CSE 304

PYTHON PROGRAMMING WITH WEB FRAMEWORKS

UNIT – 3

Internet Client Programming

Dr.S. Priya/AP-III/ICT

Internet Client Programming

- Internet Client:
  - use the services provided by the internet server
- Protocols
  - File Transfer Protocols(FTP)
  - Network News Transfer Protocol (NNTP)
  - E-Mail
    - ✓Simple Mail Transfer Protocol(SMTP)- (Mail sending)
    - ✓PoP3,IMAP4 (receiving Mail)

File Transport Protocols (FTP)

- Transfer file between two computers
- To download s/w and code
- Login credentials
- Anonymous login
  - Gust login
  - Unregistered user
- Use only TCP

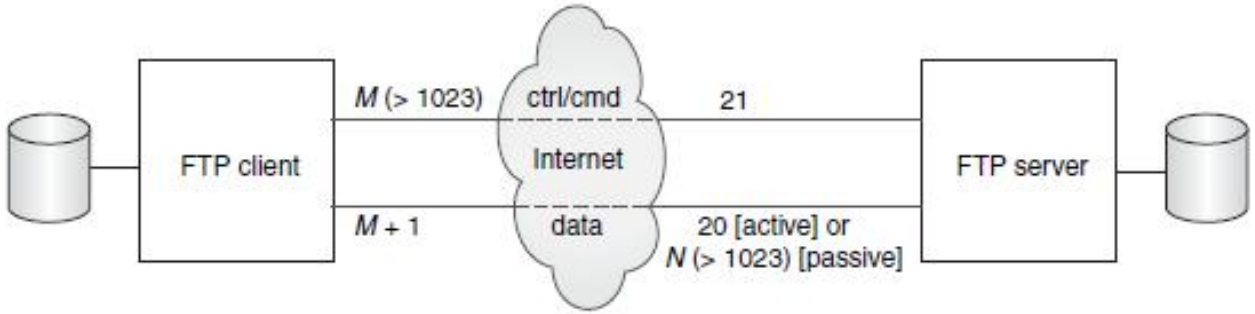


Figure 3-1 FTP Clients and Servers on the Internet. The client and server communicate using the FTP protocol on the command or control port data; is transferred using the data port.

Python with FTP

1. Connect to server
2. Log in
3. Make service request(s) (and hopefully get response[s])
4. Quit

- Module imported
  - ftplib
    - ✓ FTP
- Connecting to FTP server
  - f=FTP('host address')
- Login with FTP server
  - f.login(user='anonymous ',passwd=' ',acct=' ')
- Closing connection
  - f.quit()

ftplib.FTP class methods

Methods	Description
login(user='anonymou s', passwd='', acct='')	Log in to FTP server, all arguments are optional
pwd()	Current working directory
cwd(path)	Change current working directory to path
dir( )	Displays directory listing pf path
nlst( )	Like dir(), but returns a list of filenames instead of displaying
retrlines(cmd[,cb..])	Download text file given FTP cmd RETR filename;
retrbinary(cmd, cb [, bs=8192, [, ra]])	Similar to retrlines() for binary file; callback cb for processing each block downloaded required;

Methods	Description
storlines(cmd, f)	Upload text file given FTP cmd, STOR filename; open file object f required (in binary mode)
storbinary(cmd, f, [, bs=8192])	Similar to storlines() but for binary file; open file object f required
rename(old, new)	Rename remote file from old to new
delete (path)	Delete remote file located at path
nlst( )	Like dir(), but returns a list of filenames instead of displaying
mkd(directory)	Create remote directory
rmd(directory)	Remove remote directory
quit()	Close connection and quit

Example -1

```
>>>from ftplib import FTP
>>>f=FTP('ftp1.at.proftpd.org')
>>>f.login()
'230 Anonymous access granted, restrictions apply'
>>>f.dir()
-rw-rw-r--   1 ftp      ftp          451 Jul  1  2005 README.MIRRORS
drwxrwxr-x   3 ftp      ftp         4096 Jul  1  2005 devel
drwxrwxr-x   3 ftp      ftp         4096 Dec  2  2010 distrib
drwxrwxr-x   4 ftp      ftp         4096 Jul  1  2005 historic
>>>f.pwd()
'/'
>>>f.storlines('stor sample.txt',open('sample.txt','rt'))
>>>f.quit()- ‘Goodbye.'
```

FTP with exception

```
import os
import socket

HOST = 'ftp.cs.brown.edu'
DIRN = 'pub/papers/theory'
FILE = 'cs252.tar.gz'

def main():
    try:
        f = ftplib.FTP(HOST)
    except (socket.error, socket.gaierror) as e:
        print('ERROR: cannot reach "%s"' ,HOST)
        return
    print('*** Connected to host "%s"' ,HOST)
```

```
    try:
        f.login()
    except ftplib.error_perm:
        print ('ERROR: cannot login anonymously')
        f.quit()
        return
    print ('*** Logged in as "anonymous"')

    try:
        f.dir()
        f.cwd(DIRN)
    except ftplib.error_perm:
        print ('ERROR: cannot CD to "%s"' , DIRN)
        f.quit()
        return
    print ('*** Changed to "%s" folder' , DIRN)
    f.dir()
```

FTP with exception program continuation

```
    try:
        f.retrbinary('RETR cs252.tar.gz',open(FILE, 'wb').write)
    except ftplib.error_perm:
        print ('ERROR: cannot read file "%s"' ,FILE)
        os.unlink('README')

    else:
        print ('*** Downloaded  to CWD' ,FILE)
    f.quit()
    return


if __name__ == '__main__':
    main()
```

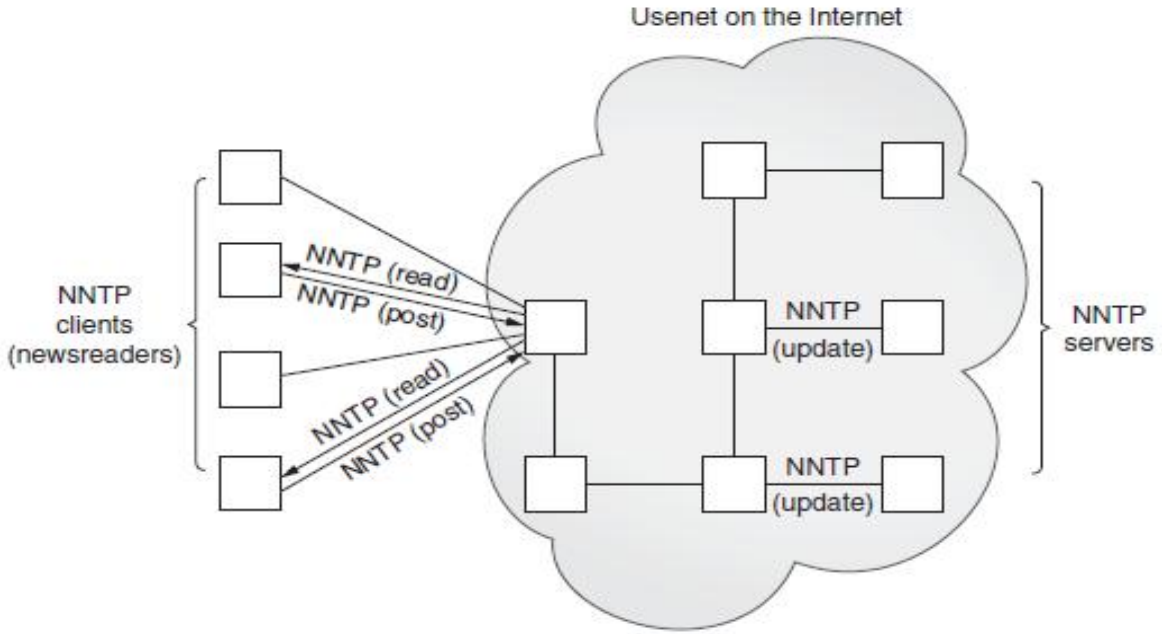
```
*** Connected to host "%s" ftp.cs.brown.edu
*** Logged in as "anonymous"
drwxrwxrwt   2 0        0          6975158 May  4 23:11 incoming
drwxr-xr-x  33 0        2             2323 Aug 13  2013 pub
drwxr-xr-x  52 0        2             1087 Feb 10  2016 u
*** Changed to "%s" folder pub/papers/theory
-rw-rw-r--   1 0        1200           470 Aug 23  1994 INDEX
-rw-r--r--   1 8186     1200      14707731 Aug 21  1997 cs252.tar.gz
-rw-r--r--   1 8186     1200        122469 Jun 21  1993 exptrees.ps.Z
-rw-r--r--   1 2506     1200         59993 Mar 19  1995 theorem.tar.Z
-rw-r--r--   1 2506     1200       1861208 Aug 23  1994 tutorial-talk.ps
-rw-r--r--   1 2506     1200        761790 Aug 23  1994 tutorial.ps
*** Downloaded  to CWD cs252.tar.gz
```

Newsgroup and Usenet

- Newsgroup:
  - Discussion forum - about particular topics in a server (article,post)
  - enable remotely connected users to share, discuss and learn about their topic of interest by exchanging text messages, images, videos and other forms of digital content
- Usenet
  - S/W to allow the client to connect to the newsgroup
- Organized based on subject wise
  - news,soc(sciety),sci (science),comp(computers),...
  - Main category, sub category
  - eg: Comp.lang.python

Network News Transfer Protocol (NNTP)

- Download or post article/message
- Simple
- One port 119 - login,data and control



Python and NNTP

- Connect to server
- Login (optional),but choose newsgroup of internet
- Make service
- Quit
- Module imported

```
from nntplib import NNTP
```

- Connecting to server

```
N=NNTP('host address')
```

- Choose newsgroup

```
r,c,f,l,g=n.group('newsgroup')
```

- Closing connection

```
n.quit()
```

nntplib.NNTP class methods

method	description
group(name)	Select newsgroup name & return a tuple (rsp, ct, fst, lst, group)
xhdr(hdr, artrg[, ofile])	Returns list of hdr headers for article range artrg
body(id[, ofile])	Get article body given its id, which is either a message ID (enclosed in <...> or an article number (as a string)
head(id)	Similar to body(), contain article headers
article(id)	Similar to body(), contain both headers and article body
stat(id)	Set article “pointer” to id, returns tuple similar to body (rsp, anum, mid) but contains no data from article
next()	Used with stat(), moves article pointer to next article and returns similar tuple
last()	Used with stat(), moves article pointer to last article and returns similar tuple
post(ufile)	Upload data from ufile file object (using ufile.readline()) & post to current newsgroup
quit()	Close connection and quit

```
>>>import nntplib
>>>n=nntplib.NNTP('news.gmane.io')
>>>resp, count, first, last, name = n.group('gmane.comp.python.committers')
>>>resp,overviews=n.over((first,first+10))
>>> for id,over in overviews:
        print(id,decode_header(over['subject']))
```

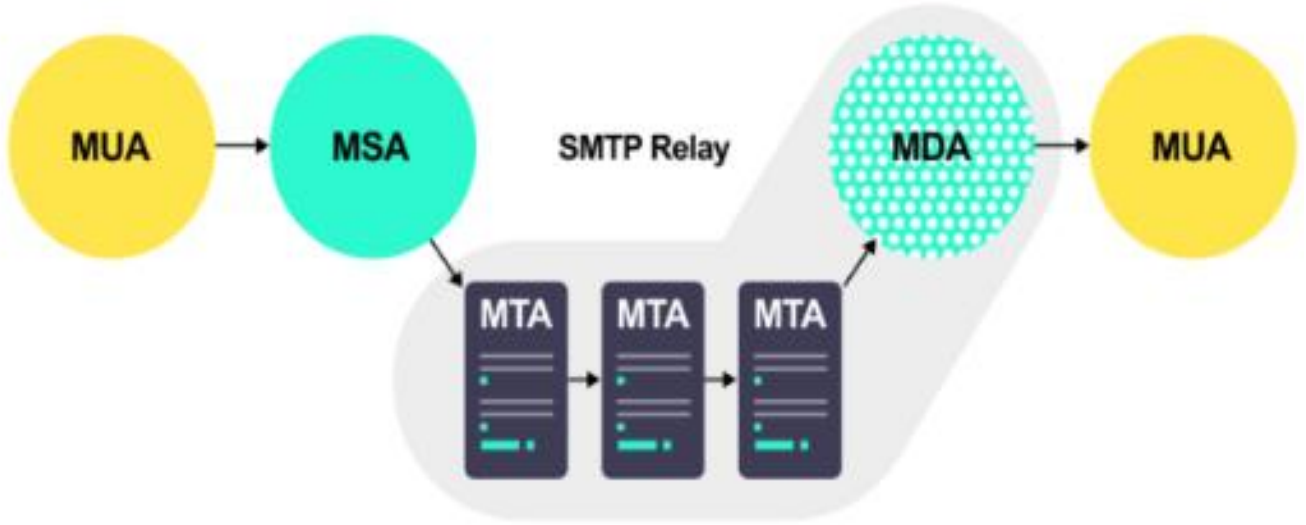
```
...
1 Nominate Hirokazu Yamamoto (oceancity) for commit privs.
2 Re: Nominate Hirokazu Yamamoto (oceancity) for commit privs.
3 Re: Nominate Hirokazu Yamamoto (oceancity) for commit privs.
4 Re: Nominate Hirokazu Yamamoto (oceancity) for commit privs.
5 Re: Nominate Hirokazu Yamamoto (oceancity) for commit privs.
6 Re: [Python-Dev] next beta
7 Re: [Python-Dev] next beta
8 Re: [Python-Dev] next beta
9 Re: Nominate Hirokazu Yamamoto (oceancity) for commit privs.
10 Re: [Python-Dev] next beta
11 Re: [Python-Dev] next beta
>>>
```

```
>>>from datetime import date,timedelta
>>>resp,groups=n.newgroups(date.today()-timedelta(weeks=15))
>>>for i in range(len(groups)):
        print(i,":",groups[i])
```

```
0 : GroupInfo(group='gmane.comp.lang.hare.announce', last='1', first='1', flag='m')
1 : GroupInfo(group='gwene.com.electronicbook', last='11', first='1', flag='m')
2 : GroupInfo(group='gwene.com.takeonrules', last='61', first='1', flag='m')
3 : GroupInfo(group='gwene.org.vuxml.freebsd', last='133', first='1', flag='m')
4 : GroupInfo(group='gwene.de.books.ub.bonn', last='556', first='1', flag='m')
5 : GroupInfo(group='gwene.de.uni-kiel.ub', last='80', first='1', flag='m')
6 : GroupInfo(group='gwene.at.books.wien.bibliothek.digital', last='2043', first='1', flag='m')
7 : GroupInfo(group='gwene.de.books.wolfenbuettel.digital', last='385', first='1', flag='m')
```

E- Mail

- Exchange of information between users(one/group)
- MUA - Mail user Agent (end user - sender or receiver)
- MSA - Mail Submission Agent
- MDA - Mail Destination Agent
- MTA (Mail/Message Transfer Agent) - s/w to transfer the email -> using multiple agent
- MTA - transfer email using SMTP:
  - Open-source MTAs
    - SendMail– Postfix– Exim– Qmail
  - Commercial MTAs–
    - Microsoft Exchange, Lotus Notes Domino Mail Server
- Sending Email - SMTP
- Receiving Email - POP3,IMAP4



Sending email using SMTP

1. Connect to server
2. Login (if applicable)
3. Make server request
4. Quit

- Module to be imported

  From smtplib import SMTP

- Connection to server

  n=SMTP('smtp.yourdomain.com',portno)

- login

  Login with own username and password

- Closing the connection

  n.quit()

smtplib.SMTP class methods

method	description
login(user, passwd)	Log into SMTP server with username and passwd
sendmail(from, to,msg[, mopts, ropts])	Sends msg from from to to (list/tuple) with optional ESMTP mail (mopts) and recipient (ropts) options
ehlo() or helo()	Initiates a session with an SMTP or ESMTP server using EHLO or HELO, respectively. Should be optional because sendmail() will call these as necessary
starttls(keyfile=None, certfile=None)	Directs server to begin Transport Layer Security (TLS) mode. If either keyfile or certfile are given, they are used in the creation of the secure socket.
set_debuglevel(level)	Sets the debug level for server communication.
quit()	Closes connection and quits.

Note

- goto manage your settings of your gmail account.
- select security tab from left panel.
- select Less secure app access (by default it will be off state) change that to on state.
- ensure once the demo is over again set this select secure app access with off state.

Sending email using smtp

```
from smtplib import SMTP
import getpass

server=SMTP('sm

server.starttls()

server.ehlo()
    (250,  b'smtp.

server.login('regn
password'))

    enter passwor
```


```
sent_from = "regno@sastra.ac.in"
to = ["regno@sastra.ac.in"]
subject = 'Message From Python Code'
body = "Hello every one!,welcome to python class"

", ".join(to), subject, body)

ent_from, to, email_text)

print ('Email sent!')
                    'Email sent!
server.quit()
```

Receiving email

- POP3 (Post Office Protocol v3)
- IMAP4 (Internet Message Access Protocol v4)
  - Access mail from more than one device
  - Synchronize many device

Receiving email using POP3

1. Connect to server
2. Login (if applicable)
3. Make server request
4. Quit

- Module:

```
from poplib import pop3
```

- Connecting to server:

```
p-=pop3('pop.yourdomain.com')
p.user('user name')
p.pass_('passwd')
```

- Closing the connection

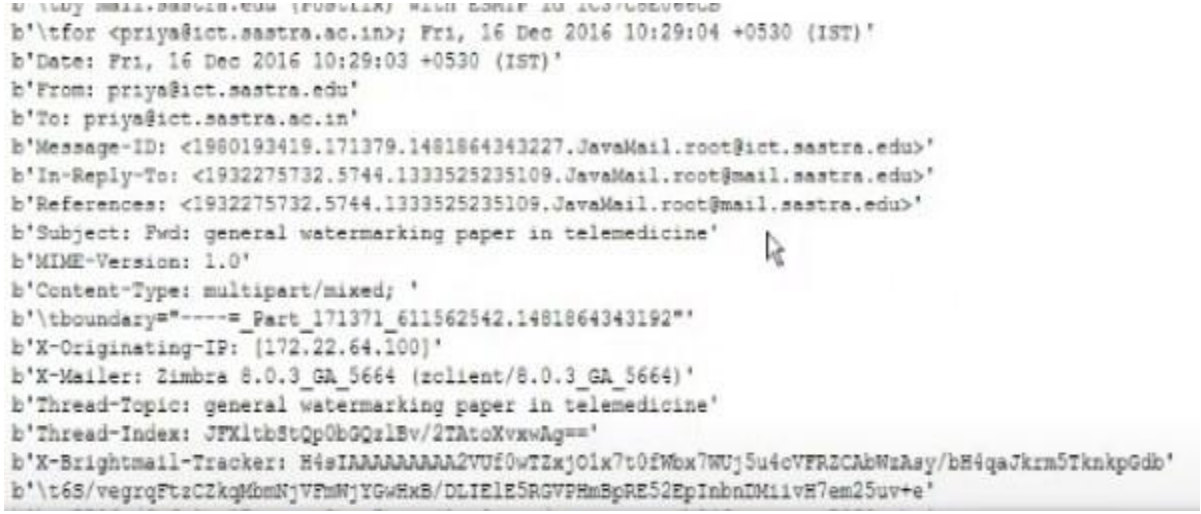
```
p.quit()
```

poplib.pop3 class methods

Method	Description
`user(login)`	Sends the `login` name to the server; awaits reply indicating the server is waiting for user's password
`pass_(passwd)`	Sends *passwd* (after user logs in with `user()`); an exception occurs on login/passwd failure
`stat()`	Returns mailbox status, a 2-tuple (*msg_ct, mbox_siz*): the total message count and total message size, a.k.a. octets
`list([msgnum])`	Superset of `stat()`; returns entire message list from server as a 3-tuple (*rsp, msg_list, rsp_siz*): server response, message list, response message size; if *msgnum* given, return data for that message only
`retr(msgnum)`	Retrieves message *msgnum* from server and sets its "seen" flag; returns a 3-tuple (*rsp, msglines, msgsiz*): server response, all lines of message *msgnum*, and message size in bytes/octets
`dele(msgnum)`	Tag message number *msgnum* for deletion; most servers process deletes upon *quit()*
`quit()`	Logs out, commits changes (e.g., process "seen," "delete" flags, etc.), unlocks mailbox, terminates connection, and then quits

```
from poplib import POP3_SSL
import getpass
import logging

s=POP3_SSL('pop.gmail.com')


s.user("regno@sastra.ac.in")
        b'+OK send PASS'

s.pass_(getpass.getpass('EnterPassword:'))
        EnterPassword:*********
        b'+OK welcome

resp,tot_mail,tsize=s.list()
```

```
id,size=tot_mail[0].split()
r,msg_text,size=s.retr(int(id))
for line in msg_text:
    print(line)
```




```
s.quit()
    b'+OK
```

Receiving email using IMAP4

1. Connect to server
2. Login (if applicable)
3. Make server request
4. Quit

- Module:

```
from imaplib import IMAP4_SSL
```

- Connecting to server:

```
S=IMAP4_SSL('imap.yourdomain.com')
S.login('username''passwd')
```

- Closing the connection

```
S.logout()
```

Imaplib. IMAP4_SSL class methods

Method	Description
`close()`	Closes the current mailbox. If access is not set to read-only, any deleted messages will be discarded.
`fetch(message_set, message_parts)`	Retrieve e-mail messages (or requested parts via *message_parts*) stated by *message_set*.
`login(user, password)`	Logs in *user* by using given *password*.
`logout()`	Logs out from the server.