

New York University

Computer Science Department

Courant Institute of Mathematical Sciences

1. Problem 1 – Modeling the network usage/performance of HTML pages download

Assuming you are surfing the Web and click on a link to an HTML page that never downloaded before, how much time will it take for your client browser to download the page? Please take into account the number of hops (and RTTs) it takes to obtain the IP address of the server hosting the page via a DNS lookup, the RTT and transmission time to communicate with the server hosting the various objects contained in the target HTML page, and consider the various HTTP configuration scenarios that were discussed in class

Answer:

The DNS lookup time will vary depending on whether or not the DNS entry is cached within the hierarchy of DNS servers (e.g., root, TLD, Authoritative, and Provider). We can refer to the DNS lookup time at t_{DNS} .

For non-persistent HTTP connections, it takes $2 \cdot \text{RTT} + T_{\text{transmit}}$ where RTT is the round trip time and T_{transmit} is the time it take to transfer an HTML object from the HTTP server to the client browser. In the case of non-persistent connections, it takes one RTT to setup a TCP connection and another RTT to get the HTTP response.

For a persistent connection, it takes one RTT to open the initial connection, which stays open for the duration of the keep alive, and a single RTT per HTML object thereafter.

So as an example, assuming we are trying to load an HTML page that contains four objects within it, it will take the following amount of time to load the page:

Non-persistent HTTP connection: $4 \cdot (2\text{RTT} + T_{\text{transmit}}) + t_{\text{DNS}}$

Persistent HTTP connection: $2 \cdot \text{RTT} + 4 \cdot T_{\text{transmit}} + t_{\text{DNS}}$

2. Problem 2 – Analyzing network usage/performance improvements in mainstream Web architectures

- (a) Research and explain the Common Gateway Interface (CGI) architecture that was created in the mid-1990s to add support for transactional services to Web based client-server applications. Model the performance of individual transactions in this context (only consider the transactional link used to invoke a remote program in your model).**

Answer:

The Common Gateway Interface (CGI) architecture leverages the original Web architecture, which was designed to return information contained in static HTML pages and/or interact with users via HTML forms, and adds the ability to insert links to programs (within HTML pages or as actions on forms) that can be executed on the server side in a single-threaded fashion. Typical CGI programs are calling dynamic scripts (e.g., Bourne Shell, Perl, and PHP scripts) that execute in their own separate process. Therefore, multiple concurrent requests will execute in a separate process and, as a result, request response time decreases as the number of concurrent requests increases. When the Web server detects a request for a link to a program, it spawns a process to run the corresponding script, which then executes and assembles an HTML response which is passed back to the Web server and sent back by the Web server to the original client that issued the request.

- (b) Research and explain the evolution of Web Frameworks since the mid-1990s and point out notorious performance improvements.**

Answer:

A tag-based approach was developed to augment the set of HTML tags with additional tags (e.g., ColdFusion Markup Language). Another approach was based on using a script-based approach that allowed the meshing of logic with HTML markup (e.g., PHP, ASP). These new tagging and scripting capabilities made it easier to generate HTML and it became possible at some point to process these tags and scripts in the same process as the Web server to optimize performance. A significant performance improvement was achieved with the introduction of application server technologies by Sun Microsystems and Microsoft. The two companies did not wait for the CORBA standard to fully emerge and started developing their own application server technology (i.e., Sun Microsystems' J2EE then JEE and Microsoft (DCOM, COM+, ASP.Net)). The major advantage of these technologies was to provide single process multithreaded engines (e.g., Sun Microsystems' Servlet Containers) to execute logic without having to spawn processes for each client request. More recently, another significant performance improvement was achieved via the use of AJAX technology made it possible to eliminate the need for Web page refreshes to dynamically render portions of Web content.

Please see below for recommended readings detailing specific Web frameworks and their evolution:

<http://fr.slideshare.net/mraible/the-future-of-web-frameworks>

http://raibledesigns.com/rd/entry/my_comparing_jvm_web_frameworks

<http://blog.websitesframeworks.com/2013/03/web-framework-comparison-matt-raible-opinion-138/>

- (c) Single Page Applications or “SPAs” are becoming the standard in modern Web frameworks based on Ajax and full stack approaches such as MEAN. Model a typical transaction scenario using SPAs, and explain the improvements achieved compared to earlier Web framework approaches.

Answer:

Single page applications leverage AJAX technology to dynamically update content in page regions based on user input. This approach eliminates the need for full page refresh and results in superior performance compared to the approaches used in older Web frameworks. Furthermore, full-stack programming using the MEAN ("MongoDB Express.js AngularJS Node.js") stack and Node.js allows the use of a single language (i.e., JavaScript in that case) on both the client and server tiers making it easier for new developers to get up to speed and prototype Web applications quickly. MongoDB's NoSQL features allows you to quickly change and alter the data layer.

Please see <https://thinkster.io/mean-stack-tutorial> to experiment with a typical transaction scenario using the MEAN SPA.

3. Problem 3 – Mainstream Internet Electronic Mail

- (a) Research the approach, conformance to RFCs, and specific protocols used to send and receive mail via iCloud, Google, Yahoo, Aol, and Outlook. Explain each approach and its specific differentiators in detail.

Answer:

| iCloud | Google | Yahoo | AOL | Outlook |
|--|--|---|--|---|
| IMAP is supported for receiving emails | Both IMAP and POP are supported for receiving emails | Both IMAP and POP are supported for receiving emails | Both IMAP and POP are supported for receiving emails | Both IMAP and POP are supported for receiving emails |
| SSL or TLS are required | SSL or TLS are required | SSL is required for incoming emails and TLS is required for outgoing emails | Both non-encrypted and encrypted connections are supported for receiving emails and only encrypted | SSL is required for incoming emails and TLS is required for outgoing emails |

| | | | | |
|--|--|---|---|--|
| | | | connections are supported for outgoing emails | |
| Port 993 is used for IMAP | Port 995 is used for POP and port 993 is used for IMAP | Port 995 is used for POP and port 993 is used for IMAP | Port 110 is used for regular IMAP and port 993 is used for SSL IMAP. Port 110 is used for regular POP connection and port 995 is used for SSL POP | Uses port 995 for POP and 993 for IMAP |
| Port 587 is used for SMTP | Port 465 is used if SSL is used, otherwise port 587 is used. If SSL is used, it is only possible to send emails to Gmail or Google applications users | Either port 465 for SSL or 587 for TLS are used. Mails can be sent either way | TLS and port 587 are the only options for SMTP connections | Only TLS can be used for outgoing connection and both ports 25 and 587 can be used |
| Non-conformant to RFC 5322 and RFC 3696 due to the fact that it does not support more than 21 characters for local part of the username (the specification mentions that applications have to support 64 octets). It also does not support non-encrypted connections | Non-conformant to RFCs 5322/3696 due to the fact that it does not support IPv4 address literals in email addresses. It also does not support non-encrypted connections | Mostly compliant to RFCs 5322/3696 but does not support non-encrypted connections | Requires mail to be compliant to RFC 2821/ 2822 (these RFCs are obsolete at this point). It also does not support non-encrypted connections | Mostly compliant to RFCs 5322/3696 but does not support non-encrypted connections |

(b) What specific information can you obtain using email headers and how do you gain access to such in the various approaches mentioned in 3. (a). Please provide examples and corresponding screenshots as necessary.

Answer:

As per RFC 5822, the information given in the table below can be placed in email headers:

| Field | Min Number | Max number | Additional Info |
|----------------|------------|------------|--|
| Trace | 0 | unlimited | |
| resent-date | 0* | unlimited* | One per block, required if other resent fields are present |
| resent-from | 0 | unlimited* | One per block |
| resent-sender | 0* | unlimited* | One per block, MUST occur with multi-address resent-from - see 3.6.6 |
| resent-to | 0 | unlimited* | One per block |
| resent-cc | 0 | unlimited* | One per block |
| resent-bcc | 0 | unlimited* | One per block |
| resent-msg-id | 0 | unlimited* | One per block |
| orig-date | 1 | 1 | |
| from | 1 | 1 | |
| sender | 0* | 1 | MUST occur with multi-address from |
| reply-to | 0 | 1 | |
| to | 0 | 1 | |
| cc | 0 | 1 | |
| bcc | 0 | 1 | |
| message-id | 0* | 1 | SHOULD be present |
| in-reply-to | 0* | 1 | SHOULD occur in some replies |
| references | 0* | 1 | SHOULD occur in some replies |
| subject | 0 | 1 | |
| comments | 0 | unlimited | |
| keywords | 0 | unlimited | |
| optional-field | 0 | unlimited | |

Obtaining an email header in iCloud:

- Log into the iCloud message box and select the email for which you want to see the header
- Click on the “View “ menu in the toolbar and go to Message>All Headers to show the message header

Obtaining an email header in Google:

- Log into the Gmail account and open the message for which you want to see the header
- Click on the down arrow on the top right of the message and select “Show Original” to show the email header

Obtaining an email header in Yahoo:

- Log into Yahoo mail and open the message for which you want to see the header
- Click on the tab “More” just above the top of the email and click on “View Full Header” to show the full header in a dialog box.

Obtaining an email header in AOL:

- Open the email message, click on details and choose “View Message Source” to show the header in a new window

Obtaining an email header in Outlook 2013:

- Double click on the email message to open it in a new window
- Click on the small arrow button shown in the “Tags” section of the “Message” tab
- The header of the message will show in the last section of the dialog box

4. Problem 4 – Network Management Tools

(a) Research and document at least five tools that can be used to manage the basic network capabilities that were described in class so far (e.g., traceroute, ping, nslookup).

(b) For each one of the tools identified in 4. (a), explain what type of information you can obtain using the tool and/or the service(s) it provides. Please provide examples and corresponding screenshots as applicable.

Answer (a) & (b):

The following tools are used to manage basic network capabilities described in class:

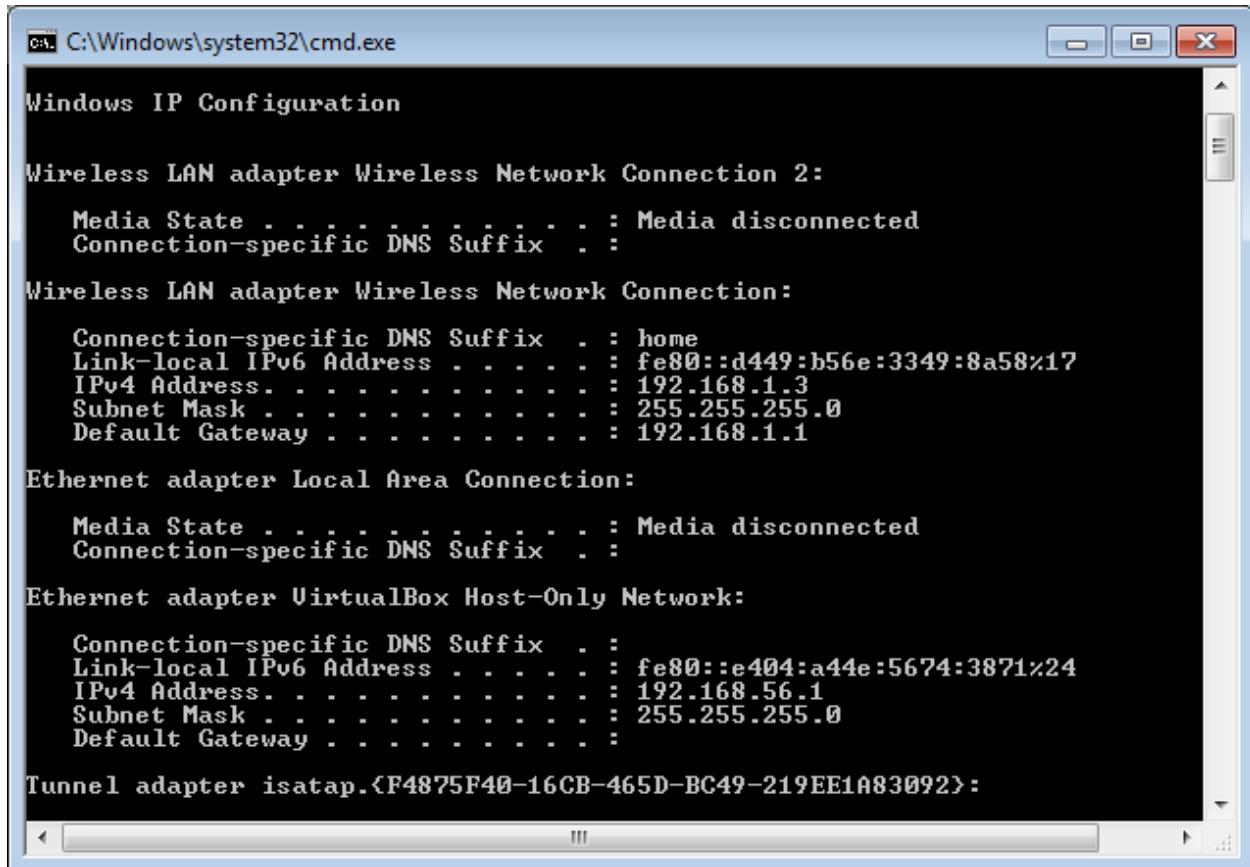
Ipconfig, netstat, gpresult, route, tracert, ping, nslookup

IPCONFIG:

This tool is used to obtain details information about network connections. It displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) (release and renew) and Domain Name System (DNS) setting (flushdns). The default gateway, subnet mask, and state of the network adapter are also accessible via this command.

The syntax of the command is shown below:

```
ipconfig [/all] [/renew [Adapter]] [/release [Adapter]] [/flushdns] [/displaydns] [/registerdns]
[/showclassid Adapter] [/setclassid Adapter [ClassID]]
```



```
Windows IP Configuration

Wireless LAN adapter Wireless Network Connection 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 
Wireless LAN adapter Wireless Network Connection:

    Connection-specific DNS Suffix  . : home
    Link-local IPv6 Address . . . . . : fe80::d449:b56e:3349:8a58%17
    IPv4 Address. . . . . : 192.168.1.3
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.1.1

Ethernet adapter Local Area Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 
Ethernet adapter VirtualBox Host-Only Network:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::e404:a44e:5674:3871%24
    IPv4 Address. . . . . : 192.168.56.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

Tunnel adapter isatap.{F4875F40-16CB-465D-BC49-219EE1A83092}:


```

Using the “/all” gives DNS, MAC, and other detailed information about each network component.

Using “/release” forces the network adapter to release the current IP address.

Using “/renew” forces the network adapter to renew the current IP address

To learn more about the parameters available for any command, you can add a slash “/” after the command name and it will show all the parameters available with that particular command.

NETSTAT:

Netstat is the command used in Windows (lsf is used in Linux/Unix).

This command is used to display the number of active TCP socket connections, the ports on which the computer is listening, and to which processes these sockets belong.

This command also display Ethernet statistics, the IP routing table, IPv4 statistics (for the IP, ICMP, TCP, and UDP protocols), and IPv6 statistics (for the IPv6, ICMPv6, TCP over IPv6, and UDP over IPv6 protocols).

When used without any parameters, netstat displays all active TCP connections.

Using “\a” shows all connections and listening ports.

Using “\b” displays the executable programs using the corresponding socket.

The syntax of the command is shown below:

```
netstat [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-x] [-t] [interval]
```

Please type “netstat –a –b” in the command window to see the output.

GPRESULT (Group Policy Settings):


```
C:\Windows\system32\cmd.exe

GPRESULT [/S system [/U username [/P [password]]] [/SCOPE scope]
[/USER targetusername] [/R ! /U ! /Z] [/X ! /H] <filename> [/F]

Description:
  This command line tool displays the Resultant Set of Policy (RSOP)
  information for a target user and computer.

Parameter List:
  /S      system      Specifies the remote system to connect to.

  /U      [domain\user] Specifies the user context under which the
                    command should execute.
                    Can not be used with /X, /H.

  /P      [password]  Specifies the password for the given user
                    context. Prompts for input if omitted.
                    Can not be used with /X, /H.

  /SCOPE  scope       Specifies whether the user or the
                    computer settings needs to be displayed.
                    Valid values: "USER", "COMPUTER".

  /USER   [domain\user] Specifies the user name for which the
                    RSOP data is to be displayed.

  /X      <filename>   Saves the report in XML format at the
                    location and with the file name specified
                    by the <filename> parameter. (valid in Windows
                    Vista SP1 and above and Windows Server 2008 and a
                    bove)

  /H      <filename>   Saves the report in HTML format at the
                    location and with the file name specified by
                    the <filename> parameter. (valid in Windows
                    Vista SP1 and above and Windows Server 2008 and a
                    bove)

  /F      Forces gpresult to overwrite the file name
                    specified in the /X or /H command.

  /R      Displays RSOP summary data.

  /U      Specifies that verbose information should
                    be displayed. Verbose information provides
                    additional detailed settings that have
                    been applied with a precedence of 1.

  /Z      Specifies that the super-verbose
                    information should be displayed. Super-
                    verbose information provides additional
                    detailed settings that have been applied
                    with a precedence of 1 and higher. This
                    allows you to see if a setting was set in
                    multiple places. See the Group Policy
                    online help topic for more information.
```

This command is used to determine the computer configuration for system and user settings. These can be configured centrally from a domain controller as well as locally from the computer itself.

In particular, "/r" is used to display the set of policies, when they were last processed, and the actual settings for computer and user policies.

ROUTE:

This command is used to view/manipulate the TCP/IP routing table.

```
C:\Windows\system32\cmd.exe

Manipulates network routing tables.

ROUTE [-f] [-p] [-4|-6] command [destination]
      [MASK netmask] [gateway] [METRIC metric] [IF interface]

-f          Clears the routing tables of all gateway entries. If this is
            used in conjunction with one of the commands, the tables are
            cleared prior to running the command.

-p          When used with the ADD command, makes a route persistent across
            boots of the system. By default, routes are not preserved
            when the system is restarted. Ignored for all other commands,
            which always affect the appropriate persistent routes. This
            option is not supported in Windows 95.

-4          Force using IPv4.

-6          Force using IPv6.

command     One of these:
            PRINT      Prints a route
            ADD        Adds a route
            DELETE     Deletes a route
            CHANGE     Modifies an existing route

destination Specifies the host.
MASK          Specifies that the next parameter is the 'netmask' value.
netmask       Specifies a subnet mask value for this route entry.
            If not specified, it defaults to 255.255.255.255.
gateway       Specifies gateway.
interface     the interface number for the specified route.
METRIC        specifies the metric, ie. cost for the destination.

All symbolic names used for destination are looked up in the network database
file NETWORKS. The symbolic names for gateway are looked up in the host name
database file HOSTS.

If the command is PRINT or DELETE. Destination or gateway can be a wildcard,
(wildcard is specified as a star '*'), or the gateway argument may be omitted.

If Dest contains a * or ?, it is treated as a shell pattern, and only
matching destination routes are printed. The '*' matches any string,
and '?' matches any one char. Examples: 157.*.1, 157.*, 127.*, *224*.

Pattern match is only allowed in PRINT command.
Diagnostic Notes:
    Invalid MASK generates an error, that is when (DEST & MASK) != DEST.
    Example> route ADD 157.0.0.0 MASK 155.0.0.0 157.55.80.1 IF 1
    The route addition failed: The specified mask parameter is invalid.
    (Destination & Mask) != Destination.

Examples:

> route PRINT
> route PRINT -4
> route PRINT -6
```

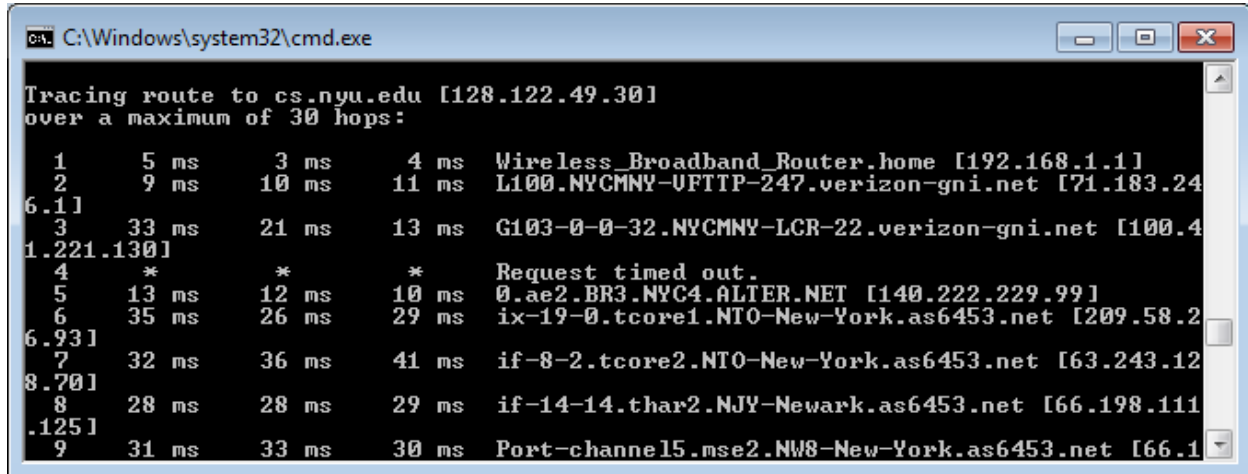
TRACERT (Trace Route):

This command is a route-tracing utility used to determine the path that an IP packet has taken to reach its final destination. The command shows all the hops to the destination. The command sends the first echo packet with a TTL of 1 and the TTL is then incremented by 1 for each subsequent transmission until the target responds or the maximum TTL is reached. The route is determined by examining the ICMP Time Exceeded messages sent back by intermediate routers (some routers drop packets with expired TTLs and are invisible to tracert). Please type "tracert [IP Address]" or "tracert [Host Name]" from the command prompt to run that command.

The syntax of the command is shown below:

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]

[-R] [-S srcaddr] [-4] [-6] target_name



```
C:\Windows\system32\cmd.exe

Tracing route to cs.nyu.edu [128.122.49.30]
over a maximum of 30 hops:
  0  5 ms  3 ms  4 ms  Wireless_Broadband_Router.home [192.168.1.1]
  1  9 ms  10 ms  11 ms  L100.NYCMNY-UFTTP-247.verizon-gni.net [71.183.24
  2  33 ms  21 ms  13 ms  G103-0-0-32.NYCMNY-LCR-22.verizon-gni.net [100.4
  3  *      *      *      Request timed out.
  4  13 ms  12 ms  10 ms  0.ae2.BR3.NYC4.ALTER.NET [140.222.229.99]
  5  35 ms  26 ms  29 ms  ix-19-0.tcore1.NTO-New-York.as6453.net [209.58.2
  6  32 ms  36 ms  41 ms  if-8-2.tcore2.NTO-New-York.as6453.net [63.243.12
  7  28 ms  28 ms  29 ms  if-14-14.thar2.NJY-Newark.as6453.net [66.198.111
  8  31 ms  33 ms  30 ms  Port-channel5.mse2.NW8-New-York.as6453.net [66.1
```

PING:

This command is used to send ICMP echo packets to the destination server (e.g., to check whether a host is alive). This command also provides other useful information such as TTL (Time to Live), Round trip time, packet loss, etc.

The command line usage is as follows:

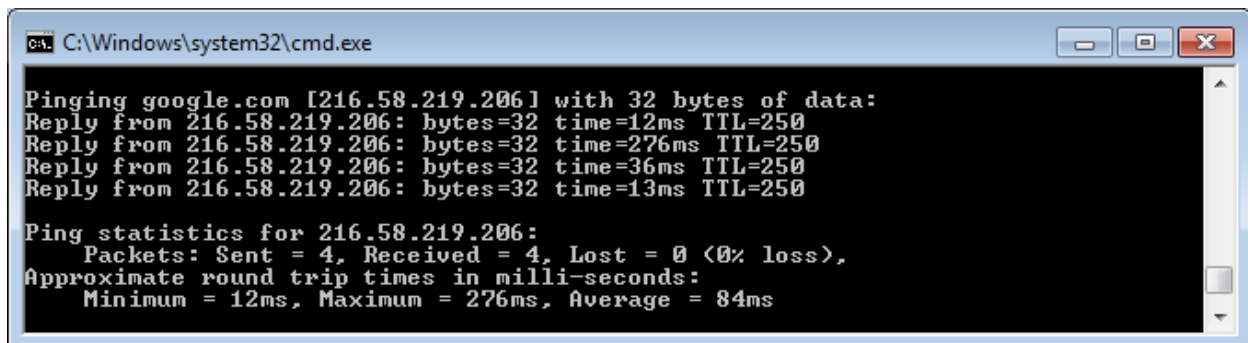
Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]

[-r count] [-s count] [[-j host-list] | [-k host-list]]

[-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]

[-4] [-6] target_name

Typing “ping google.com” on the command line results in the following:



```
C:\Windows\system32\cmd.exe

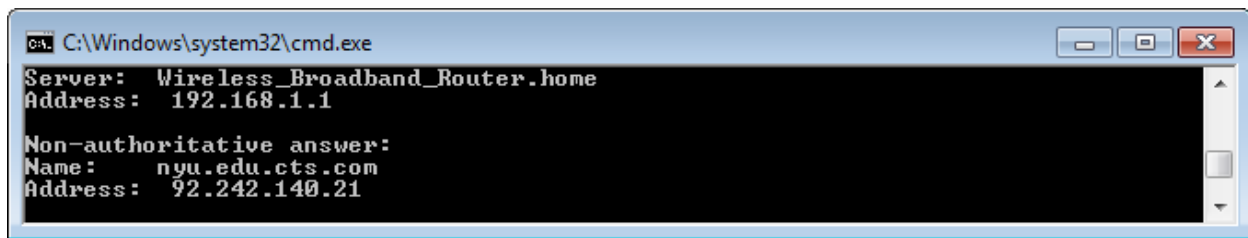
Pinging google.com [216.58.219.206] with 32 bytes of data:
Reply from 216.58.219.206: bytes=32 time=12ms TTL=250
Reply from 216.58.219.206: bytes=32 time=276ms TTL=250
Reply from 216.58.219.206: bytes=32 time=36ms TTL=250
Reply from 216.58.219.206: bytes=32 time=13ms TTL=250

Ping statistics for 216.58.219.206:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 12ms, Maximum = 276ms, Average = 84ms
```

NSLOOKUP:

This command performs a DNS lookup.

Typing “nslookup nyu.edu” on the command line results in the following:



```
C:\Windows\system32\cmd.exe
Server: Wireless_Broadband_Router.home
Address: 192.168.1.1

Non-authoritative answer:
Name: nyu.edu.cts.com
Address: 92.242.140.21
```

You can turn on the debug flag “-debug” to get additional details.

5. Problem 5 – P2P File Distribution via BitTorrent

(a) Research and explain how DHTs are used in BitTorrent to create a distributed tracker. Please be as specific as possible and provide usage scenarios including peer churn.

Answer:

BitTorrent uses a DHT to store peer contact information and the corresponding protocol is implemented on top of UDP. This approach enables each peer to become a tracker.

Peer discovery mechanism usage scenario:

Node IDs are chosen at random within a 160 bit space to ensure an extremely low probability of collisions.

All torrent nodes maintain a routing table with information about the nodes that are close to each other. The proximity of nodes is established via a distance metric (i.e., bitwise XOR of Node ID and BitTorrent’s InfoHash) that compares the Node ID of a given routing table node with the BitTorrent’s InfoHash.

To find peers in a torrent, a given node uses the aforementioned distance metric to compare the BitTorrent’s InfoHash with the IDs of the nodes in its routing table. It then contacts the nodes it knows about that have IDs closest to the InfoHash to ask them for the contact information of peers currently downloading from the torrent. If a contacted node knows about the peers of a torrent, it responds with the peers contact information. In the alternative, the node responds with information for the nodes in its routing table that are closest to the InfoHash of the torrent. The original node keeps on querying iteratively until it cannot find any nodes that are closer to the InfoHash. After the search is exhausted the original node inserts its own peer contact information onto the nodes that are closest to the InfoHash of the torrent.

Routing table update and Peer Churn handling scenarios:

In BitTorrent the bucket node ID space is from 0 to 2^{160} (160 bits total). The bucket size is currently 8. Initially, there is only a single bucket and so any node is added to the bucket. Thereafter, using the peer discovery mechanism (see 5.a.) more closer nodes are added to the bucket.

A node is a “good” node if it has responded to a query within the last 15 minutes. When a bucket is full with good nodes, it is split into two buckets with ranges 0 to 2^{159} and 2^{159} to 2^{160} , only if the node ID falls into the range of the bucket, otherwise the good node is discarded.

To identify peer churn, the good nodes are queried on a regular basis, so a peer that has churned will not respond to the query, in which case it will be queried once more before being removed from the bucket. A new good node will then replace it in the bucket.

(b) What potential issues need to be considered to optimize the performance of DHT usage in practical BitTorrent deployments? Please provide examples of issues and suggested improvements and estimate the performance savings.

Answer:

The process of updating the DHT of nodes must be optimized to ensure scalability. Requiring each node to keep track of the other nodes require lots of memory as well as computing overhead as the number of peers increases. A more efficient approach consists of using a circular DHT that only requires nodes to keep track of successor and predecessor nodes ($N/2$ messages exchanges as required in that case). The approach employed by practical BitTorrent networks is more efficient in that it requires each node to keep track of the closest nodes by computing the aforementioned distance metric (see 5.a). Using this approach, the overhead of maintaining peer information and the number of messages that need to be exchanged is much lower and the actual complexity is $(\log(n))$ for both the number of messages that are exchanged and the number of peers that need to be tracked.