

## Evaluating external network bandwidth load for Google Apps

This document describes how to perform measurements to better understand how much network load will be caused by using a software as a service product from within your own network.

Traditional uses of software, such as thick client mail applications, require mainly local area network resources because they typically connect only to a local server, such as a mail server. In the case of email, users almost exclusively communicate with a local mail server, while the mail server is the only application which communicates with the Internet. Local mail may normally be confined to the local network and not cause any Internet traffic at all.

We recommend performing your own measurements to better understand what impact using software as a service, in particular Google Apps, may have on your network's Internet connection. Using published statistics may not provide you with accurate information as many organisations will have different characteristics for reasons including the following:

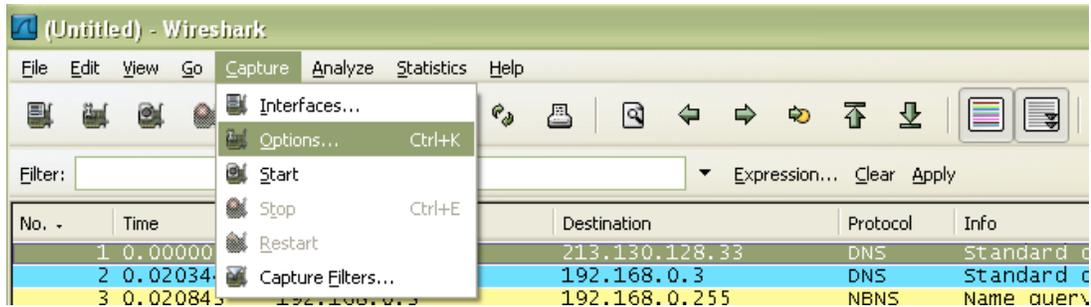
- Not all products provided in Google Apps may be enabled in your Google Apps domain instance. Having all users using all products all the time will incur more network load than users only using one application.
- The total network bandwidth consumed should be measured rather than just that from Google Apps. Users are likely to use the local network's Internet connection for many other tasks than just Google Apps.
- Individual peak loads and average loads are less important than average user latency. Simultaneous load caused by several users will increase latency and this is normally the limiting factor rather than bandwidth. This is also affected by other traffic to and from the Internet.
- User behaviour and culture regarding email can have an effect on total network load. If users adopt an approach to sharing documents with Google Docs rather than sending attachments, overall network load can be reduced.
- Enterprise users may access Apps over HTTPS rather than plain HTTP. This can have a dramatic effect on network load, especially if there is any local caching involved through advanced HTTP proxies.

### Capturing packet data

In order to generate statistics for network load, network packet data will need to be collected for analysis. Below are instructions for using [Wireshark](#) software (previously known as Ethereal) to capture the complete Internet packet data for one user accessing Google Apps mail.

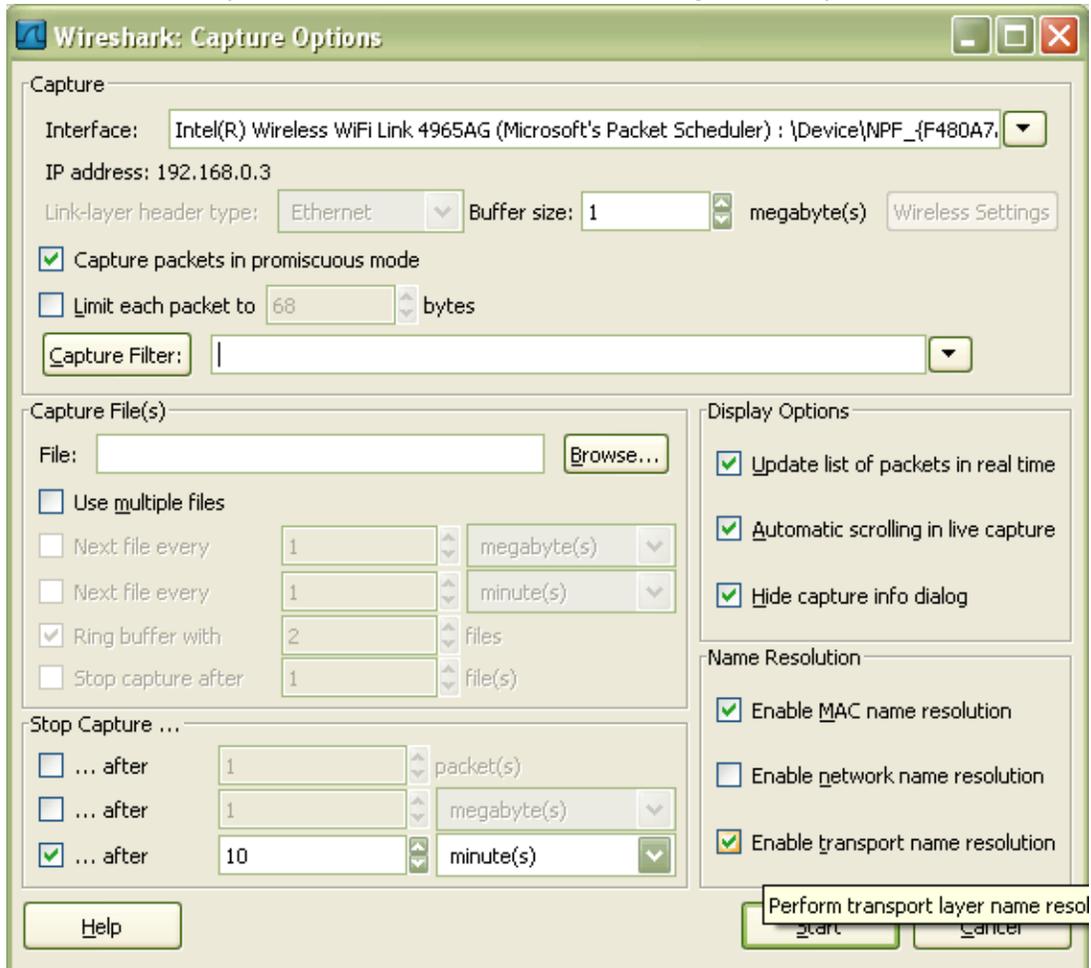
Download and install Wireshark network sniffing tools. This is a free open source application available from the Internet.

Once installed, if you are interested in capturing only Internet traffic you should apply filters to exclude any local network addresses. You will need to configure this according to your local network settings. From the **Capture** menu, select **Options**.



In the Options window, be sure to select the correct network interface. Add any capture filters in the Capture section, such as exclusion of any local network addresses (such as your gateway and local DNS servers). To make an accurate measurement, we suggest limiting the time for the capture to some specific period which will allow for meaningful data to be collected.

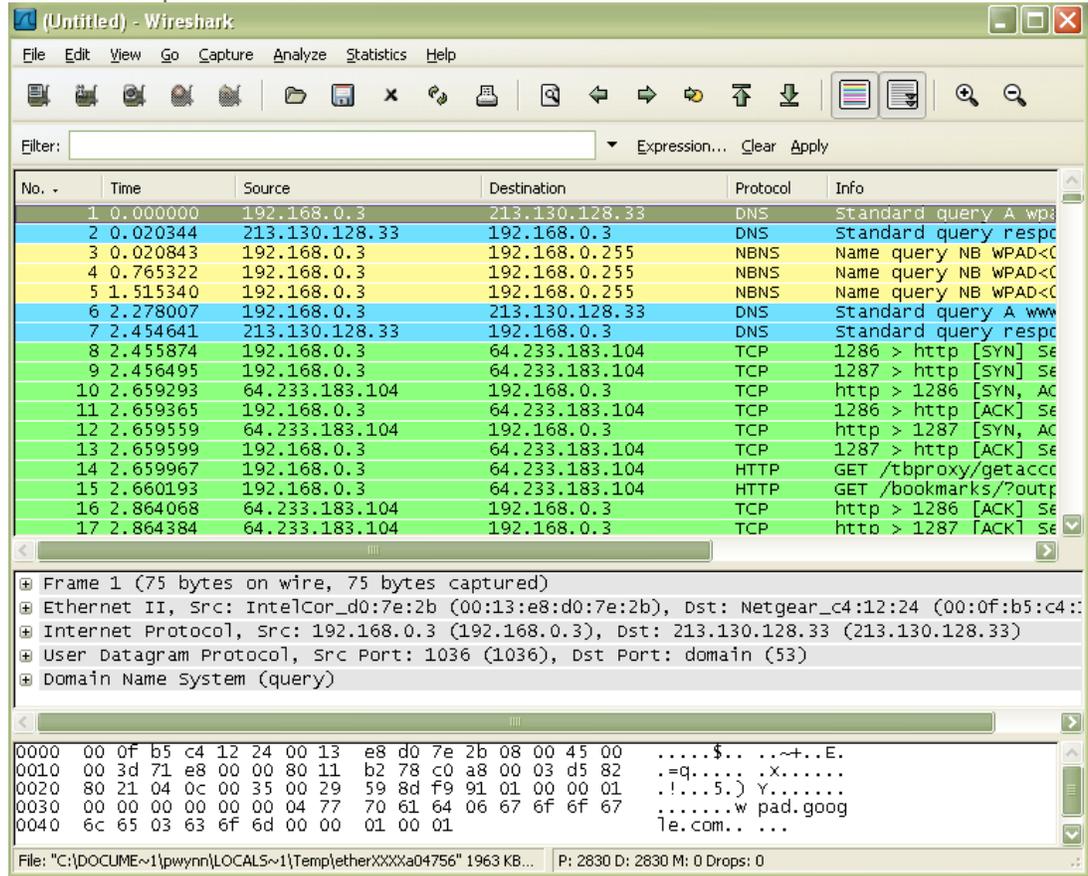
Once this is complete, click the `start` button to begin the capture.



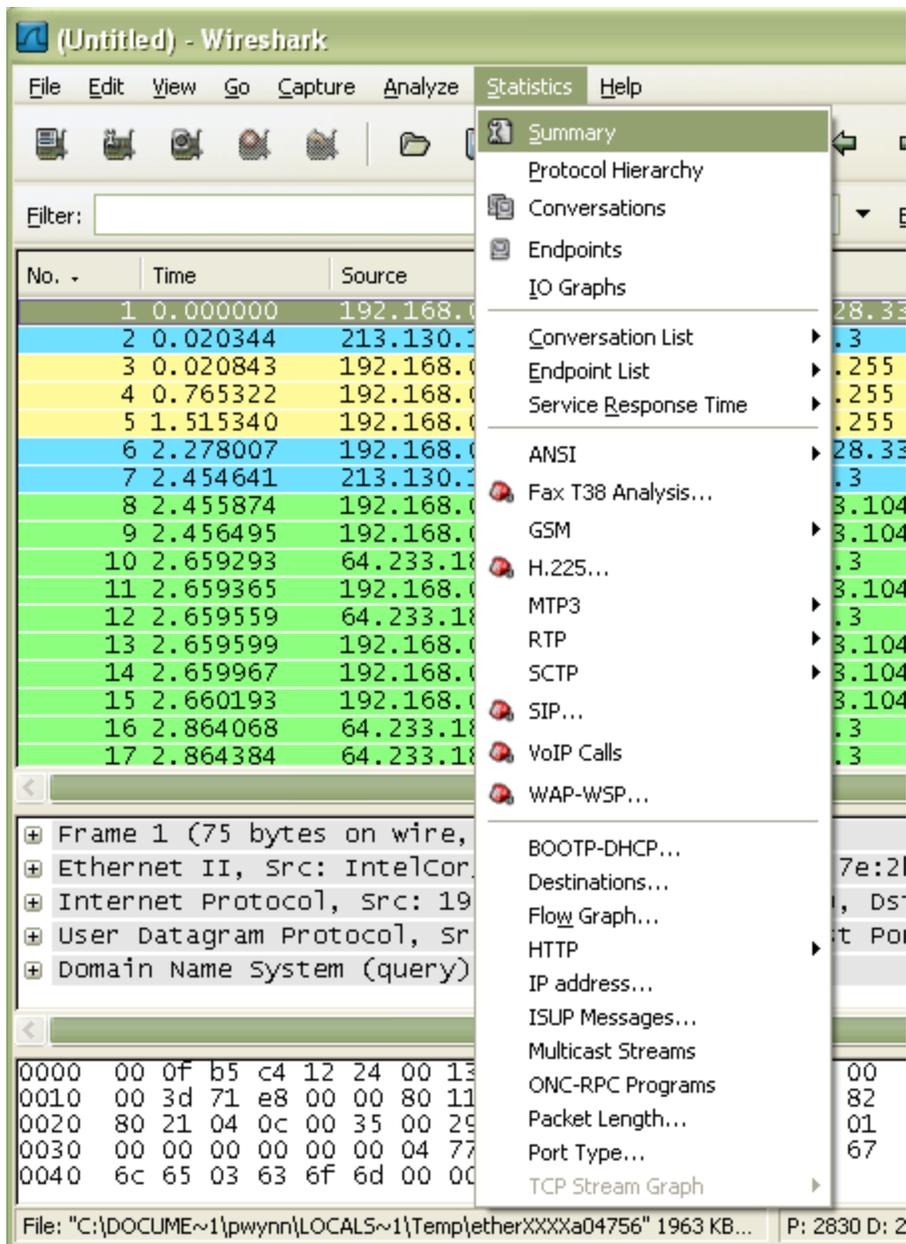
Once you have started the capture, log into Google Apps mail or other applications and make normal use of the software. The time limit applied to the capture will help you to forget that you are recording the packet data. It is important to make normal use of the tools when capturing data.

Once complete you will see all packets in the main Wireshark window. The

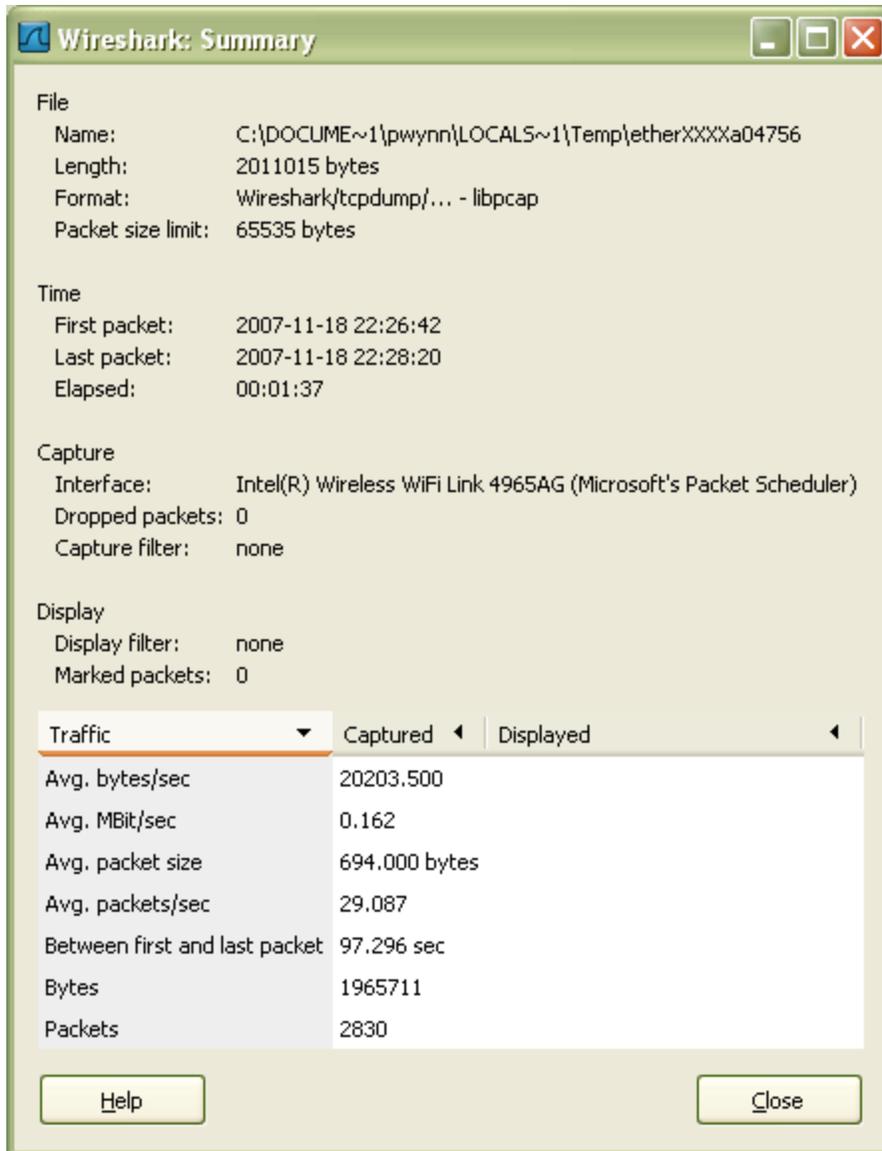
example below shows all data requested from the Internet. Your packet trace may not include information such as DNS requests if you have filtered out local network requests to DNS servers.



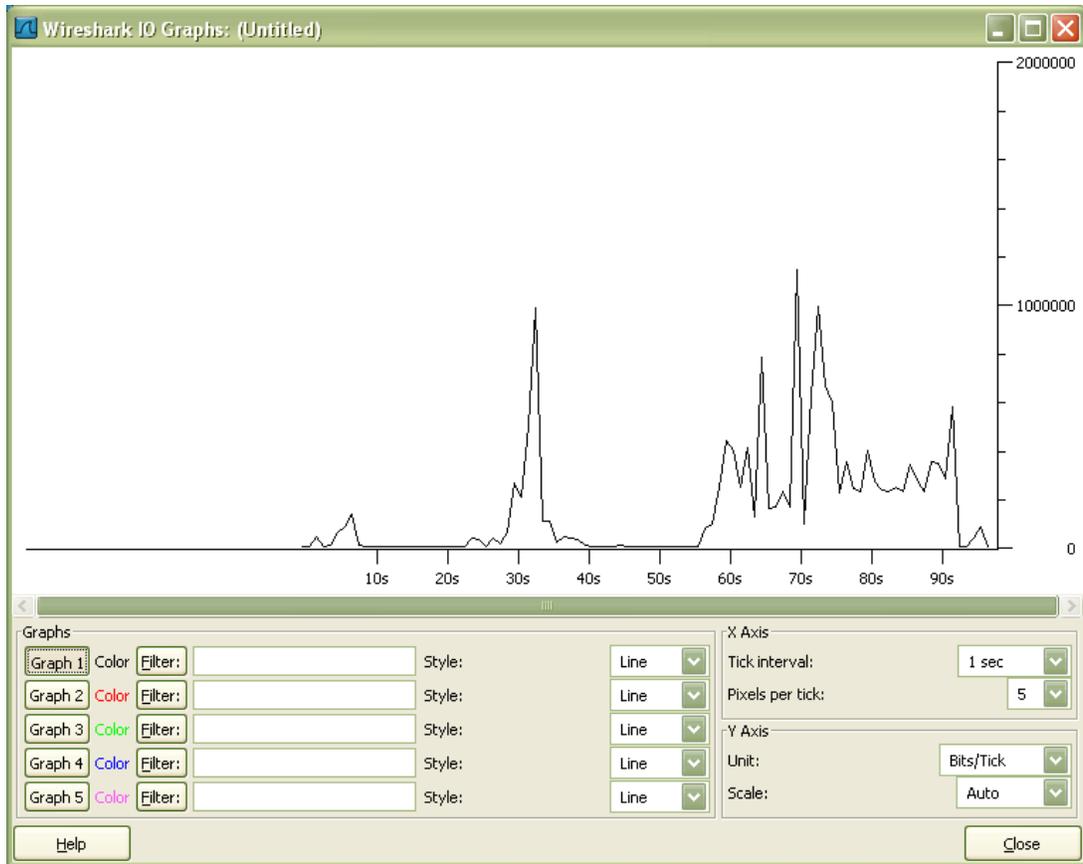
From the **Statistics** menu you can now select **Summary** to view a report of the last packet capture. This will provide an overview of the trace.



The average bandwidth consumed will provide a rough indication of how many concurrent users will be possible on one Internet connection, but this does not take simultaneous peaks into account.



To better understand the characteristics of the network use, you can view a traffic graph to see what peak transmission rates were caused and how frequent they were. For example, periods of sustained high load, such as for a minute may be hidden by average network bandwidth statistics. Identifying such periods of heavy activity will reveal how many concurrent users may share an Internet connection. This can be displayed by selecting **IO Graphs** from the **Statistics** menu.



The graph above shows that although the average data rates were reported to be approximately 160kb/s, there is a clear period of 40 seconds where the average data rate is in fact closer to 300kb/s. Combining users with the same activity profile on the same connection will still be possible when based on the average bandwidth, but latency will increase.

## Sample data from a client packet capture

The following is some sample data from a packet capture using Wireshark on Windows.

Specific conditions:

- No network filtering was applied (this includes complete network load from the client PC)
- Only Gmail was accessed.
- Gmail connection was purely over HTTPS
- Google Talk was enabled, also over HTTPS.

During the packet capture only Gmail was used directly, with no user time spent with other applications. As a result this is not a typical situation and the data is likely to be skewed higher than the common case. As mentioned above we recommend configuring and running your own packet capture to obtain more accurate statistics.

The first packet capture was purely to analyse the load imposed by logging in

and starting Gmail. No other user actions were performed during this time. This capture is depicted in the graph above.

- Capture time: 1:37 - This includes logging in through the standard user authentication service.
- 2011015 total bytes captured
- 162kb/s average
- Authentication causes an initial peak (close to 1Mb/s), but there is little load during the redirects.
- Once logged into Gmail, there is 40s of fairly sustained load at about 300kb/s.
- There is a peak of 1.2Mb/s for no more than 2 seconds once logged in.

The second packet capture shows what traced from continuous, normal user actions within Gmail. This does not include logging into the service, but all other conditions remain the same.

- Capture time: 9:58 - This includes 2 composed emails and reading some others)
- 940466 total bytes captured
- 12kb/s average
- This is active use - not using other software at the time.
- Peaks were seen twice just over 250kb/s for no more than 2 seconds each, while most time showed little activity.

As there are so many statistics and variables which will be of use to a network administrator, we recommend running your own packet capture.

## Example data from partners & customers

Below are some network statistics gathered from some of our customers and partners using their own tools or the ones mentioned above. We have not verified these statistics and they are merely provided as example guidance.

- A customer noticed that performance in IE 7 & 8 was considerably worse. This can be remedied as per below. [Q4 2009]
- A company of approximately 1500 people experienced about 50% capacity use of a 16mb/s shared internet connection for general, daily use of Google Apps. [Q2 2008]
- Google log-on consumption: +- 303 000 bytes (using Internet Explorer) +- 280 000 bytes (using Chrome) and +- 750 000 bytes (using firefox) [Q4 2008]
- Stay alive traffic was around 200 - 300 bytes per second average traffic over all browsers [Q4 2008]
- Accessing an email / composing an email - between 55 000 bytes and 110 000 bytes depending on the amount of text etc. [Q4 2008]
- Video chat requires at least 128kbps per client but will consume more if bandwidth is available, up to 1mbps. [Q4 2008]

## Appendix

### Poor performance in Internet Explorer

A customer was struggling with a very slow loading time for GMail on Internet Explorer 7 & 8 and we managed to track the cause being a registry setting that throttles IE to have a maximum of 2 concurrent threads per window. Once the customer adjusted up to 12, the Gmail load time went from ~120 seconds to 5 seconds.

The registry setting is

[HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings]

"MaxConnectionsPerServer". We found that this key was set to "2" by default at the customer's site. It is not clear if this was a customer-specific setting or one general to IE.

We do not support IE versions 6 or earlier.