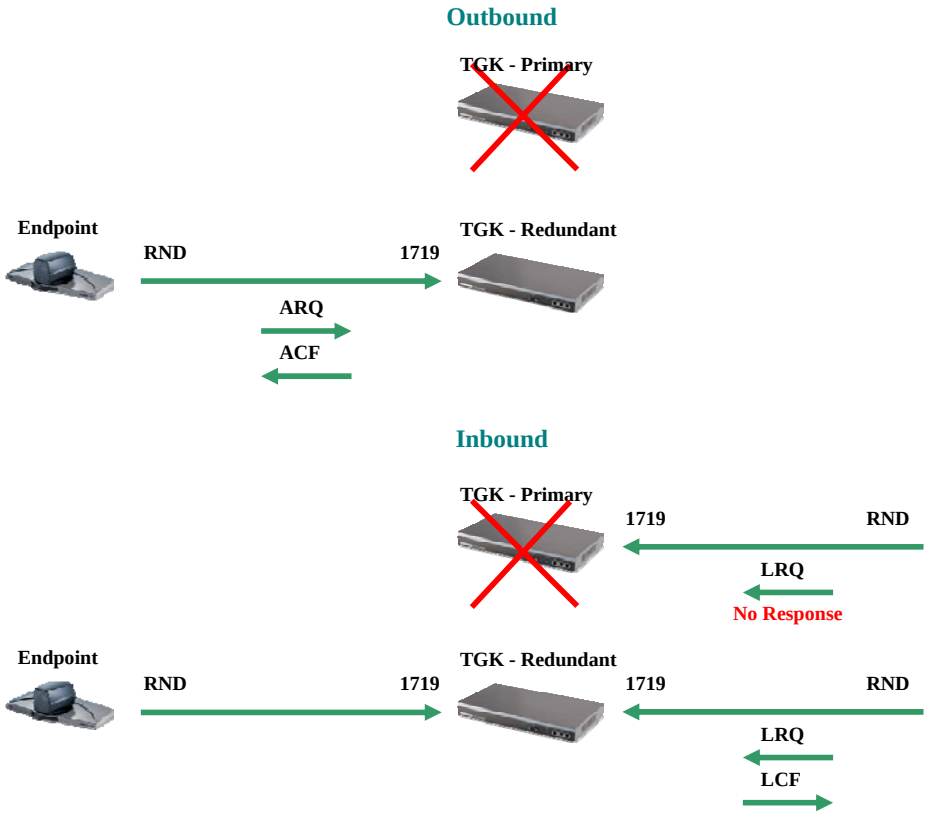
4.4.1.1 Initial deployment

In the alternate gatekeeper redundant deployment, the call routing model is very simplistic. In the initial deployment, all endpoints will be registered to the primary gatekeeper, so all RAS call messages will flow through the main gatekeeper, both on outgoing and incoming calls.



4.4.1.2 Loss of communication with primary gatekeeper

In the event that IP communication fails, the endpoint will then re-register to the redundant gatekeeper. The redundant gatekeeper will then take over with the inbound and outbound call routing.

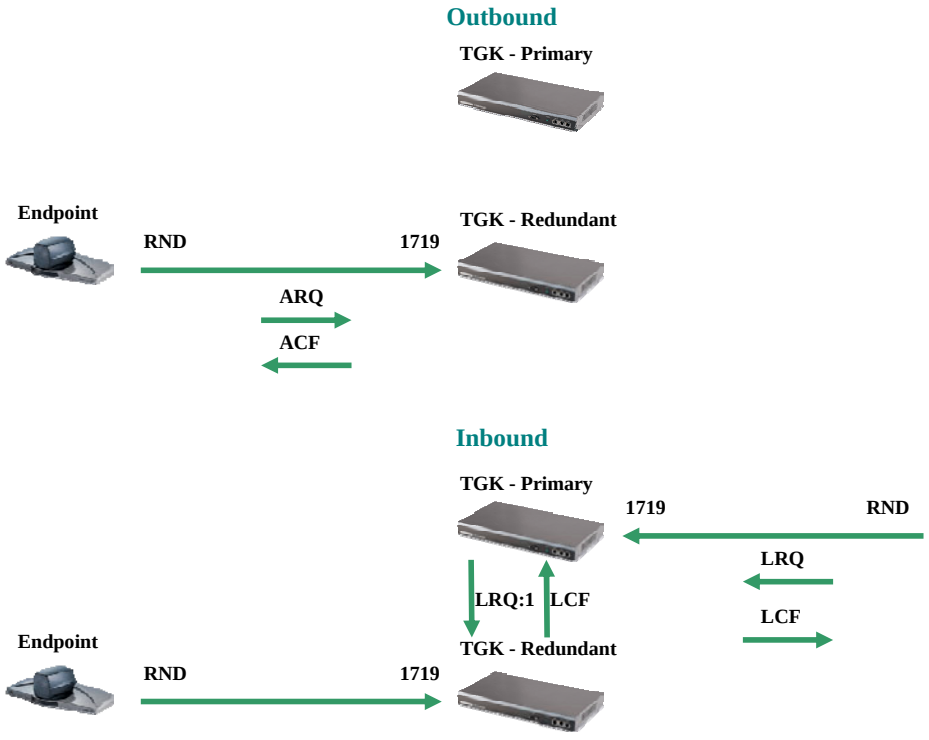


4.4.1.3 Primary gatekeeper communication restored

When the primary gatekeeper is recovered, it will resume operation of running the network. However, most of the endpoints that failed over to the redundant gatekeeper will remain registered to the redundant gatekeeper (until manually moved back to the primary gatekeeper), it will continue to play a major role in call routing within the network. Upon receipt of an LRQ from a remote gatekeeper, the primary gatekeeper will LRQ all of its alternates with an LRQ Hop Count of one (1) to verify whether or not any of the alternates have that endpoint registered locally with them. If one of the alternates confirms back, the primary gatekeeper will then confirm back to the gatekeeper that originally LRQ'd into the network and the call will route accordingly. However, if none of the alternate gatekeepers has the endpoint registered locally, the primary gatekeeper will continue with the call logic as it normally would.

Note: within the F4.x software release, if an endpoint fails over to the redundant gatekeeper, it will send a GRQ to the primary gatekeeper every 20 seconds looking for restoration of communication. When communication is restored between the endpoint and the gatekeeper, the gatekeeper will respond to the GRQ with a GCF; at which time the endpoint will then unregister from the redundant gatekeeper and re-register back to the primary gatekeeper, thus restoring normal operation of the network.

Note: the reason for the LRQ Hop Count of one to the alternate gatekeepers is to ensure that the alternates do not LRQ out to their neighbors and locate the endpoint.



4.4.1.4 Communication lost to redundant gatekeeper

If communication is lost to the redundant gatekeeper, all endpoints will then fall back to the primary gatekeeper and the network will operate as it did initially. This condition will also be met if all endpoints registered to the redundant gatekeeper are manually forced to re-register to the primary gatekeeper.

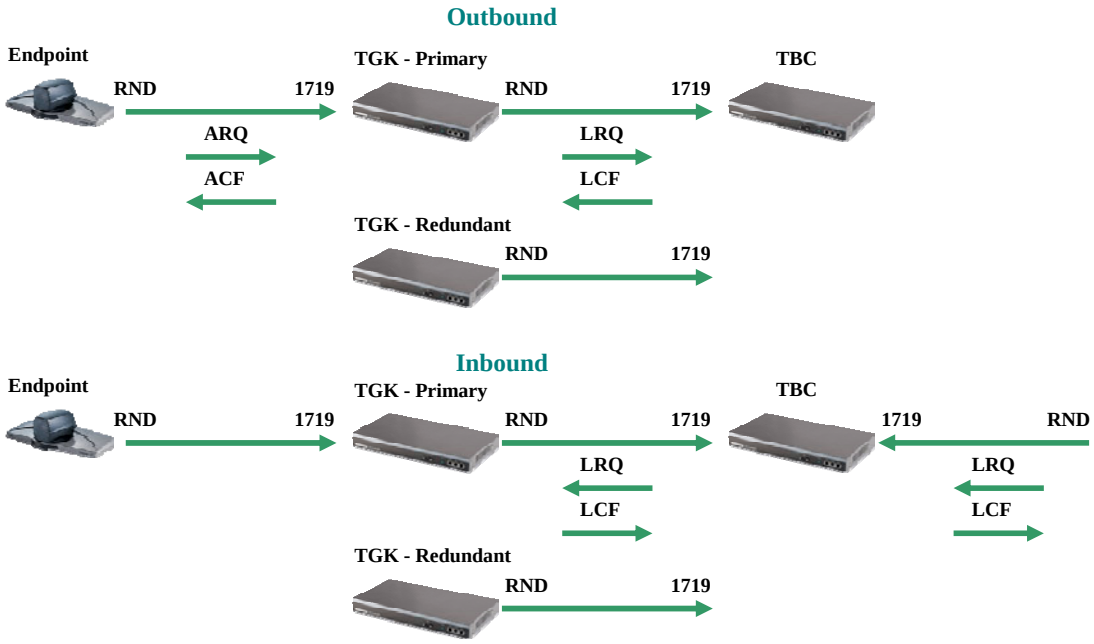
4.4.2 Alternate Gatekeeper (Single Border Controller)

The above solution is the most recognized redundant solution on the market today. Adding traversal capabilities to this solution does not alter it in many ways.

4.4.2.1 Initial Configuration

Within the N3 and Q2 software release for the TANDBERG Gatekeeper and Border Controller, multiple internal gatekeepers can register to a single Border Controller outside the firewall. In order to accomplish this task, both gatekeepers must have the exact same system name, for the system name is the "username" the gatekeeper uses when registering to the Border Controller.

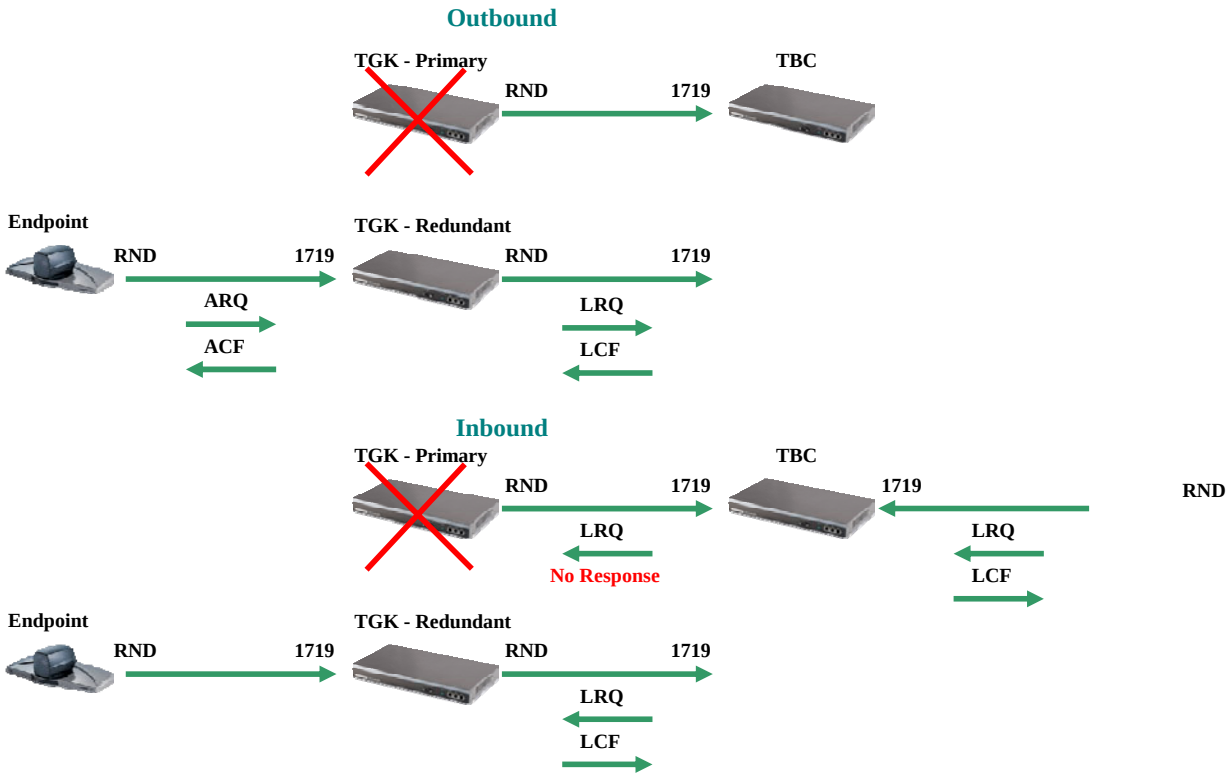
In addition, the Border Controller will only make use of one of the inbound links for allowing calls to traverse inbound. For example, if the Border Controller has two gatekeepers registered to it, calls will only be able to traverse inbound to the gatekeeper that registers first to the Border Controller (the gatekeeper that registers first is considered the *primary* link).



4.4.2.2 Communication lost with primary gatekeeper

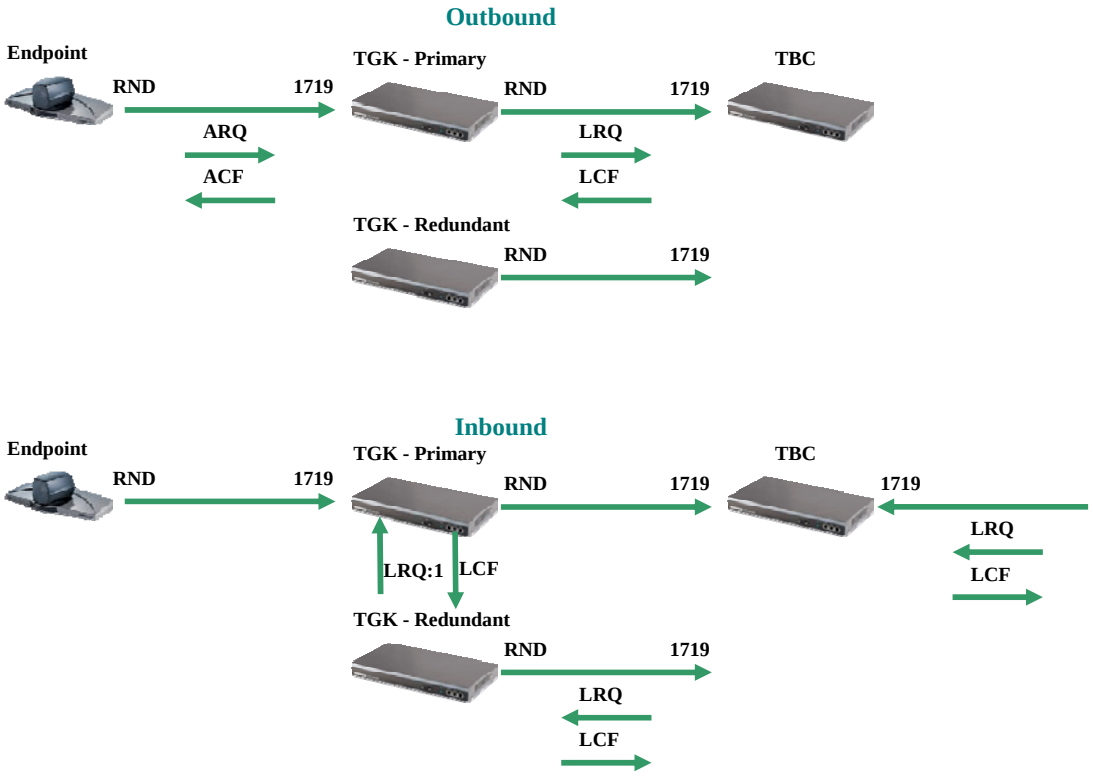
If communication with the primary gatekeeper fails, the secondary gatekeeper will assume the responsibilities of the primary gatekeeper for both inbound and outbound calling.

Note: at this point, all endpoints registered with the primary gatekeeper will also fail over and register with the redundant gatekeeper through the alternate gatekeeper process, as discussed above.



4.4.2.3 Communication with primary gatekeeper restored

Once communication is restored to the primary gatekeeper, it will automatically re-register to the Border Controller outside the firewall. Although this link may be used for outbound traversal calls, it will not be used for inbound traversal calls, even if endpoints happen to register back to the primary gatekeeper. Since the redundant gatekeeper traversal link is now the link that has been registered the longest, it will remain the inbound traversal link until that link is interrupted or the entire traversal configuration is reset.



4.4.2.4 Communication with redundant gatekeeper fails

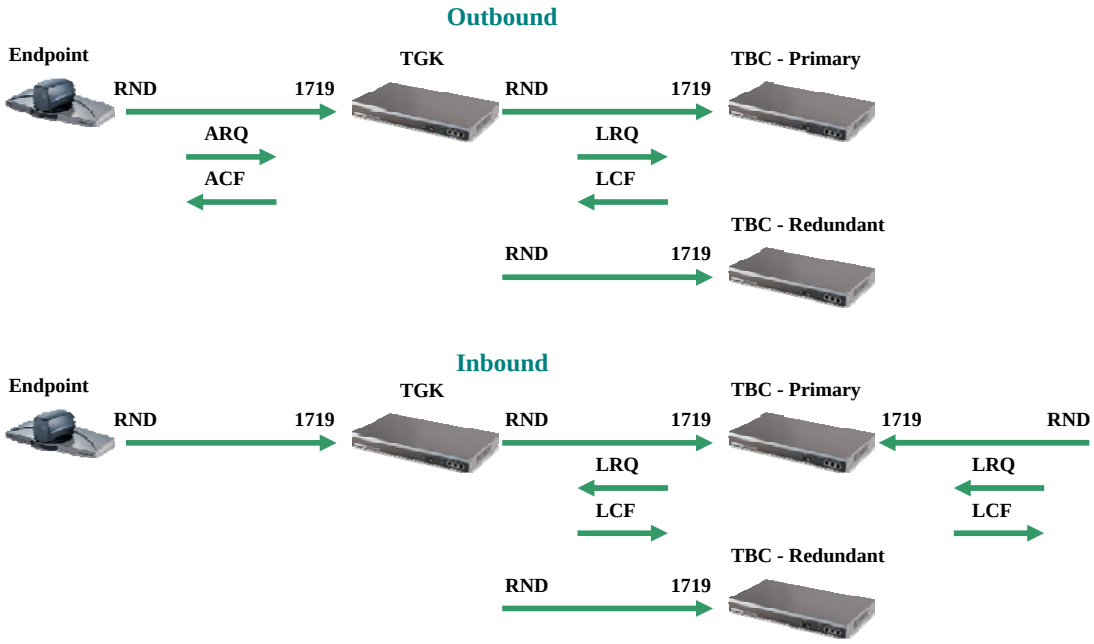
If, after communication is restored to the primary gatekeeper, communication is lost with the secondary gatekeeper, all of the endpoints that were registered with the redundant gatekeeper will revert back to the primary gatekeeper. The inbound traversal link will also be reset to the primary gatekeeper since that link has now been registered to the Border Controller for the longest amount of time. In essence, the condition shown in section 0 has been restored.

4.4.3 Alternate Border Controller (Single Gatekeeper)

When deploying a redundant solution on the Border Controller side of the Expressway solution, the call routing is not much different than that of an alternate gatekeeper setup. However, it can be vital to understand how the gatekeeper is going to route outbound calls through the solution.

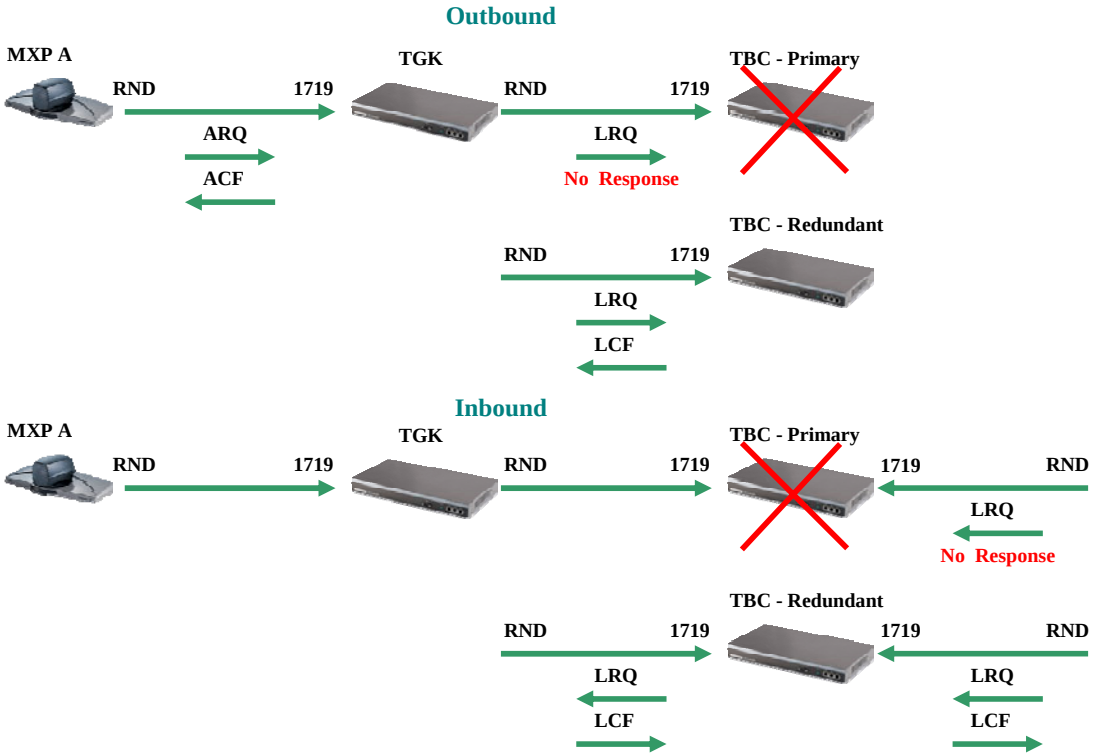
4.4.3.1 Initial Configuration

When routing an outbound call through the solution, the gatekeeper will always route the call through what it determines as the primary Border Controller. In the initial setup, the Border Controller that the endpoint initially registered to is considered the primary link to the outside world.



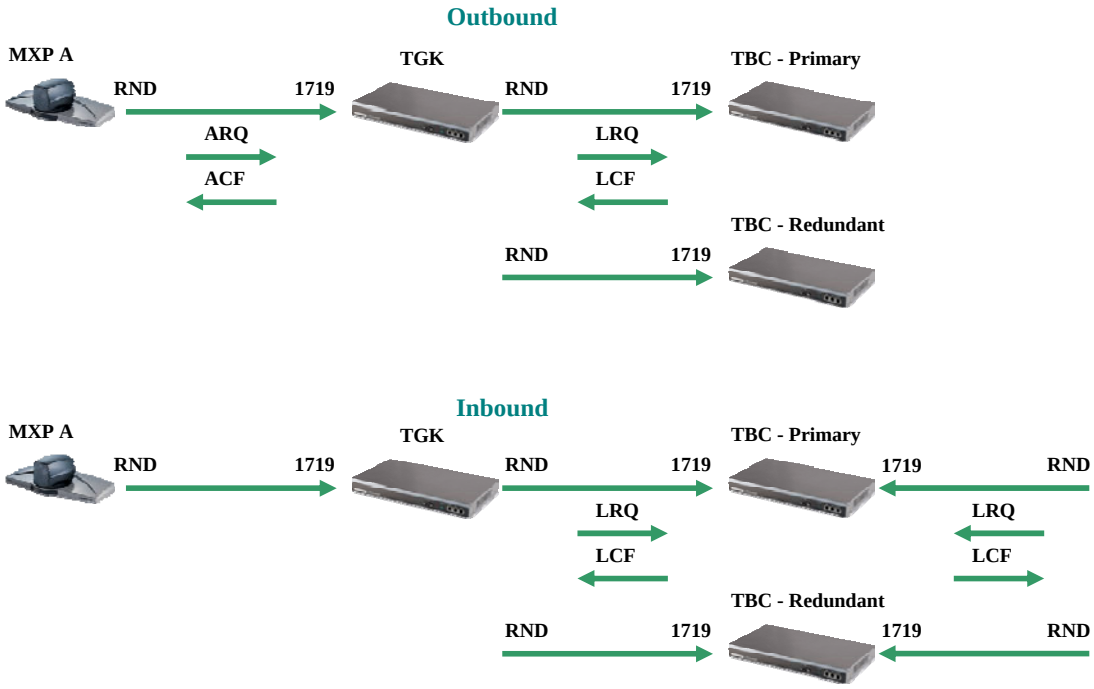
4.4.3.2 Failed communication with primary Border Controller

In the event that the primary Border Controller fails, the redundant Border Controller will assume the responsibilities of the primary automatically. Since the internal gatekeeper is already registered, calls will immediately traverse outbound through the backup Border Controller. Inbound calls will need to hit the redundant Border Controller in order to traverse inbound.



4.4.3.3 Communication with primary Border Controller restored

When the primary Border Controller is recovered, the gatekeeper will automatically re-register and resume using the primary Border Controller as the primary link. At this time, no future outbound calls will traverse over the link to the secondary Border Controller. This will essentially restore the original condition as addressed in section 4.4.3.1.



4.4.3.4 Communication with redundant Border Controller Fails

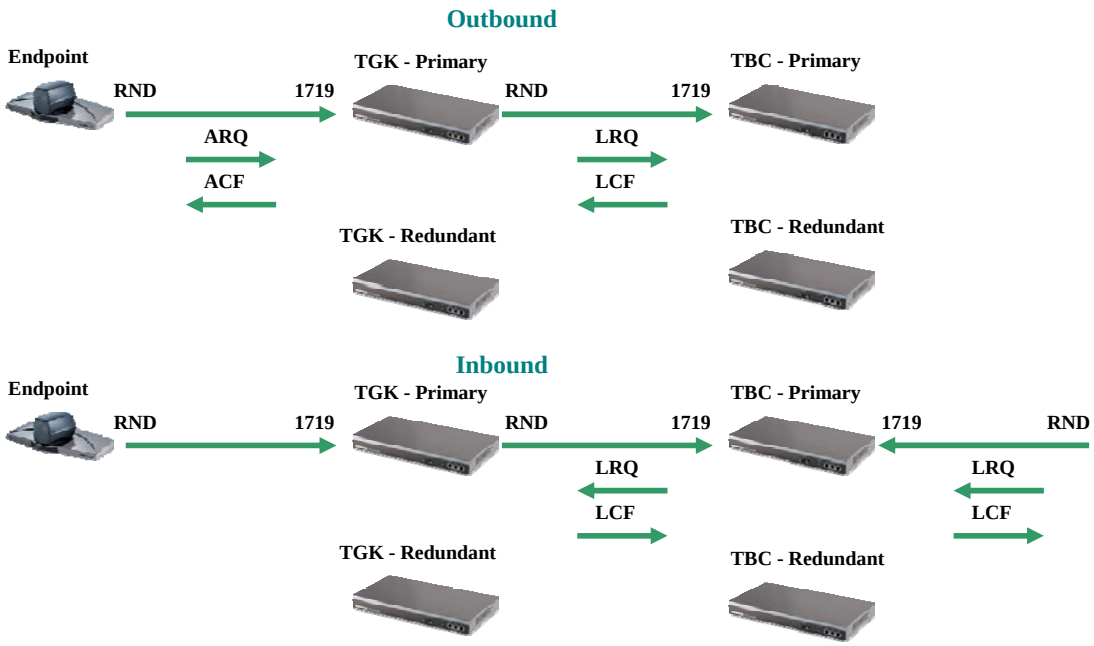
If communication fails with the redundant Border Controller, it will have no effect on the call routing plan as discussed in the section previous. All inbound and outbound traversal calls will continue to traverse over the link between the primary gatekeeper and Border Controller.

4.4.4 Alternate Gatekeeper and Alternate Border Controller

When thinking about an entirely redundant firewall solution, it becomes necessary to combine all of the concepts as discussed above. For the endpoints inside the firewall, we will have alternate gatekeepers and for redundant traversal, we will have a combination of alternate gatekeepers and Border Controllers. **Note:** in order to simplify the following diagrams, only call routing lines will be drawn. For any other communication, please see the diagrams as laid out above.

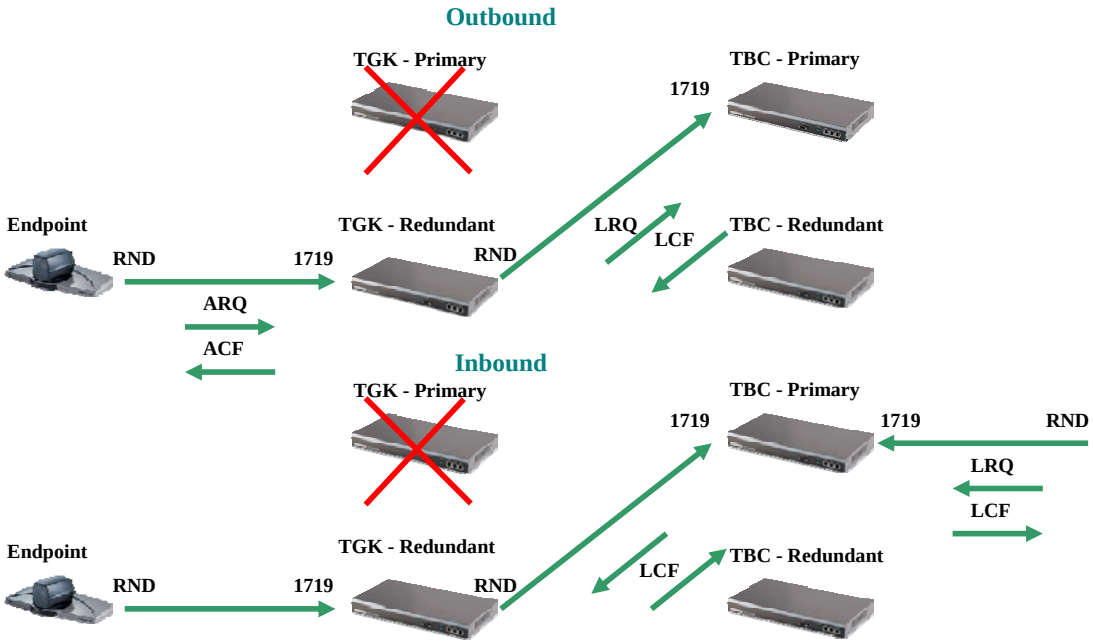
4.4.4.1 Initial deployment

The initial deployment of the alternate solution will show all endpoints registered to the internal gatekeeper as well as all inbound and outbound calls traversing the original traversal link.



4.4.4.2 Loss of communication with primary gatekeeper

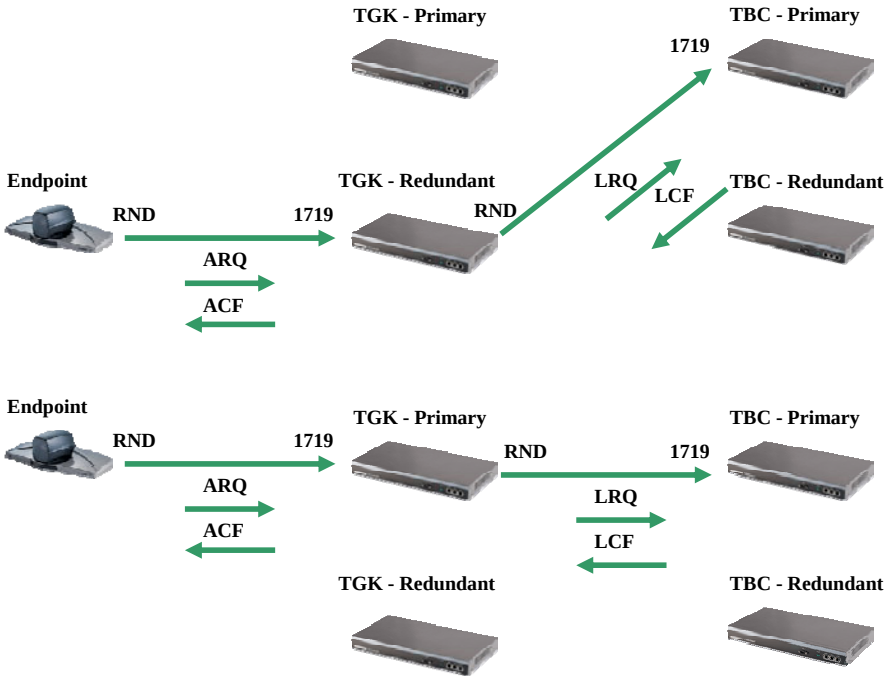
When the primary gatekeeper loses communication, the redundant gatekeeper will assume all responsibilities. All inbound and outbound traversal calls will flow through this gatekeeper to the primary Border Controller.



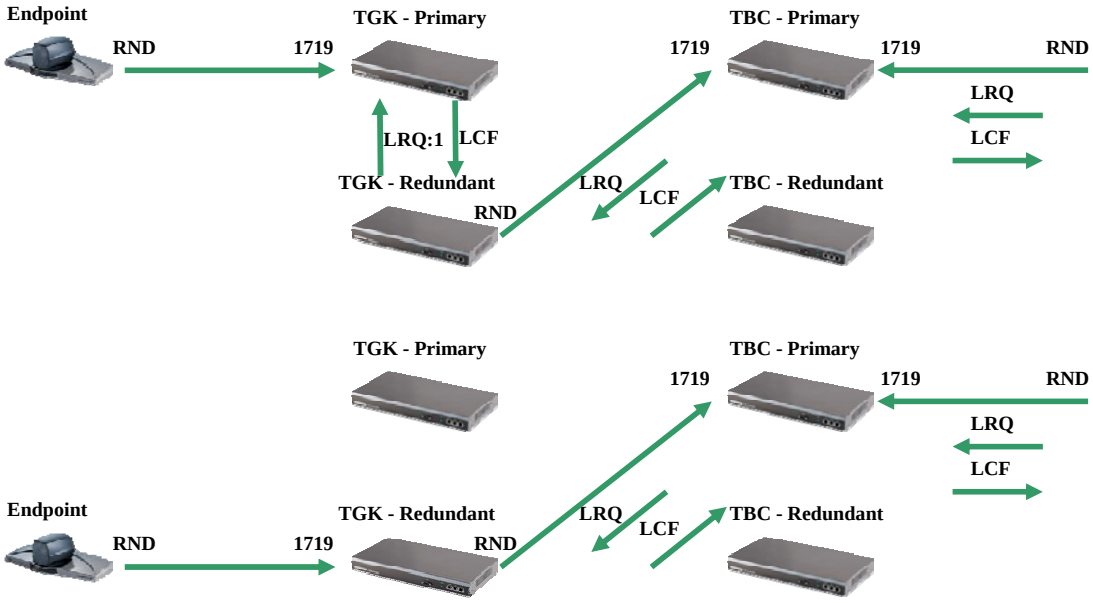
4.4.4.3 Primary gatekeeper communication restored

If the communication is then restored to the primary gatekeeper, all inbound and outbound traversal calls will continue to utilize the traversal link between the primary Border Controller and the redundant gatekeeper. However, at this point, outbound calls may utilize both traversal links, depending upon which gatekeeper to which the endpoints are registered.

Outbound

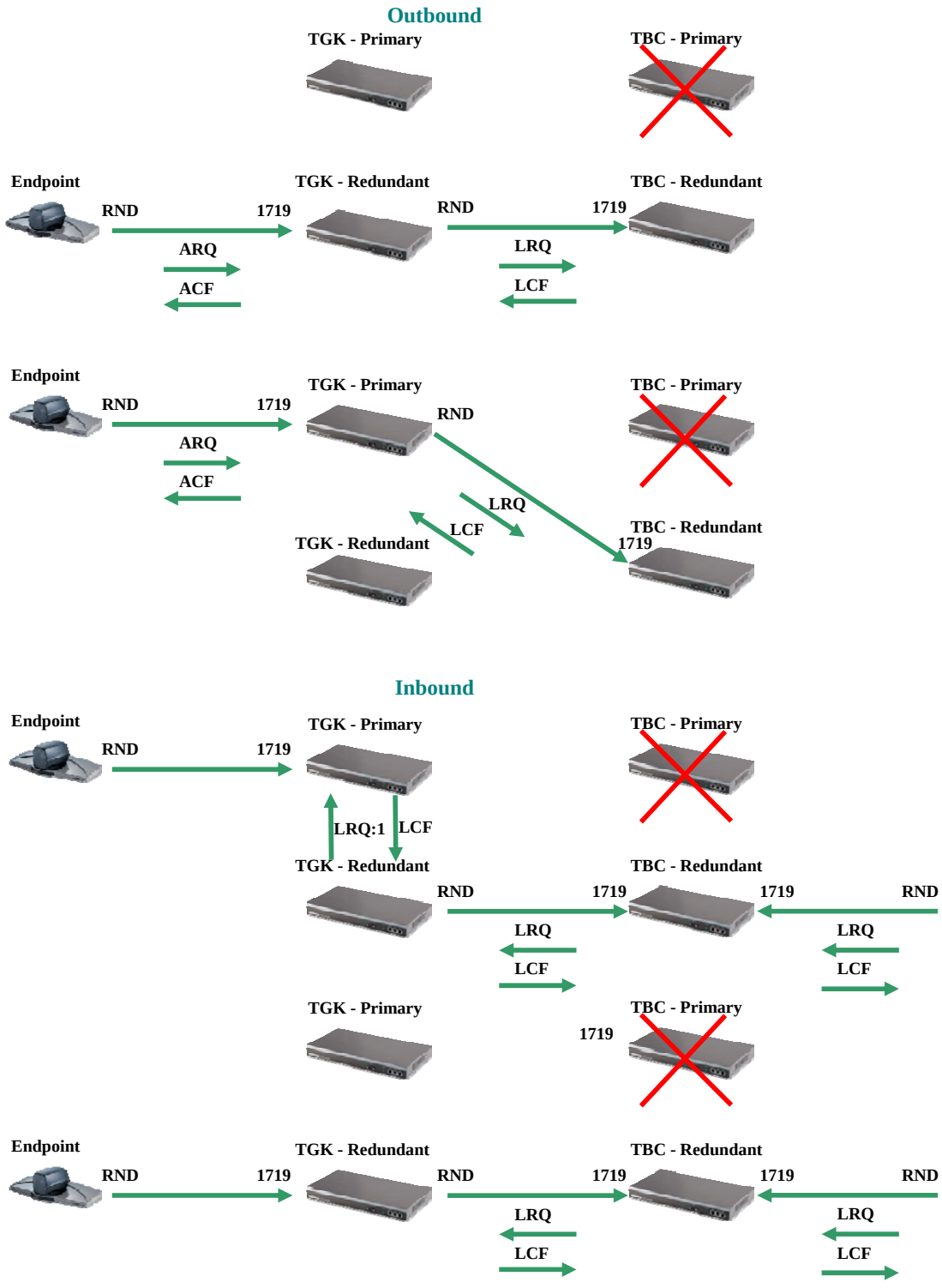


Inbound



4.4.4.4 Communication with primary Border Controller fails

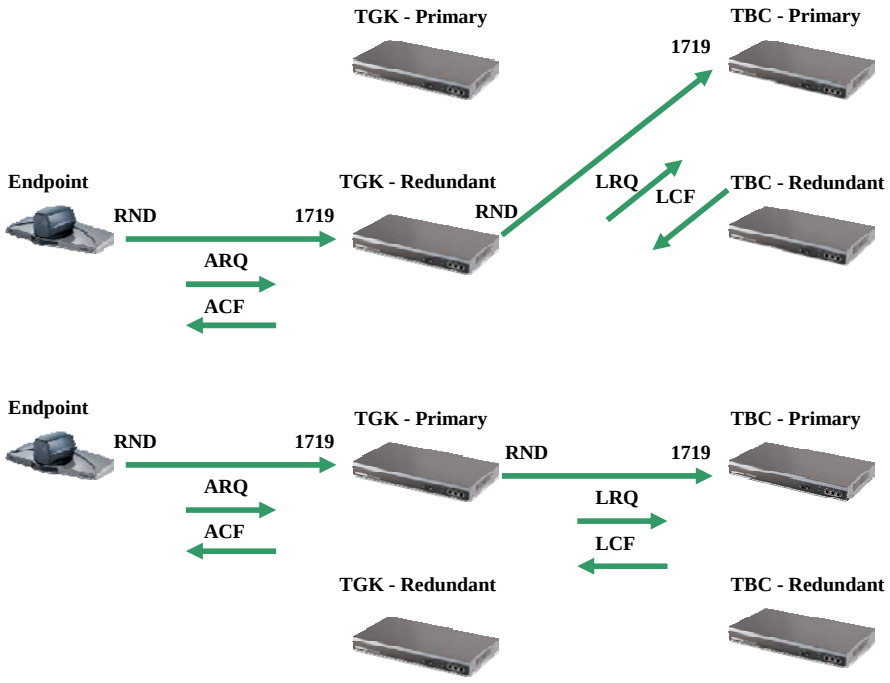
If, at this point, the primary Border Controller were to lose communication, outbound traversal calls would continue to flow through both gatekeepers, however they would now flow outbound through the redundant Border Controller. Inbound traversal calls would continue to flow inbound through the redundant gatekeeper only. The redundant Border Controller would consider the redundant gatekeeper as the primary traversal link since it has been registered for the longest amount of time.



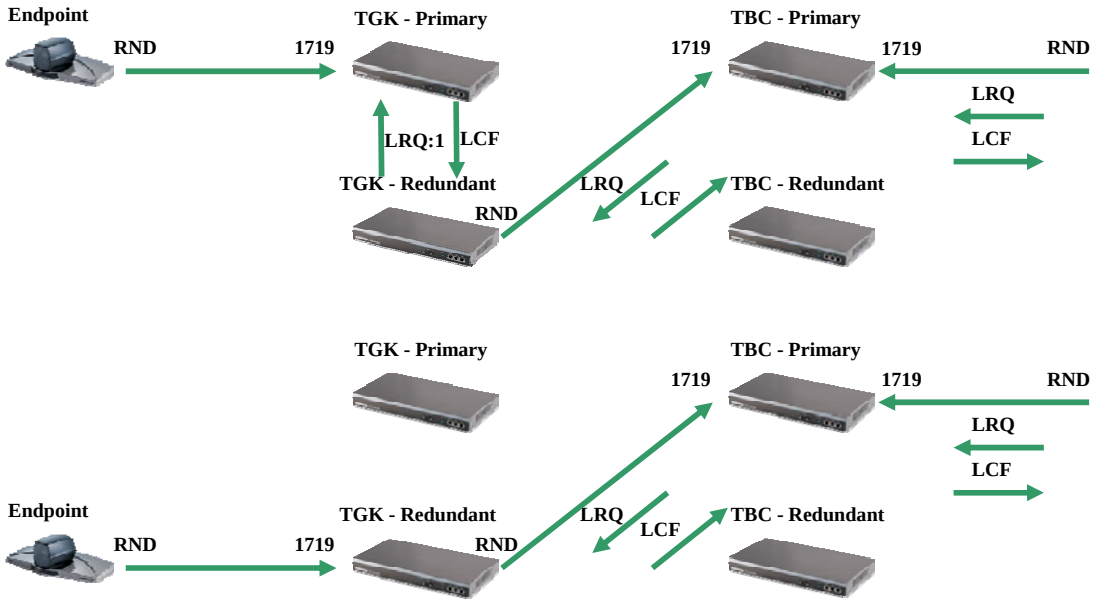
4.4.4.5 Communication with the Primary Border Controller is restored

When communication to the primary Border Controller is restored, it will again assume the responsibility of all inbound and outbound traversal calls, as it did in section 4.4.4.3.

Outbound

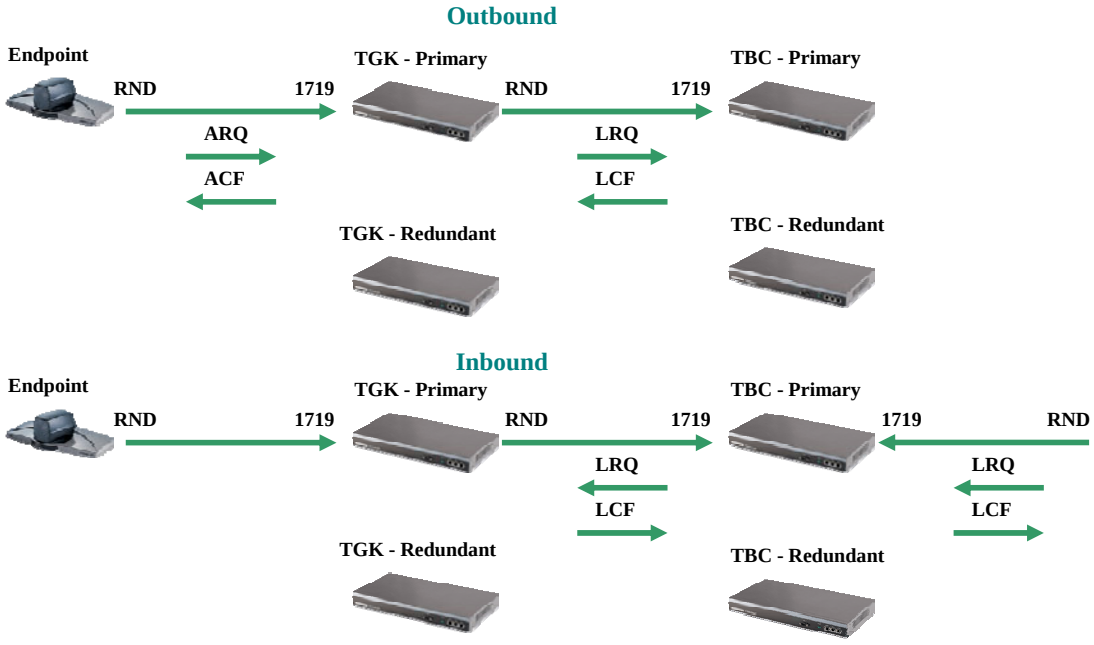


Inbound



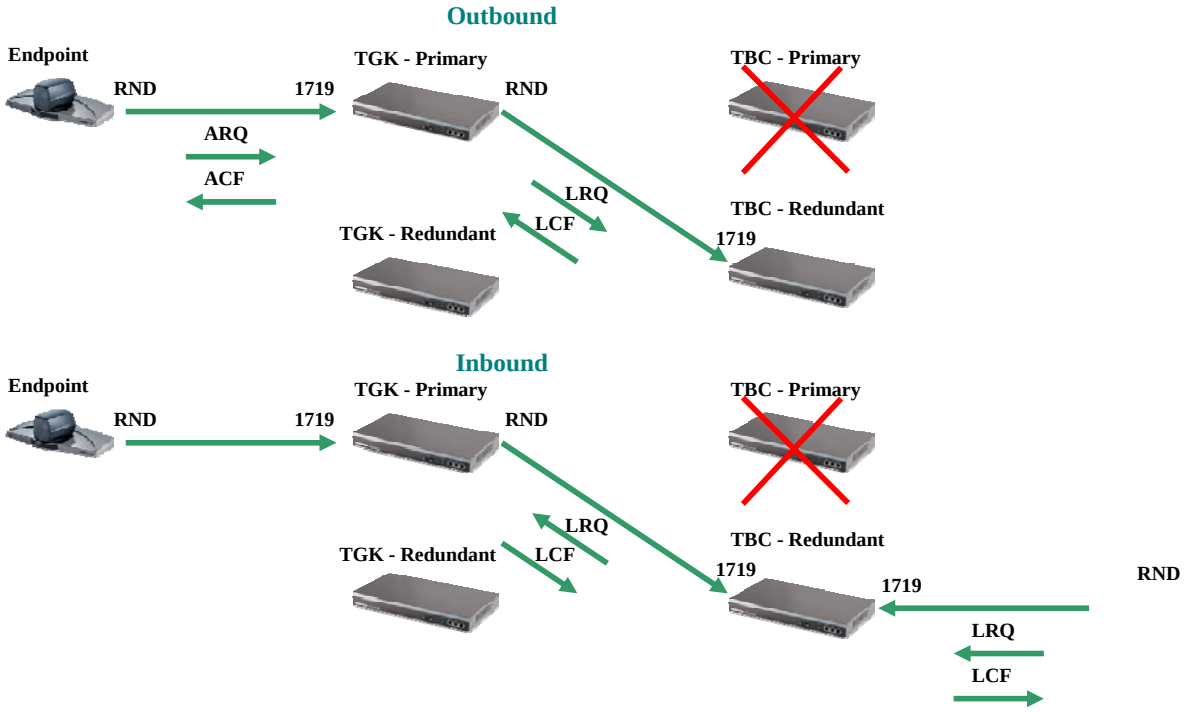
4.4.4.6 Reset Initial Configuration

In order to continue the discussion over how the truly redundant network will react in certain scenarios, it is important to re-establish the baseline network.



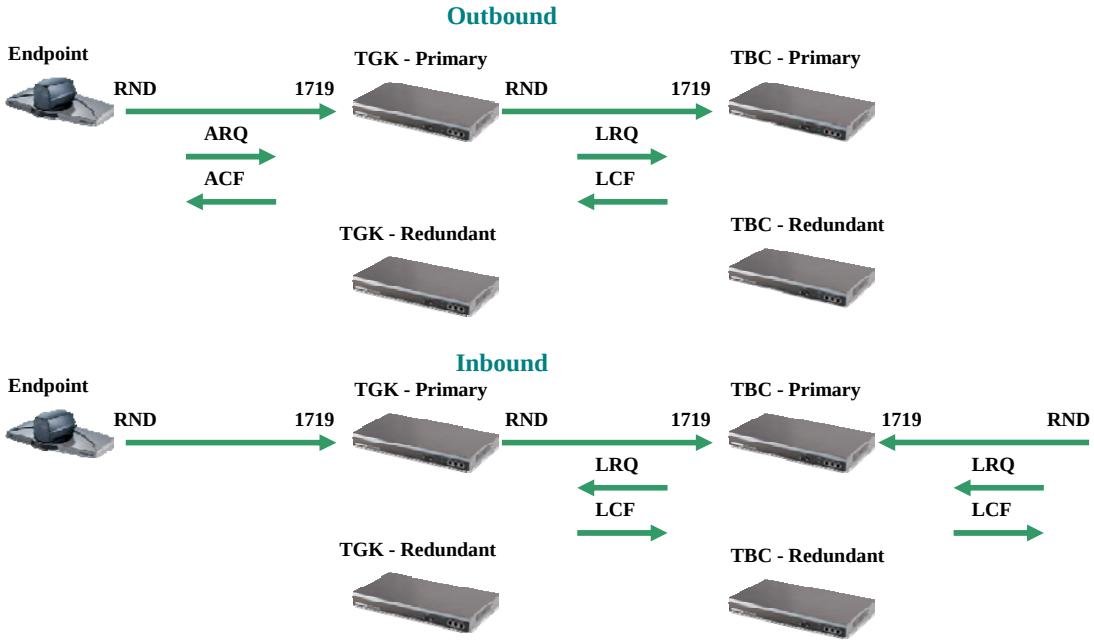
4.4.4.7 Loss of communication with Primary Border Controller

If, after the establishment of the initial configuration, the primary Border Controller looses communication with the network, then all inbound and outbound calls will begin to route through the redundant Border Controller. Since, at this point, we have not had a scenario in which the redundant gatekeepers will come play, all calls will still route through the primary gatekeeper.



4.4.4.8 Communication restored on primary Border Controller

When communication is restored on the Border Controller, it will assume primary responsibilities for traversal calls immediately and, thereby, restore the initial configuration.



5. Supplemental Notes/References

5.1 References/Related Documents

For more information on any of the features discussed within this document, please review:

- TANDBERG Website – http://www.tandberg.net
- TANDBERG Documentation – http://www.tandberg.net/support/documentation.php
- TANDBERG Gatekeeper Documents:
  - D11381 TANDBERG Gatekeeper User Manual
  - D11380 TANDBERG Gatekeeper Installation Sheet
  - D50360 TANDBERG Software Release – Gatekeeper (N3)
  - D50404 TANDBERG Software Release – Gatekeeper (N4)
- TANDBERG Border Controller Documents
  - D13691 TANDBERG Border Controller User Manual
  - D13380 TANDBERG Border Controller Installation Sheet
  - D50361 TANDBERG Software Release – Border Controller (Q2)
  - D50405 TANDBERG Software Release – Border Controller (Q3)
- TANDBERG Infrastructure Documents
  - D50383 XML and TANDBERG CPL
- TANDBERG H.323 Documents:
  - D50305 TANDBERG and H.323