Networking Best Practices for Large Deployments

Google™ Apps

Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
www.google.com

Part number: NETBP_GAPPS_3.5

October 15, 2013

Contents

Chapter 3

Introduction

About This Guide

This document discusses best practices for optimizing your large-scale IP network for Google Apps for Business, Apps for Government, and Apps for Education.

The recommendations and information in this guide have been gathered through our work with a variety of customers and partners in many network environments. We thank our customers and partners for sharing their insight and experience.

Target Audience

This document is intended for Google Apps customers with complex networks, especially those that are spread across a large geographical area. Administrators with smaller networks or networks in a single location may find that some of this information is useful and may find answers to specific questions, but some of the major network routing, capacity, and testing issues might not apply.

Benefits

Optimizing your network configuration will help you to improve your Google Apps implementation in the following ways.

- Improve the responsiveness of Google Apps by reducing latency in your IPv4 network.
- Reduce bandwidth consumption by optimizing network routing and network services.
- Predict network performance and capacity needs by collecting baseline metrics for latency, packet loss, and network availability, before your Google Apps implementation begins.
- Reduce upload and download times with Google Apps for large files, such as internal videos and attachments.
- Efficiently migrate data from existing legacy servers into Google Apps.

Level of Effort

The level of effort needed to implement the recommendations in this guide will depend on your requirements, your current infrastructure, and the skills of your network team. The design principles and implementation best practices in this document are not industry- or technology-specific. The principles in this document do not require specific technical expertise outside of an industry-standard network architecture and network engineering skill set.

Getting The Most Out Of This Guide

This guide includes testing and planning methodology, answers to common questions about the impact of Google Apps on IPv4 networks, and results of field studies on best practices for integrating your network with Google Apps.

This guide is designed to be used in the following ways:

- As a reference guide of network best practices and recommended network tools and methodology. The checklist in the next chapter provides a reference to each network topic, with links to further information. See “Network Action Checklist” on page 9.
- As an in-depth discussion of a variety of network best practices and related topics. You can read this document in its entirety to gain a detailed understanding of all topics related to networks and Google Apps.
- As a reference guide to answer questions on specific topics about network best practices. Use the table of contents or search function to quickly find specific topics of interest.

Life Cycle of Your Google Apps Implementation

The information presented in this guide is associated with multiple milestones of your Google Apps deployment:

1. **Network Evaluation**: This section contains information on evaluating your current network before you deploy Google Apps. While this information may be helpful after you have deployed Google Apps, you will see the best results if you run these evaluations and tests before any other steps.
2. **Network Configuration**: This section contains notes and information on how to set up your network to work best with Google Apps. This section includes network routing information, IPv4 addresses, protocols and port numbers, proxy server configuration, DNS configuration, firewall setup, and mail server setup.
3. **Client Configuration**: This section provides advice on setting up the environment for your users. This includes client information, mobile network expectations, and migration.
4. **Network Monitoring**: This section includes notes on maintenance of your network, and troubleshooting if problems occur after a complete deployment.

Related Documentation

For additional information and background about network best practices and related topics, refer to the following related resources:

- **Google Apps Technical Transition Guide**: Overview of how to transition your organization to Google Apps from another messaging platform.
- **Google Apps Deployment Resources**: A resource center for IT administrators to collect information on deploying Google Apps. Designed for large organizations.
- **Google Apps Partner Connect**: Requires login. Site with useful information for partners, including other related *Notes From The Field*.

Disclaimer for Third-Party Product Configurations

This guide describes how Google Apps products work with common servers and the configurations that Google recommends. These instructions are designed to work with the most common scenarios. Any changes to your configuration should be made at the discretion of your administrators.

Google does not provide technical support for configuring third-party products. In the event of a third-party issue, you should consult your network administrator. GOOGLE ACCEPTS NO RESPONSIBILITY FOR THIRD-PARTY PRODUCTS. You may also contact Google Solutions Providers for consulting services. Links to third-party Web sites are provided for your convenience. The links and their content may change without notice. Please consult the appropriate products' Web sites for the latest configuration and support information.

Chapter 4

Network Action Checklist

About This Checklist

This section contains a summary checklist of all action items in this guide. If you don't have the time to review this guide end-to-end, we suggest you start by reviewing this Network Action Checklist.

Each topic is described in detail later in this guide.

Network Evaluation

Evaluate your current network and plan for capacity needs. To achieve the best results during testing, use the following methodology:

- ☐ Conduct an inventory of all your network locations, including location name, Internet access type (e.g., T1, VPN, DSL), and available Internet bandwidth.
- ☐ Test DNS resolution from all network locations to Google Apps, to ensure that clients in your network can resolve Google Apps hostnames.
- ☐ Test ICMP connectivity from all network locations to Google Apps, to ensure that clients in your network can reach Google servers.
- ☐ Test TCP/UDP reliability from all locations to Google Apps, to ensure that clients in your network can reliably establish and maintain a connection to Google Apps.
- ☐ Assess WAN bandwidth between your Internet egress location and network locations which use that egress point.
- ☐ If you intend for your users to connect to Google Apps through a proxy server, create a test environment and measure how many connections to expect per user, so you can calculate the expected number of outbound connections on your proxy server.

For more information on evaluating your network, see “Test Your Network Environment” on page 13 and “Proxy Server Evaluation and Sizing” on page 16.

Network Configuration

The following recommendations describe network configurations that will help provide your users with the best experience with Google Apps. These recommendations increase network availability and performance, and can reduce costs by simplifying the network equipment required to reach Google Apps.

Network Routing

To achieve the best performance with network connections to Google Apps:

- ☐ Route network traffic to the Internet as close to the end user as possible, in terms of geography and network topology.
- ☐ Focus on addressing latency issues over bandwidth requirements. Above a minimum bandwidth level, bandwidth considerations are generally less significant for Google Apps.
- ☐ Open your firewalls to the ports that Google Apps services use. For details, see “Google Protocols” on page 21.
- ☐ Avoid using specific IPv4 addresses to permit access to Google Apps. See “Google IPv4 Addresses” on page 19.
- ☐ Consider traffic prioritization if you are using a hub-and-spoke network topology or if your network has multiple locations with a single network egress point.

For more information about network routing, see “Network Addressing and Protocols” on page 19 and “Network Routing” on page 22.

Proxy Servers

- ☐ Avoid routing Google Apps data through a proxy that inspects the content of HTTP traffic, because this will reduce performance, and a great deal of Google Apps content is dynamic or encrypted.
- ☐ Keep your proxy servers in a location that is close to your users and their Internet egress point, in terms of both geography and network topology.
- ☐ If you need to filter web traffic by URI, consider using a PAC configuration file on the client’s desktop, since URIs in encrypted HTTP traffic are not visible to the proxy.
- ☐ If you are using a proxy server that supports SSL Terminations, you can set up your proxy server to inspect Google Apps content while relaying the secure connection.

For more information about setting up proxy servers, see “Proxy Servers” on page 24.

Other Services

- ☐ Use a DNS resolver in a location that is close to the user, in terms of both geography and network topology. Using DNS resolvers located in remote network locations will greatly slow down connections to Google Apps.

- [ ] If it's not feasible to use a DNS resolver that's close to the user, use a DNS server that supports the edns-client-subnet extension (**Draft Proposal 2671**)—such as **Google's DNS server** or **OpenDNS**—which allows the resolver to pass part of the client's IP address.
- [ ] Adhere to the advertised TTL value for all DNS record types.
- [ ] Set up firewall rules to allow unrestricted outbound HTTPS traffic to Google Apps. You do not need to set up special rules for inbound traffic; Google Apps does not generally initiate inbound traffic to users.
- [ ] Avoid routing inbound and outbound mail through a gateway inside your network. If inbound and outbound mail is routed to a gateway inside your network, mail traffic will consume unnecessary network resources.

For more information on network services, see "Other Network Services" on page 28.

Client Configuration

After your network is configured, prepare your user environment to work with Google Apps. This can include setting up clients, SSO authentication, and user data migration.

Client Access

When planning for clients that will connect to Google Apps, consider the following:

- [ ] Suggested browsers include Google Chrome, Mozilla Firefox, Microsoft Internet Explorer, or Apple Safari for your Google Apps users. Install and enable one of these browsers if your users currently have legacy browsers. Modern browsers provide a better user experience by improving the speed in which web pages are rendered.
- [ ] Consider the use of Android or ActiveSync mobile devices instead of BlackBerry devices. BlackBerry Enterprise Services can consume resources on your network.

For more information on setting up client environments, see "Client Access" on page 33.

Authentication

If you plan to set up Single Sign-On (SSO) authentication, consider the following:

- [ ] Set up SSO servers in distributed network locations, rather than a central location.
- [ ] Implement your SSO server together with your VPN servers, to avoid routing authentication traffic of VPN connected users to a different location.
- [ ] Set up internal DNS servers to redirect SSO traffic to the nearest SSO server, and ensure that alternate SSO servers are in place for redundant service in case of disruptions that prevent users from accessing the SSO server in a particular location.

For more information on SSO Authentication, see "Authentication" on page 36.

Migration

Google Apps deployments can involve migration traffic, either from local clients like **Google Apps Migration for Microsoft Outlook,** or from server-side clients like **Google Apps Migration for Lotus Notes** and **Google Apps Migration for Microsoft Exchange**.

Migration of legacy data into Google Apps is typically a resource-intensive activity, but usually only happens once. If you plan to migrate user data, consider the following:

- ☐ Ensure that your migration servers are in the same location as your legacy data servers, or at least that the connectivity between servers has low latency and high bandwidth.
- ☐ Avoid routing traffic from the migration servers to Google through proxy servers, to reduce migration performance and to avoid unnecessary proxy server load.
- ☐ Assess your network capacity before migration to determine the maximum amount of data that you can migrate concurrently. Adjust your migration plan accordingly.
- ☐ During migration, some of the connections established to Google servers can stay open for a period of time depending on the migration tool. To avoid any possible migration errors, and to reduce the need to remigrate data, it is important to keep these sessions open and not close them prematurely with any proxy or firewall timeouts.

For more information about data migration, see “Migration” on page 38.

Network Monitoring

Use monitoring tools to maintain and administer an existing IPv4 network that is already working with Google Apps.

- ☐ There are a variety of network monitoring tools available that are well-suited to monitor Google Apps traffic. For a list of recommended network monitoring tools, see “Monitoring Tools” on page 41.
- ☐ Network Packet Captures can assist in identifying possible performance issues during your troubleshooting or when working with your network provider or Google support. For more information, see “Network Packet Captures” on page 42.

Chapter 5

Network Evaluation

Summary

When planning for the implementation, you will achieve better results if you first understand your current network capacity and the expected amount of network load from Google Apps. The best way to predict how much load to expect is to benchmark your bandwidth usage in your network, and create a test environment to simulate how much capacity each user will require. This section discusses approaches to test your network environment in detail.

If you have already deployed Google Apps without running environment tests and benchmarking, it may still be valuable to do so, since this may give a benchmark for future planning and capacity requirements, and may help to identify potential issues before they affect your users' experience.

Test Your Network Environment

To see the best results, test your network environment prior to deploying Google Apps to discover any possible issues. Network testing prior to a Google Apps implementation is primarily focused on assessing capacity and performance of network bottlenecks, Internet proxies, firewalls, and any other network components responsible for routing or monitoring Internet-based traffic.

Below are the recommended steps to assess and test your network prior to deployment.

- Conduct an inventory of all of your network locations, including location name, Internet access type (e.g., T1, VPN, DSL), and available Internet bandwidth.
- Test DNS resolution from all network locations to Google Apps, to ensure that clients in your network can resolve Google Apps hostnames.
- Test ICMP connectivity from all network locations to Google Apps, to ensure that clients in your network can reach Google servers.
- Test TCP/UDP reliability from all locations to Google Apps, to ensure that clients in your network can reliably establish and maintain a connection to Google Apps.
- Assess WAN bandwidth between your Internet egress location and network locations which use that egress point.

Inventory of Network Locations

When planning for a Google Apps implementation, it is important to create an inventory of all locations from which users will access Google Apps. The goal of this inventory is to gather information about the Internet connectivity and capacity from each network location.

When conducting an inventory, include the following information about each network location:

- The location's name and a description of its Internet access. Example: "Headquarters, DS3."
- Internet bandwidth average and peak usage. Example: "50% average usage, 70% peak usage."
- Number of proxy servers, and current average and peak usage.
- Number of firewall appliances, and current average and peak usage.
- Number of DNS servers, and current average and peak usage.

Once you have collected this information for every network location, use the data to assess current capacity, and whether any upgrades are needed.

Network Testing

Use the information you have gathered during your network inventory to test each network route, DNS server, and proxy server. Run the following tests for all relevant network connections in each location.

Note: The third-party testing software described in this section is available for various operating systems, including Linux, Unix, Mac OS X, and Windows.

DNS Resolution Test

Ensure that clients in your network can resolve Google Apps hostnames and URIs by testing DNS Resolution from all of your network locations to Google Apps hostnames, as follows:

1. Open the sample list **text file hosted on the apps-deployments code site**. (Note that this is a sample only, not a complete list.)
2. Save the sample .txt file in the directory where you will be using the test commands:
   a. Click **View raw file**.
   b. Right-click the page, and choose **Save As**.
3. Run the following command to test DNS resolution:

```
% dig +all +trace -f GoogleAppsDomains.txt
```

ICMP Connectivity Test

Ensure that clients in your network can reach the hostname `mail.google.com`, by testing ICMP connectivity from all your network locations to Google Apps. Test that your users can reach Google Apps, especially from all users' VLANs.

```
% ping -s 512 -c 400  -n mail.google.com
```

If you see slow or failed connections on your ping requests, this may indicate a loss of connectivity. Investigate each step of your connection to identify the source of the problem.

TCP/UDP Reliability Test

Ensure that clients in your network can reliably establish and maintain a connection to Google Apps servers. Use the `Hping` tool to test link reliability over a period of time. Run the following command:

```
% time hping3 -S mail.google.com  -p 443 --fast -c 1000
```

If there is time, run this test for each domain listed in the `GoogleAppsDomains.txt` file mentioned above.

Note: TCP/UDP reliability tests are intrusive and can affect network performance. Run these tests during off-hours to gather data while causing the minimum possible the impact to your network.

Available WAN Bandwidth Assessment

Use the `iperf` tool to assess the amount of bandwidth available from each location to its network egress point. This test is run on both the client and the network egress point.

This test is intended to assess bandwidth within your WAN network. It is not suited for testing bandwidth between your network and Google Apps servers.

On each remote location that is connected over a WAN network, run this command:

```
% iperf -c CLIENT IP ADDRESS -d
```

On the network egress location, run this command:

```
% iperf -s
```

Note: WAN Bandwidth tests are intrusive and can affect network performance. Run these tests during off-hours to gather data while causing the minimum possible the impact to your network. If you need to run these tests during business hours, be careful of possible effects this test may have on your network performance.

Network Testing Tools

You can obtain the tools discussed above from the following online sources:

- Download the `Hping` packet analyzer tool from **hping.org**.
- Download the `iperf` bandwidth performance measuring tool from **SourceForge**.

Proxy Server Evaluation and Sizing

In a cloud computing environment, there are typically more outbound requests for external hosts than are generated in a legacy environment. The increase in outbound requests may impact the number of proxy servers required in your network.

If you intend for your users to connect to Google Apps through a proxy server, you can determine what level of proxy server load to expect by running these tests beforehand. Use this information to estimate whether you need to increase your proxy server capacity.

Follow these steps to evaluate your proxy server needs:

1. Create a test environment with each platform and browser that you plan to use in your user environment.
2. For each browser, measure the number of connections that occur during testing, including minimum and maximum concurrent connections, both for idle use and active use.
3. Based on this information and the number of users you expect on your system, calculate expected number of connections for your proxy server.
4. Use these calculations to plan for any proxy server capacity changes needed.

See below for more information on these steps.

Benchmark Proxy Load Per User

To benchmark the amount of proxy resources used by a typical user, establish a testing environment where you can test the various platforms and browsers that you support. Your testing environment should include testing machines on your network that can connect to Google Apps using the same routes that you plan to use for your users. (For example, if you plan to deploy users with Microsoft Windows 7, with Chrome 10 and Firefox 3.6 as your standard OS and browsers, use this same environment when running benchmark tests.) Once your testing environment is ready, direct traffic to a test proxy where you can measure the number of connections.

Collect the following data for each environment, while using Google Apps services available in your domain. For instance, open Gmail, Google Talk, Google Docs, and Google Calendar.

- Average connections/sec
- Peak connections/sec
- Non-peak connections/sec

Additionally, Google Apps, like many web-based applications that run in the cloud, keeps several connections open to the remote server to poll for new data. To evaluate the load caused by these open connections, measure the following in your test environment.

- Minimum amount of connections an idle user has with your browser platform
- Maximum amount of connections an idle user has with your browser platform

Once you have gathered these numbers, you can compile this information to estimate the load you might experience given your unique environment.

Estimate Expected Proxy Resources Needed

To estimate the amount of load you can expect during a Google Apps rollout, multiply the number of connections for each test environment by the number of users you expect for that environment.

Use the following calculations.

```
Estimated average load = Sum (average load of each test machine environment X
estimated number of users who will use that environment)

Estimated peak load = Sum (peak load of each test machine environment X
estimated number of users who will use that environment)

Estimated idle load = Sum (idle load of each test machine environment X
estimated number of users who will use that environment)
```

If the estimated average load, plus any additional traffic your proxies handle, exceeds your current capacity, make plans to expand your proxy server capacity, or change your proxy server implementation so that your proxy servers do not handle the requests that your users will make to Google Apps.

Example

In the following example, a large enterprise plans to deploy the following:

- 5000 users running Chrome on Windows 7.
- 3000 users running Firefox on Windows 7.

During benchmarking, tests show the following sample numbers of concurrent connections through the proxy server. (Note: These are for example only. Your environment will vary.)

- **Chrome on Windows 7**

  Connections when entering URI: 1
  Connections during initial load: 3
  Connections during login: 6
  Connections after a few minutes idle: 4
  Connections when opening Calendar and Docs: 4
  Connections when loading a document: 6

  Average load: 3.6 connections
  Peak load: 6 connections
  Idle load: 3.1 connections

- **Firefox on Windows 7**

  Connections when entering URI: 1
  Connections during initial load: 4
  Connections during login: 9
  Connections after a few minutes idle: 3
  Connections when opening Calendar and Docs: 11
  Connections when loading a document: 17

  Average load: 4.1 connections
  Peak load: 17 connections
  Idle load: 3.8 connections

Based on this, the expected load is:

- **Average**: (5000 x 3.6) + (3000 x 4.1) = 30,300 connections.
- **Peak**: (5000 x 6) + (3000 x 17) = 81,000 connections.
- **Idle**: (5000 x 3.1) + (3000 x 3.8) = 26,900 connections.

Based on this estimate, the proxy environment needs to be able to support at least 30,000 connections, possibly more to avoid problems during peak periods, or if growth is expected. If the current proxy server environment is running at 50% capacity with 20,000 connections, this is a sign that it will be necessary to deploy significantly more proxy servers, or to route Google Apps traffic so that it bypasses the proxy server.

Chapter 6

Network Configuration

Summary

This section includes details on how to optimize your network for Google Apps. This includes information on Google's IPv4 addresses, protocols used, routing suggestions, proxy server configuration options, and DNS configuration. Use this information as a guide when configuring your network, and as a reference for what types of requests Google Apps clients will make to Google servers.

Network Addressing and Protocols

Google IPv4 Addresses

Google Apps exists in a multi-tenant server environment that includes both Google Apps and consumer accounts. Therefore, Google Apps shares the same IPv4 address space as Google consumer services. For example, Google Docs servers could use the same IPv4 address space as Picasa Web. In addition, a specific IPv4 address for a Google hostname, such as `mail.google.com` or `docs. google.com`, might be serving *both* Google Apps and consumer users at the same time.

For any Google hostname, such as `mail.google.com` or `docs.google.com`, the IPv4 address is not static and is valid only for its time-to-live (TTL) value returned in the DNS lookup of the hostname.

For example, if we query the A record for `mail.google.com`, several results are returned:

```
% dig a mail.google.com +ttl

;; ANSWER SECTION:
mail.google.com.        68665   IN      CNAME   googlemail.l.google.com.
googlemail.l.google.com. 152    IN      A       74.125.225.86
googlemail.l.google.com. 152    IN      A       74.125.225.87
googlemail.l.google.com. 152    IN      A       74.125.225.85
```

The second column in the result set is the TTL for the records in seconds. Based on these sample results, we can determine that the IPv4 addresses are valid for only about 2.5 minutes.

Google IPv4 addresses for specific hostnames are not static. For example, do not assume mail.google.com will always be 74.125.225.245. If you need to configure your environment to accept mail from Google for a mail gateway, include all of the subnets from the _spf.google.com record per this **Administrator Help Center** article.

It is not recommended to use Google's IPv4 address space to permit access to Google (see "Google Global Cache"); however, IPv4 addresses can be used to implement traffic redirection and prioritization to the Internet knowing the implications of Google Global Cache (a recommendation stated throughout this document). A more robust option to implement these prioritizations can be Google's hostnames (see "Google Host Names").

Google Host Names

Google owns and operates a large amount of domains to serve the various services, products, partnerships, and ventures in use. To efficiently serve and operate such a large, global Internet presence requires advanced network engineering and optimizations. Therefore, any system that uses Google's hostnames should not be used as a means to allow access. Google providing a static list of hostnames for use in customer networking configuration is impractical. Rather, hostnames should be used to implement traffic redirection or prioritization to the Internet; a recommendation stated throughout this document.

For a sample list of wildcarded hostnames, see http://code.google.com/p/enterprise-deployments/source/browse/trunk/apps/utils/GoogleAppsWildcardedDomains.txt.

Note: This list comes with no expressed warranty as to its accuracy at any given point in time. It is merely a starting point for a network administrator and should be maintained by the Enterprise going forward through log analysis.

Google Global Cache

Many of Google's services and applications participate in the Google Global Cache (GGC) content delivery system. The goal of this system is to provide the best service to all users by deploying the knowledge and learnings from our Network Engineering teams.

The GGC system involves Network Operators and Internet Service Providers in the distribution of commonly accessed resources. The participants in GGC have deployed a number of Google owned and operated servers inside their network to serve popular Google content. This results in IPv4 addresses being used with Google services and applications that are owned by these host operators. Therefore, any use of Google's IPv4 addresses to allow access should not be used. Rather, IPv4 addresses may be used to implement traffic redirection or prioritization knowing that there may be some Google related traffic going to IPv4 addresses not listed.

Google's use of GGC for content delivery is most effective for users with a large "network-distance" from Google (see Google's data center locations). Google's use of GGC is dynamic in both the services and client networks it applies to. Refer to the frequently asked questions at peering.google.com for more information related to GGC and its use.

Google Protocols

Google Apps services are mostly web-based or API-based. The table below shows common Google Apps services, and the protocol used for each. As shown in the table, the datagrams that Google Apps uses are almost always TCP-based, except as noted.

Application	Protocol	Port
Mail, Calendar, Docs, Sites	TCP	443
Google Apps Sync for Microsoft Office	TCP	443
Google Apps Connector for BlackBerry Enterprise Server	TCP	443
Google Talk (Web)	TCP	443
Google Talk (Desktop Client)	TCP (XMPP)	5222 or 443
Google Talk (Voice and Video)	Special; see below	Special; see below
Google+ Hangouts	Special; see below	Special; see below
Google Apps Migration for Microsoft Exchange	TCP (API)	443
Google Apps Migration for Lotus Notes	TCP (API)	443

Google Talk Voice and Video and Google+ Hangouts

Google Talk Voice and Video

During a voice or video call, the Google Talk Voice and Video plug-in attempts to establish a connection from the caller to the callee (for details, **see Google Talk Developer Documentation**). The plug-in attempts different protocols and transport methods, depending on the network connection at the time of the call.

Google Talk Voice and Video attempts to make a connection using any of the following protocols and connection endpoints, in the following order of preference

1. Direct UDP connection between caller and callee, using STUN, on random ports.
2. UDP connection between caller and callee with NAT-table traversal, using STUN, on random ports.
3. UDP connection between caller and callee through a Google Relay server, using STUN, on random ports. (This connection works for symmetric NAT.)
4. Direct TCP connection between caller and callee.
5. TCP connection between caller and callee through a Google Relay server.
6. SSL over TCP connection between caller and callee through a Google Relay server, on port 443.

Google+ Hangouts

Google+ Hangouts attempts to establish a connection between a participant and a Google server using a method similar to that of Google Talk Voice and Video. For details, see the **Administrator Help Center**.

Firewall Configurations

To provide users with the full capabilities (voice, video, and text) of Google Talk Voice and Video and Google+ Hangouts, allow UDP out from clients on your network. See the Help Center article "**Optimize your Network for Hangouts**" for more detail.

If you don't want to allow UDP out from clients on your network, at a minimum, permit TCP out from clients on your network to Google on ports 80 and 443. (See the Help Center article "**Optimize your Network for Hangouts**" for more detail.) Remember, though, that forcing a TCP connection for services such as voice and video may create a poor experience for your users; therefore, we recommend allowing the use of UDP out from your network.

Google Talk negotiates and establishes voice and video calls using the open source `libjingle` library. For more information, see the libjingle **Google Code Project Page**. A diagram of network behavior for the libjingle library is available in the **Libjingle Network Diagram**.

Network Routing

When routing to Google Apps, the simplest network routing generally provides the best performance. Reduce complexity and unnecessary network routing from users' locations to Google data centers. A goal for your network design should be to reduce the total round trip time from your network to Google. If you see performance issues, address any latency problems before you increase bandwidth.

To achieve the best performance with connections to Google Apps:

- Route network traffic to the Internet as close to the end user as possible, in terms of geography and network topology.
- Focus on addressing latency issues over bandwidth requirements. Above a minimum bandwidth level, bandwidth considerations are generally less significant for Google Apps.
- Open your firewalls to the ports that Google Apps services use.
- Consider traffic prioritization if you are using a hub-and-spoke network topology or if your network has multiple locations with a single network egress point.

WAN Optimization

When planning your network cloud strategy, try to reduce latency and round-trip time. Users in remote offices will experience reduced performance if WAN traffic must traverse large geographical areas to reach the Internet. Implement network egress points as geographically close as possible to the user, since traffic across your WAN causes more congestion on some of your more bit-expensive links. Parts of this optimization can be accomplished through DNS resolution changes. For more information, see "DNS Resolution" on page 29.

Traffic Prioritization

You may be able to improve Google Apps performance with traffic prioritization, by giving Google Apps traffic priority over other network traffic to reduce network latency during congestion. Traffic prioritization is possible on the data link layer and the network layer; see the sections below for more information.

You may wish to consider traffic prioritization to reduce potential latency if you have any of the following environments:

- Hub and spoke network topologies.
- Multiple locations with a single network egress point.

Network Layer (Layer 3) Prioritization

Google Apps uses the same set of IPv4 addresses that other Google products use, including consumer products like Gmail and Picasa. It is not possible to distinguish traffic to different products.

If you require network-layer prioritization, we suggest you do one or more of the following:

- Create a proxy PAC file that directs all Google Apps URIs to a proxy that routes only Google Apps traffic. For more information, see “Proxy PAC file configuration” on page 25.
- Configure your networking equipment to prioritize your proxy network interface.
- Distribute proxies to avoid the creation of a hub and spoke proxy topology.

For information on the Google IPv4 addresses and TCP Port usage, see “Google IPv4 Addresses” on page 19.

Peering

Peering is the direct interconnection of your network to Google's network. This reduces latency and improves the reliability of the connection between your network and Google.

For most Apps customers, the best way to do this is to choose an ISP or network provider that already peers with Google. Google peers with many Internet Service Providers in many locations across the globe. This is the easiest and fastest way to realize the benefits of peering closely to Google. Contact your ISP to find out if they have peering established with Google.

For larger corporate networks, it may be possible to peer with Google directly. There are a number of requirements to peer with Google. In general, if you are not peering with other networks already, then it is more appropriate to let your upstream network provider handle peering relationships.

For Google's peering requirements, which apply to ISPs, network operators, and corporate networks, see the Google entry on **PeeringDB**. PeeringDB also contains the list of Internet Exchanges and other locations where Google is capable of peering.

If you or your Internet Service Provider qualifies for peering based on Google's peering requirements, discuss a peering relationship with your Google Technical Account Manager, Deployment Team Member, or Google Enterprise Support representative.

Network Routing Tools

- A variety of useful tools are available to generate detailed data regarding your Internet connection performance on the external website **Measurement Lab**. You can use these tools to measure your overall Internet access performance.
- The site **PeeringDB** is a large worldwide public database with information about peering networks.

Proxy Servers

When planning your proxy server setup for Google Apps users, keep in mind the following best practices:

- Avoid routing Google Apps data through a proxy that inspects the content of HTTP traffic, since this will reduce performance, and a great deal of Google Apps content is dynamic or encrypted.
- Keep your proxy servers in a location that is close to your users and their Internet egress point, in terms of both geography and network topology.
- If you need to filter web traffic by URI, consider using a PAC configuration file on the client's desktop, since URIs in encrypted HTTP traffic are not visible to the proxy.
- If you are using a proxy server that supports SSL Terminations, you can set up your proxy server to inspect Google Apps content while relaying the secure connection.

Proxy Server Configuration

We recommend that you do not route Google Apps traffic through a proxy server. If you decide to send Google Apps traffic through your proxy, look for settings on your proxy server that might disrupt Google Apps traffic.

Look for configurations and settings that include the following conditions:

- Content filters that might mark Google-related traffic as prohibited
- Settings that can lower the total amount of possible concurrent connections/sec per client
- Exceptionally long or short SSL time-outs (The default setting is recommended)
- Outdated firmware versions
- SSL Inspection without hardware acceleration

If you choose to use a proxy server in conjunction with Google Apps, keep your proxy server as close to the client as possible, in terms of both geography and network topology. Your users will have a better experience if you minimize both the number of network hops and the round trip time between your users, the proxy server, and the Internet.

Content Inspection

Avoid content inspection on your proxy server. When Google Apps is configured to run over HTTPS, which is common and recommended, proxy servers cannot inspect content or restrict access without a special proxy configuration.

Filtering Google Apps traffic through a Proxy

The vast majority of traffic originating from your users to Google Apps servers consists of HTTPS transactions. This type of traffic is preferred because it is secure and reliable. Although interruption of traffic to Google Apps for filtering is possible, it can decrease security and reduce the overall experience for your users.

Keep the following considerations in mind when planning to filter HTTPS traffic to Google Apps.

- In browsers and protocols that support the Server Name Identifier (SNI) extension to TLS, you will see the request for the hostname in the initial HELLO from the client in your proxy logs. A list of those browsers is available on the following page in **Wikipedia**. Consult your browser documentation to learn about SNI support.
- In older browsers and SSL versions that do not support the Server Name Identifier (SNI) extension to TLS, you will not see the request for the hostname in the initial HELLO from the client in your proxy logs. In this case, your users usually see a certificate mismatch error because of the virtual hosting nature of Google Apps and similar web services. Be sure to use a browser that supports the Server Name Identifier extension for TLS.

After the initial HELLO request between the client/server and once the TLS connection is established, all traffic is encrypted including the URI path after the hostname.

If you need to filter your users' traffic, there are two recommended ways to accomplish this:

- Filter your users traffic with a proxy PAC file at the browser level prior to encryption is easier and less costly to implement. See "Proxy PAC file configuration" on page 25.
- Perform SSL interception and inspection after the encryption is more secure but is more difficult and costly to implement. See "SSL Inspection" on page 26.

Proxy PAC file configuration

A Proxy PAC file is a cost-effective way to filter traffic because URI and IPv4 evaluation is performed on the client machine prior to encryption.

A proxy PAC file is a set of JavaScript commands that the browser uses to evaluate against the URI requests received from the user.

The following sample script includes code to test if a URI matches the format `https://*.google.com/*`.

```
// If the URI matches https://*.google.com/* then route traffic
// directly to the Internet.

if (isPlainHostName(host) ||
shExpMatch(url,"https://*.google.com/*") ||
```

```
return "DIRECT";


// All other URI requests should be routed through the proxy.

else
return "PROXY corporateproxy.domain.com:8080";
```

More examples for developing a proxy PAC file can be found on the external website **FindProxyForURL**.

Proxy PAC file testing

Implementing a functional proxy PAC file requires careful testing. Use a PAC file testing tool like `pactester` to test different JavaScript functions. A PAC file tester will allow you to pass a hostname and URI and see which path the browser will take given your PAC file. Download `pactester` from the **Google Code pactest project site**.

SSL Inspection

Avoid SSL inspection if possible. SSL inspection is effectively an SSL “man in the middle attack” on your own users to examine the contents of HTTPS traffic. With SSL terminations, your users connect to a proxy as an end point. The proxy then terminates the SSL connection and inspects traffic, then establishes new a connection to the destination server forwarding the traffic. This can cause a significant increase of load on traditional proxies that perform these operations in software, rather than a network appliance.

There are many commercial appliance vendors as well as many software proxy servers that can perform SSL inspection. Typically this requires additional proxy configuration.

Each proxy server SSL Inspection setup is different, but the typical steps are as follows:

1. Self-sign an SSL Certificate with an internal hostname, such as `mail.example.com`.
2. Install the `mail.example.com` certificate on the proxy server.
3. Write custom proxy rules. For instance, rewrite connections from `https://mail.example.com/` to `https://mail.google.com/a/example.com/`.
4. Reject connections with a Host header that contains `mail.google.com`.

Note: Some proxies will allow you to keep the hostname the same, and use a built-in certificate. This requires that the user’s browser trust the certificate, or users will receive a certificate error. For information on how to resolve these problems related to SSL inspection, consult your proxy server vendor and documentation.

Blocking Access to Google Consumer Services

As an administrator, you might want to prevent users on your network from signing in to a Google service using a consumer account instead of the Google Apps account you provided them with. For example, you may not want them to use their personal Gmail accounts. In addition, you might also want to prevent users from signing in to a Google Apps account from *another* domain.

A common means of blocking access to web services is using a web proxy server to filter traffic directed at particular URIs or hostnames. This approach is ineffective in this case because all the URIs accessed between consumer and Google Apps accounts are the same.

To only allow users to access Google services using specific Google accounts from your domain, you need the web proxy server to add an HTTP header to all traffic directed to `*google.com`; the header identifies the domains whose users can access Google services. Since most Google Apps traffic is encrypted, your proxy server also needs to support SSL interception. (See **this article** for a list of proxy servers known to support both SSL interception and HTTP header insertion.)

To prevent users from signing in to Google services using Google accounts other than those you explicitly specify:

1. Route all traffic outbound to `google.com` through your web proxy server(s).
2. Enable SSL interception on the proxy server.

   Since you will be intercepting SSL requests, you will probably want to manage client certificates on every device using the proxy, so that the user's browser does not issue warnings for the requests.
3. For each `google.com` request:

   a. Intercept the request.

   b. Add the HTTP header `X-GoogApps-Allowed-Domains`, whose value is a comma-separated list with allowed domain name(s). Include the domain you registered with Google Apps and any secondary domains you might have added.

   For example, to allow users to sign in using accounts ending `@altostrat.com` and `tenorstrat.com`, create the following header with the domain names you want to allow:

   ```
   X-GoogApps-Allowed-Domains = altostrat.com,tenorstrat.com
   ```
4. Optionally, create a proxy policy to prevent users from inserting their own headers.

Monitor URI Filtering

Avoid URI filtering with SSL inspection if possible. If you are using URI filtering, set up a policy to monitor URIs in proxy logs. Look for any URIs that were incorrectly blocked or allowed. These changes in the accessed URIs can cause Google Apps to load partially, slowly, or not at all. To avoid problems with URI filtering, if you are filtering your proxy servers, set up a policy for constant monitoring of your proxy load, and be prepared to adjust the rules if necessary.

To help discover what these new URIs might be, test new Google Apps features or services in a test environment before allowing their use in production. To help with this you can install a tool like **HttpWatch** or **HttpFox**.

Proxy Configuration Tools

Download the following tools which may be helpful when configuring Proxy Servers:

- Use `pactester` or a similar tool to validate PAC files for different URIs. Download pactester from the **Google Code site**.
- Download **HttpWatch** or **HttpFox** (Firefox extension) to help you see what URIs are being requested by the browser prior to encryption.

Other Network Services

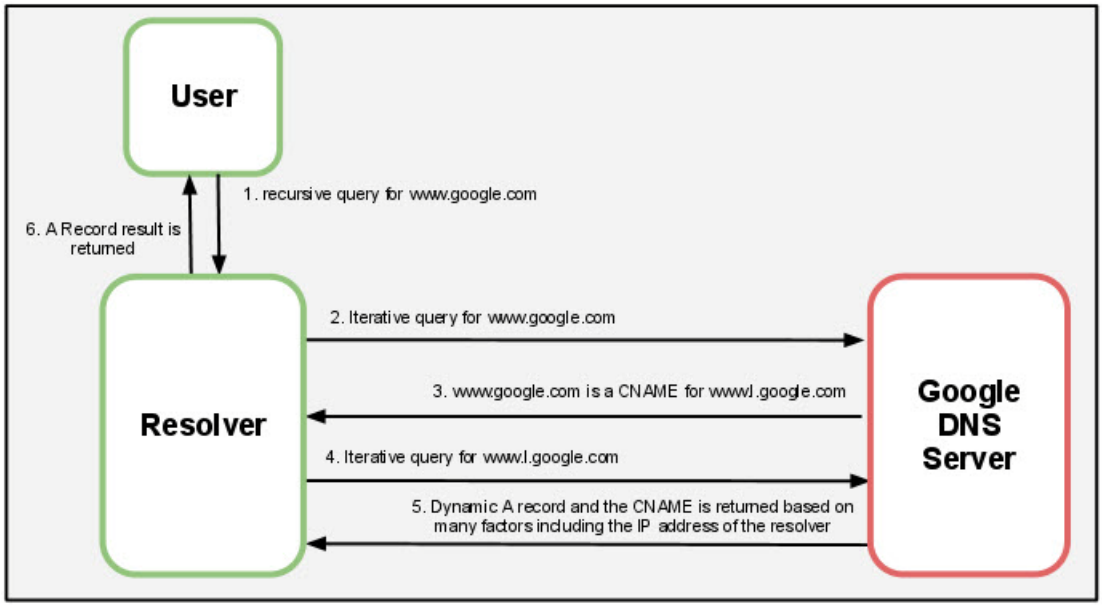
Google runs a sophisticated load-balancing system to ensure the best experience for the user. One factor in Google load-balancing systems is the way in which Google answers DNS requests for some services. Google attempts to determine the geographical location of a user partly through the location of the DNS resolver's IPv4 address.

To ensure the best experience for your users:

- Use a DNS resolver in a location that is close to the user, in terms of both geography and network topology. Using DNS resolvers located in remote network locations will greatly slow down connections to Google Apps.
- If it's not feasible to use a DNS resolver that's close to the user, use a DNS server that supports the edns-client-subnet extension (**Draft Proposal 2671**)—such as **Google's DNS server** or **OpenDNS**—which allows the resolver to pass part of the client's IP address.
- Adhere to the advertised TTL value for all DNS record types.
- Set up firewall rules to allow unrestricted outbound HTTPS traffic to Google Apps. You do not need to set up special rules for inbound traffic; Google Apps does not generally initiate inbound traffic to users.
- Avoid routing inbound and outbound mail through a gateway inside your network. If inbound and outbound mail is routed to a gateway inside your network, mail traffic will consume unnecessary network resources.

DNS Resolution

The diagram below shows a typical DNS resolution for a Google Apps user on an enterprise network.



Google serves DNS A record queries dynamically to ensure users receive the best experience at the time they make their request. To ensure that this occurs properly, configure your DNS caching resolvers to adhere to the TTL values specified with each record. Using the cached result beyond the TTL value on the DNS record can lead to a poor experience for the user, because the cached DNS record might direct users to a suboptimal IPv4 address.

Below is an example of the TTL values for www.l.google.com:

```
%dig +ttl www.l.google.com
```

For this query, you might see the following results:

```
; <<>> DiG 9.4.3-P3 <<>> +ttl www.l.google.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54488
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 4, ADDITIONAL: 4
;; QUESTION SECTION:
;www.l.google.com. IN A
;; ANSWER SECTION:
www.l.google.com. 184 IN A 209.85.225.104
www.l.google.com. 184 IN A 209.85.225.99
www.l.google.com. 184 IN A 209.85.225.103
www.l.google.com. 184 IN A 209.85.225.105
www.l.google.com. 184 IN A 209.85.225.147
www.l.google.com. 184 IN A 209.85.225.106
```

In this example, the TTL value is 184 seconds, which equates to 3 minutes. Be sure your DNS servers adhere to this value when caching results.

Using a centralized DNS server architecture will obscure the user making the request from Google's DNS servers, preventing Google from responding with an appropriate IPv4 address. If DNS queries are routed through a central server to resolve Internet hosts, users may not connect to the closest Google Apps servers. In extreme cases, this architecture can cause users in one continent to connect to servers in another, distant continent.

The ideal solution is to place local DNS resolvers close to the users. Then have the remote DNS resolvers send all DNS traffic through an Internet connection that's local to the users. Then, for internal-only addresses, forward the requests to the appropriate internal corporate DNS server.

Alternatively, you can use a DNS service that supports the edns-client-subnet extension (**Draft Proposal 2671**), such as **Google's DNS server** or **OpenDNS**.

Note: Clients and DNS servers using the edns-client-subnet extension require more data to be sent with the request, causing the traditional 512-byte limit to be exceeded. It's common for poorly implemented or configured services between the client and the authoritative DNS server to incorrectly handle the request. For more information, including instructions on how to test your infrastructure, see the **DNS- OARC site**.

Firewall Configuration

With Google Apps and other cloud applications, users reach outside your network for resources. This causes a shift of HTTP connections, from internal to external resources.

Because of this change, outbound firewalls that were previously properly sized in your network might become overwhelmed. Be aware of this possible increased footprint on your outbound firewall.

The average, peak, and idle connections from your benchmarking of proxy server load is a good estimate of the connection load to expect on your outbound firewall. The only connections you will not see on your outbound firewall are those that your proxy server does not allow through. For more information on gathering and using this data, see "Proxy Server Evaluation and Sizing" on page 16.

Outbound Firewall Rules

To ensure the best possible experience for users of Google Apps, and to provide a low-latency connection to our systems, we recommend leaving outbound firewall rules as open as possible on ports 80/443 for TCP/IP traffic.

Inbound Firewall Rules

Google Apps does not initiate connections from Google data centers into your network. All traffic is initiated by clients inside of your network to Google. The only exception to this is Google Talk video in certain circumstances. For more information, see "Google Talk Voice and Video and Google+ Hangouts" on page 21.

Mail Routing

After you change your MX records to route mail traffic to Google Apps, your email is no longer delivered to your servers by SMTP. Instead, inbound email is directed to the Google Apps servers. This essentially eliminates inbound SMTP mail traffic on your network.

Outbound Mail Connections

Depending on your needs, you may have some outbound mail traffic that you wish to send from your own network, such as automated or batched communications from applications in your system. You can use Postini Services to route and filter your outbound mail securely. If sending application mail outbound through Postini is not an option, you can also send mail through Google Apps as an authenticated Google Apps user over SSL.

Chapter 7

Client Configuration

Summary

It is important to understand the effects that different clients can have on the performance of Google Apps. This section discusses elements of the user environment that can affect Google Apps performance, suggestions for setting up authentication for use with Google Apps, and advice for migrating your users' data from an existing server into Google Apps.

Client Access

When planning for the clients that your users will use to access Google Apps, consider the following:

- For Google Apps, the recommended web browsers are Google Chrome, Mozilla Firefox, Microsoft Internet Explorer, or Apple Safari.
- The most up-to-date supported browsers are likely show the best performance in speed tests with large implementations of Google Apps.
- Consider the use of Android or ActiveSync mobile devices instead of BlackBerry devices. BlackBerry Enterprise Services can consume resources on your network.

Browser Requirements

Users will have a better experience with Google if they are using a modern browser that can render Google Apps content quickly and does not consume more processing and memory resources than necessary. Browsers that can send several requests in parallel to the same host greatly increase Google Apps performance and user experience.

A number of independent browser performance studies are available on the Internet. Consult these speed studies to get a better understanding of what browsers and browser versions have the fastest processing power for HTML and JavaScript content while consuming the fewest resources.

For Google Apps, we support the latest version of Google Chrome (which automatically updates whenever it detects that a new version of the browser is available). We also support the current and some previous major releases of Mozilla Firefox, Microsoft Internet Explorer, and Apple Safari. Check the **Administrator Help Center** for more information about supported versions of browsers.

Offline Access

Offline access can dramatically affect overall network bandwidth. Offline access causes network behavior for email and other applications to become similar to traditional email clients, since offline access uses data synchronization instead of immediate direct access. This behavior can cause load problems if all users have offline access enabled. If possible, enable offline access in Google Apps only for those users who require it.

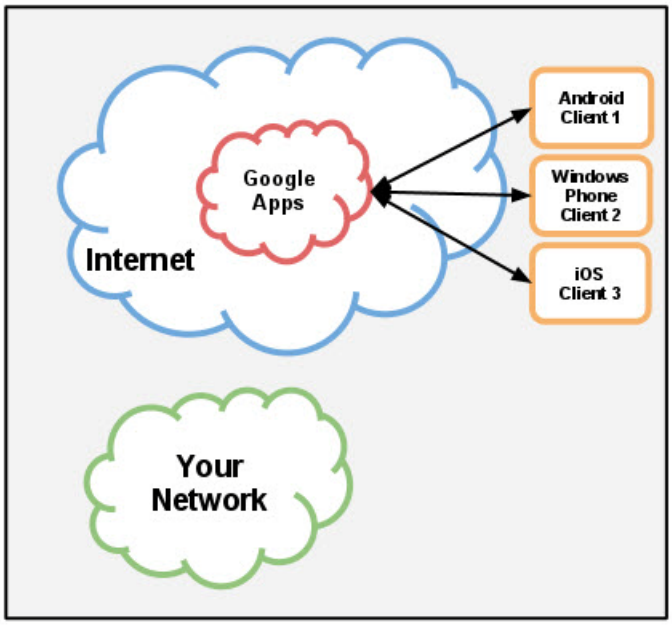
Mobile

In most cases, mobile clients have very little effect on your network load. This varies based on your specific mobile solution. See the sections below for details.

Android, iOS, and Windows Phone

Android devices (which use the Google Sync protocol) and Windows Phone and Apple iOS devices (which use the ActiveSync protocol) communicate directly to Google servers without using your network resources.

See the chart below for an illustration.



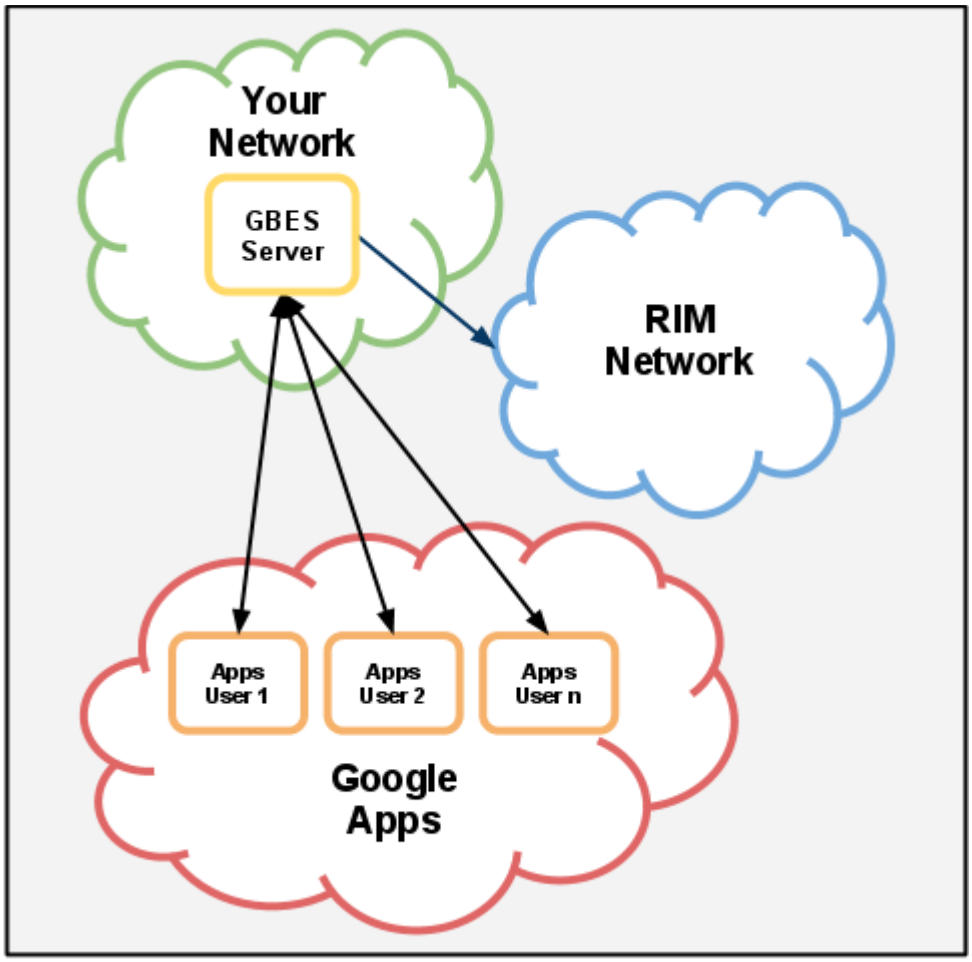
These devices do not access your network when using Google Apps. With ActiveSync or Google Sync, Google Apps delivers this mail directly to the user's device.

Google Apps Connector for BlackBerry Enterprise Server

Unlike Android and ActiveSync devices, BlackBerry Enterprise users do consume network resources when fetching mail from Google Apps, and when pushing data to the RIM network and the user's BlackBerry device.

Google Apps Connector for BlackBerry Enterprise Services acts as a bridge between Google Apps and the RIM network. The servers running Google Apps Connector for BlackBerry Enterprise Server does not need to be in a location close to the users, since users receive their message from RIM's network rather than the server itself. The communication between the BlackBerry handheld device and the BlackBerry server is similar to a BlackBerry server configured for Microsoft Exchange or Lotus Notes.

See the chart below for an illustration.



Expect additional network and server load when the BlackBerry device is activated for the first time. The server will send the user's data from the BlackBerry server running the Google Apps connector, to the user's handheld via RIM's network.

The largest amount of traffic between Google Apps and the Google Apps Connector for BlackBerry Enterprise Server occurs when a user is first added to the BlackBerry system via the BlackBerry admin panel. When a user is added to the system, the connector software will create a local cache of the user's email, calendar, and contacts. This local cache can be several hundred megabytes in size since it contains all user's data for the recent past (30 days by default). Monitor your bandwidth usage when adding multiple users at the same time.

Google Drive Sync Client

Google Drive includes an online **My Drive** folder and a local client, which both use HTTPS over TCP to sync files with each other. This two-way sync works even if the user's Google Apps domain uses Single Sign-On (SSO).

The Google Drive client determines what to sync based on a user's settings. By default, everything in the user's online **My Drive** folder syncs to the user's local **Google Drive** folder. Which file information is synchronized depends on whether the file is a Google Doc file type or another type of file, such as a PDF or graphics file:

- **Non-Google Doc file types:** Whenever a user uploads a file to Google Drive or changes it file name or location, Google Drive sends a push notification to the Google Drive client, which then syncs the entire file. If a user makes any changes to the content, file name, or location of the local copy of a file, the Google Drive client immediately detects it and automatically sends the entire updated file to Google Drive.
- **Google Docs file types:** The client stores only the metadata (title and folder location) locally on the user's machine. Therefore, the Google Drive client consumes less bandwidth when syncing with the online file.

We recommend that administrators encourage their users to convert binary documents to Google docs once they upload them to Google Drive, to leverage the collaboration built in to Google Drive. Also, if users use Google Drive to edit and share their documents, the Google Drive client won't need to sync larger binary files back up to the **My Drive** folder.

Authentication

Users can authenticate to the Google Apps service in two ways:

- Through your own Single Sign-On service
- Through Google Authentication

Large enterprise organizations often use a Single Sign-On system to authorize users. There are also options for cloud based Single Sign-On systems for smaller organizations.

Single Sign-On

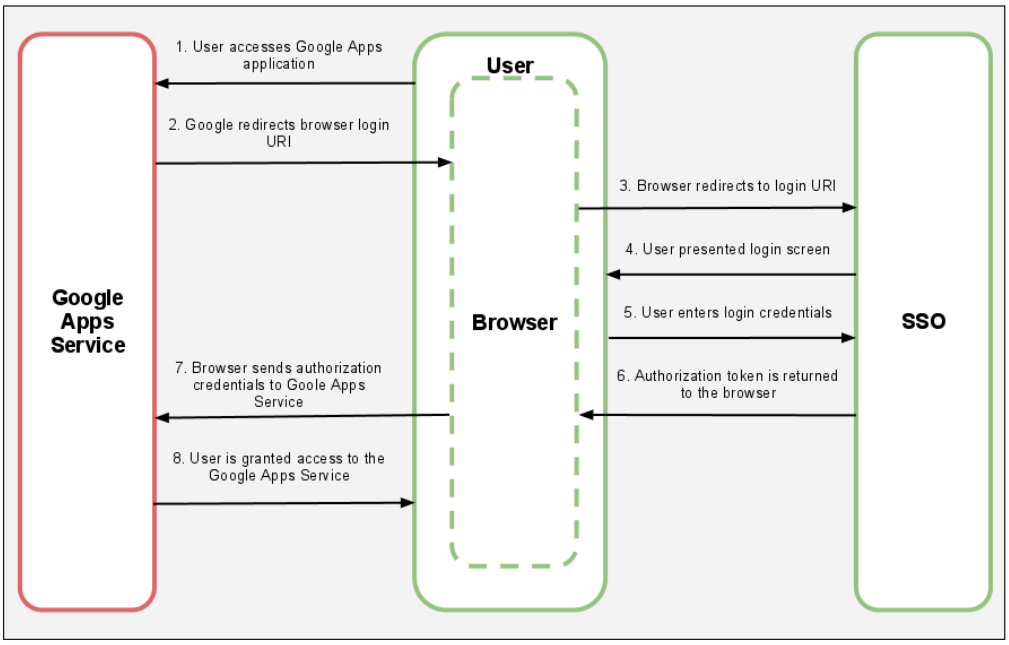
Google Apps supports SAML 2.0 based authentication for all Google Apps services. Client-side applications like Google Apps Sync for Microsoft Outlook also support Single Sign-On.

If you plan to set up Single Sign-On authentication, consider the following suggestions:

- Set up SSO servers in distributed network locations, rather than a central location.
- Set up internal DNS servers to redirect SSO traffic to the nearest SSO server, and ensure that alternate SSO servers are in place for redundant service in case of disruptions that prevent users from accessing the SSO server in a particular location.

Single Sign-On Process

When an unauthenticated user logs into Google Apps, and an SSO URI is configured for the domain, authentication takes several steps. See the chart below.



The process of SSO Authentication is as follows:

1. The user makes a request for a Google Apps service.
2. The Google Apps Authentication System redirects the user's browser to the configured URI for the SSO System. If the SSO/SAML server is not available, the user is unable to authenticate to the service.
3. The browser redirects to the login URI.
4. The SSO server displays a login screen.
5. The user enters login credentials and authenticates to the SSO System.
6. The SSO System passes an authorization token to the user's browser.
7. The user's browser sends the authorization credentials to the Google Apps Service.
8. The user is granted access to the Google Apps service.

Single Sign-On for Selected Network Locations

It is possible to create a conditional SSO system for your users that is based on a network subnet mask. This can be configured in the Google Apps control panel, under **Advanced tools -> Set up single sign-on (SSO)**. This type of configuration is recommended as it can be configured to control whether users outside your network use your SSO system.

A recommended setup is to configure Google Apps so that only users inside of your network require SSO authentication. Users outside of the network can use Google's authentication system instead. This ensures that users who cannot connect to VPN can still access basic mail services, and reduces the burden on your VPN services. This type of configuration requires that users accessing Google Apps without the use of the SSO system use a password stored in Google Apps.

Note: Google cannot enforce the use of SSL connections by third-party gadgets, Google Apps Marketplace apps, and other services. Please contact the appropriate providers of these services for clarification on their level of use of secure authentication.

Authentication Tools

A helpful tool to resolve any SAML-related errors during the authentication process is a SAML 2.0 debugger, such as **SAML 2.0 Debugger**.

Migration

Google Apps deployments often involve traffic from migrating user data, either through local clients like **Google Apps Migration for Microsoft Outlook,** or server-side clients like **Google Apps Migration for Lotus Notes** or **Google Apps Migration for Microsoft Exchange**.

If you are migrating user data as part of your Google Apps deployment, you can expect substantial data load, depending on the amount of data you choose to migrate. To limit the impact to your network, we recommend following these best practices:

- Ensure that your migration servers are in the same location as your legacy data servers, or at least that the connectivity between servers has low latency and high bandwidth.
- Avoid routing traffic from the migration servers to Google through proxy servers, to increase migration performance and to avoid unnecessary proxy server load.
- Assess your network capacity before migration to determine the maximum amount of data that you can migrate concurrently. Adjust your migration plan accordingly.
- During migration, some of the connections established to Google servers can stay open for a period of time depending on the migration tool. To avoid any possible migration errors, and to reduce the need to remigrate data, it is important to keep these sessions open and not close them prematurely with any proxy or firewall timeouts.

Server-side Migration

Your migration servers should have a low-latency, high-bandwidth connection to your email server. Migration is traffic- and bandwidth-intensive, and you can expect significant network load between your email server and the migration server.

Note: Do not install the migration software on the actual machines handling mail for your domain, since this will consume significant system resources.

Chapter 8

Network Monitoring

Summary

After your network is set up to work with Google Apps and your users are enabled, you can maintain the quality of your users' experience by monitoring the health of your network. To ensure the best user experience, follow these suggestions for monitoring tools and network traces.

Monitoring Tools

There are many commercial and open source tools to monitor various aspects of your network. A comprehensive directory of network monitoring tools is available on the **SLAC Network Monitoring Tools** site.

Specific recommended tools are listed in the table below.

Type of Monitoring	Tool	Description
Device Monitoring	**mrtg**	Monitors and graphs various aspects of network devices.
DNS changes	**dns-rr-monitor**	Monitors a specific resource record and alerts you to changes.
Host Monitoring	**smokeping**	Monitors and plots round-trip times to many destinations. Highly configurable.
Looking glass server	**Example list 1** **Example list 2**	A looking glass server provides a read-only view of a network operator's routing information -- including connections, latency, and other factors -- at a remote point on another network.
Network	**pingplotter**	Helps monitor network latency, uptime, and route changes.

Type of Monitoring	Tool	Description
Network	**multiping**	Helps monitor network latency, uptime, and route changes.
Packet Capture	**Wireshark**	Performs packet captures.
RTT latency	**wbox**	Attempts to measure RTT of web application latency using HTTP/TCP latency.
Trace	**tcptrace**	Similar to traceroute but uses TCP packets rather than ICMP packets.

Network Packet Captures

A network packet capture can help you to discover problems that may negatively affect the round-trip time or overall latency for Google Apps users, such as:

- Different types of network flooding problems (ARP, TCP, UDP, IP, etc.)
- MTU mis-matches for Ethernet
- Malicious traffic on your network

Packet captures are helpful even though Google Apps typically uses HTTPS connections. Packet captures will still show dropped packets, retransmits, window resizing, and evidence of saturated links.

One way to gather this type of data is to enable port mirroring, which allows you to capture traffic for a certain port or VLAN and divert it another port where a service listens and logs all the traffic. Another approach is to use technologies such as **Wireshark** to capture data on a machine for later analysis.