Introduction Session for Coursework 2

GZ01/3035 Networked Systems

Astrit Zhushi

Slides adopted from Jie Xiong and Georgios Nikolaidis

Department of Computer Science

University College London

Coursework 2

- Part 1 - Manual Recursive Queries (4 points)
  - Understand how recursive queries work in practice
  - Simple but important (helps you later in Part 2)
- Part 2 – Building local nameserver (8 points)
  - Implement your own DNS server with recursive query functionality

Part 1: Manual Recursive Queries

- Walking manually through the steps local DNS server takes
- Utilize CS local nameserver (haig)
- Open console in Linux

```
dig @haig.cs.ucl.ac.uk sipb.mit.edu
```

(get answer in one round)

```
dig @haig.cs.ucl.ac.uk sipb.mit.edu . +norecurse
```

(You need to manually send queries recursively)

Part 2: Ben's Local DNS Server

- Build your own operational DNS server
- Implement DNS recursive lookup algorithm
  - Handle A and CNAME
  - Caching
  - Nothing else for simplicity
- A lot of codes already given
  - Parsing libraries to construct and parse DNS protocol packets

Libraries Provided

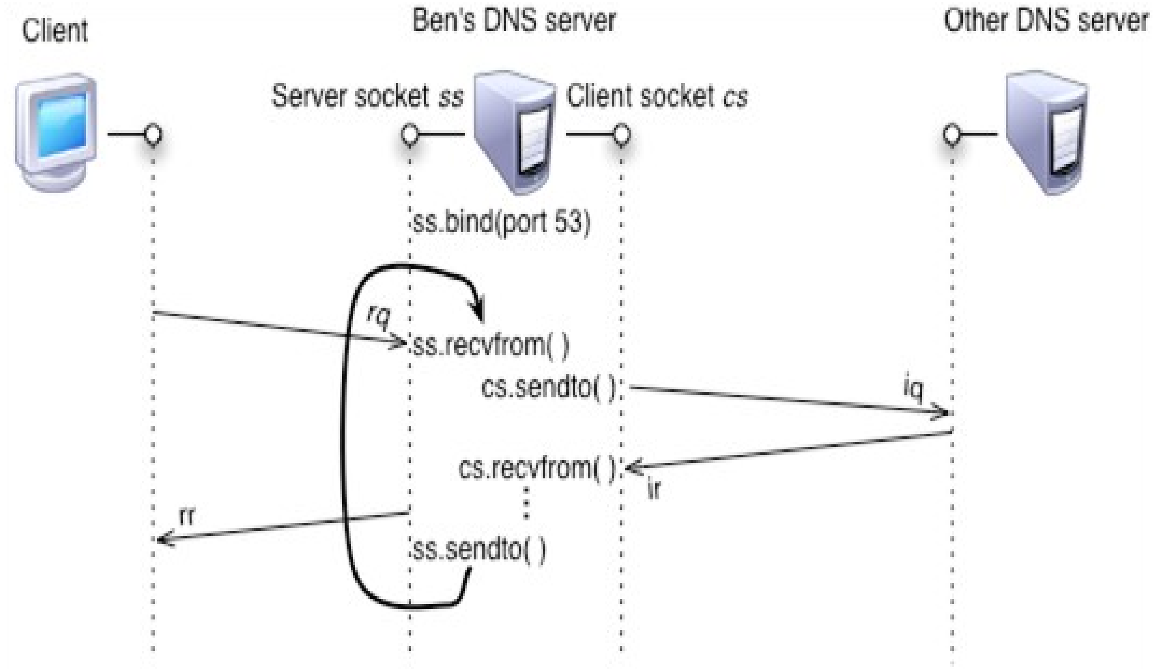
- dnslib (gz01.dnslib)
  - Construct and parse DNS protocol packets
  - *Header*, *QE* and *RR* classes
  - *fromData(data,offset)* method
- inetlib (gz01.inetlib)
  - *DomainName* class implements decompression
- Test suite
  - Provided to help you make sure your algorithm is implemented correctly
  - ./python-wrapper ./test-dns.py ./ncsdns.py

Logging and Debugging

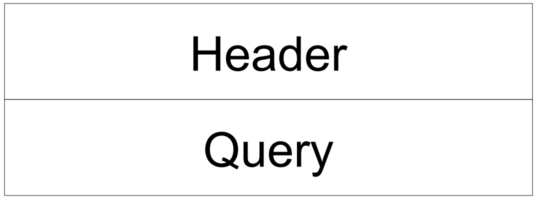
- hexdump(src, length=16) function
  - Human readable HEX and ASCII
- Logs output in console and file *ncsdns.log*
- *logger* object (nscdns.py)
  - logger.critical(string), logger.error(string), ...
- Setting the level of verbosity
  - Modify gz01/util.py as directed by comments

Part 2: Ben's Local DNS Server

- Single-threaded, only one client active at a time
- Two sockets:
  - one for incoming recursive queries (ss);
  - the other for outgoing iterative queries (cs)



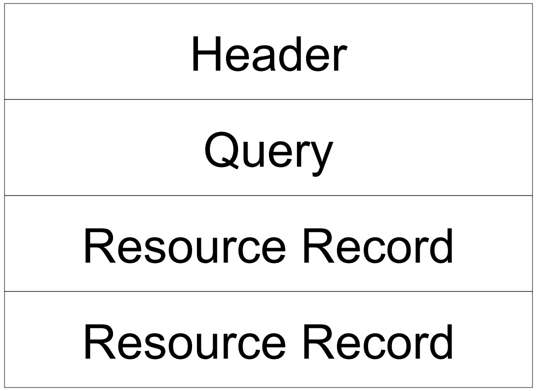
DNS QUERY



Header

Query

DNS REPLY



Header

Query

Resource Record

Resource Record

To perform a query:

```
$ dig @ipaddress -p portnumber domainname
```

Lab Machines

- Can ssh to lab machines
  - Accessible over the Internet
  - Pick one at random
  - *$ ssh username@MACHINENAME.cs.ucl.ac.uk*
    - *MACHINENAME one of the machine names in handout*
  - *$ rlogin MACHINENAME*
  - *$ exit* OR *Ctrl-d* to exit
- Login in person (CS LAB)
- To edit files:
  - *$ vim ncsdns.py*
  - *$ scp ncsdns.py username@MACHINENAME.cs.ucl.ac.uk:~/*
  - *For Windows*
    - *PuTTY, PSCP, Filezilla*

Setting up Python

- Use GZ01 staff-provided Python
- Current version 2.6.2
- Current architectures: sun4, x86_64, i686
- Execute **python-wrapper** instead of python

Python scripts, interpreter

- Conventional file extension **.py**
- Make executable directly and run:

  ```
  $ chmod +x hello.py
  ```

- Interpreter:
  - **python-wrapper**

  ```
  Python 2.6.2 (r262:71600, Sep 26 2009, 14:12:31)
  ```

- To run your code:
  - $ **./python-wrapper ncsdns.py**

Coursework Submission

- Use **handin** program on UNIX CS machines
  - You may **handin** many times, the last submission will be evaluated.
- Part 1
  - Create **cw2-part1.txt** and submit only this text file using **handin**
- Part 2
  - Usually, submit just ***ncsdns.py*** file
  - Remove intermediate python files (ending in .pyc)
- Class policy for late submission applies
  - State the number of late days in the **cw2-part1.txt** file

Help!

- Read the code/documentation
- RFCs (1034,1035) & tutorials online
  - RFC 1034, Section 5.3.3 describes the algorithm
- Moodle Forums
- gz01-staff@cs.ucl.ac.uk
- Office hours