CS 640 Introduction to Computer Networks

Lab 5 Overview

Course Instructor: Dr. Ming Liu

Lab 5 Overview

This lab is split into 2 parts

1. Python-based sender and receiver simulating sliding window protocol for **flow control**.
2. Simple Java-based **DNS server** that performs recursive DNS resolutions, and appends a special annotation if an IP address belongs to an Amazon EC2 region.

You may choose submit only part 1 or only part 2 for partial credit.

Learning Outcomes

After completing this assignment, you should be able to:

- Explain how the sliding window protocol facilitates flow control
- Explain how the domain name system (DNS) works

Implementation: Flow Control

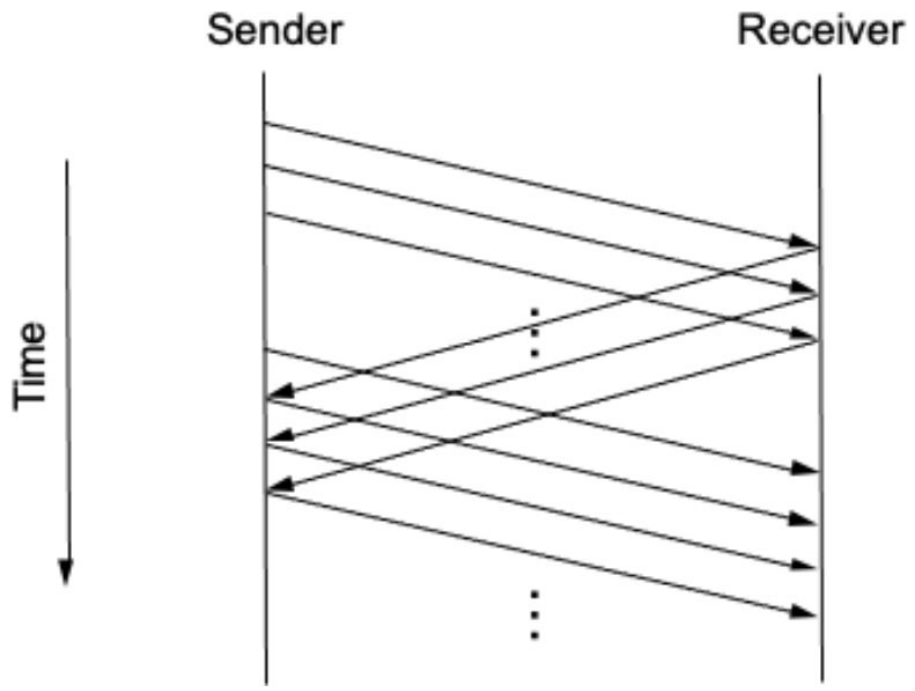

Figure: Sliding window protocol

Sender

- Transmits packets
- Guarantees max in-flight packets
- Retransmits if not ACKed.

Receiver

- Sends cumulative ACKs.

Complete `SWPSender` and `SWPReceiver` class.

Implementation: DNS Server

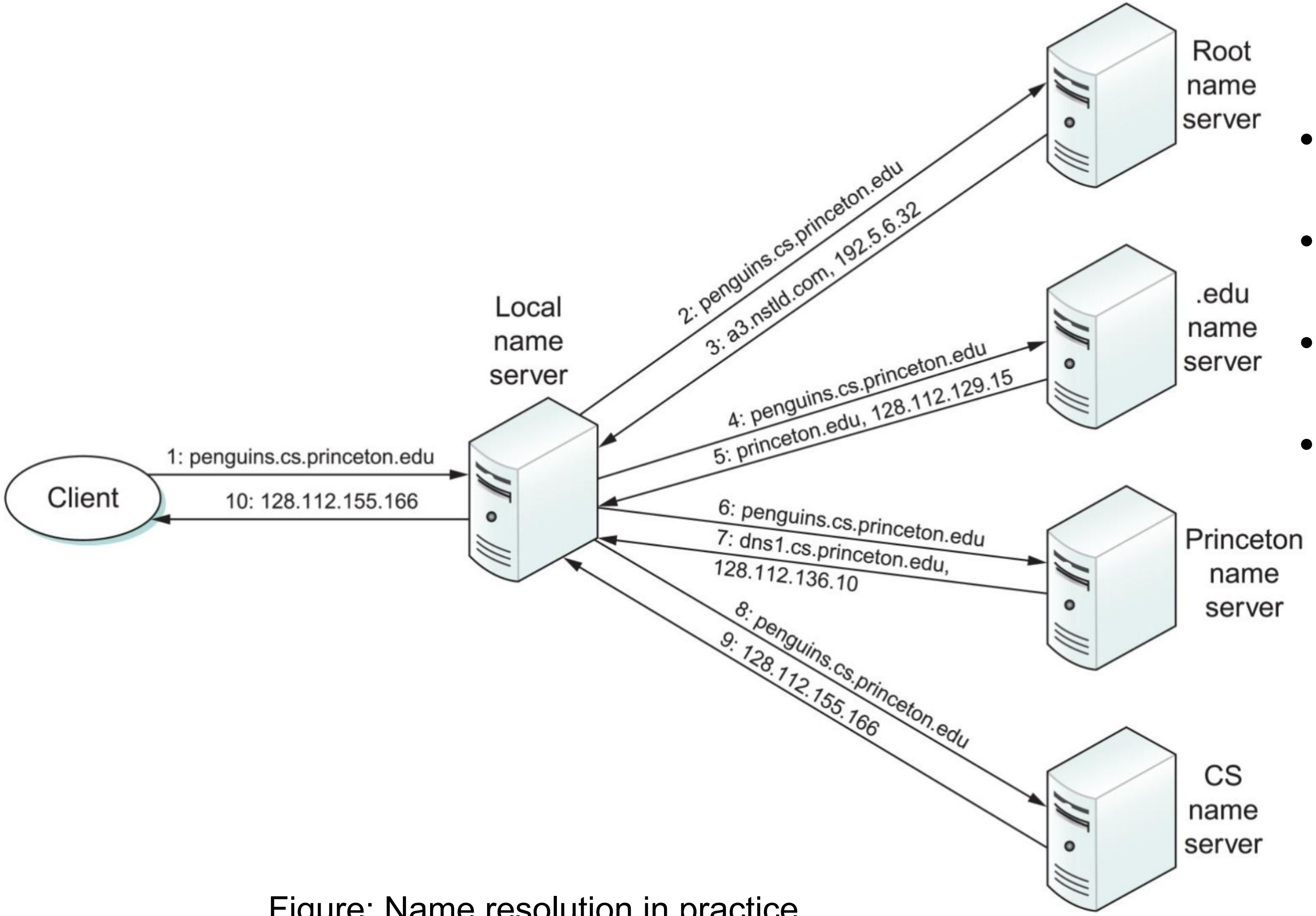
Root name server
.edu name server
Princeton name server
CS name server
Local name server
Client
1: penguins.cs.princeton.edu
2: penguins.cs.princeton.edu
3: a3.nstld.com, 192.5.6.32
4: penguins.cs.princeton.edu
5: princeton.edu, 128.112.129.15
6: penguins.cs.princeton.edu
7: dns1.cs.princeton.edu, 128.112.136.10
8: penguins.cs.princeton.edu
9: 128.112.155.166
10: 128.112.155.166

Figure: Name resolution in practice

DNS Server

- Accepts queries from clients
- Issues queries to other DNS servers
- Makes special annotation for Amazon EC2 IP.
- Does not cache DNS records.

Complete `SimpleDNS` class.

Rubric: Submission

Late policy:

- Upto 30 minutes late — lose 0% of points
- Upto 24 hours late — lose 10% of points
- Upto 48 hours late — lose 30% of points
- Upto 72 hours late — lose 60% of points
- Beyond 72 hours — lose 100% of points

Description	Points	Example Commands	Explanation
Working Makefile for part 2 and correct submission format.	2	make make run make clean	Files submitted as specified in the lab description. “make” should compile program. “make run” should start the SimpleDNS server. “make clean” should remove *.class files.

Rubric: Part 1 Flow Control

Description	Points	Example Commands	Criteria
Both server and client runs *	3	On one terminal: ./fc/server.py -p 8000 On another terminal: ./fc/client.py -p 8000 -h 127.0.0.1	No error message shown No error message shown Whole part might get zero marks if client fails to run. If server fails- - No marks for the testcases with asterisk* - At most half marks for testcases with plus sign+
Server Receives and Decodes data correctly *	3	On client side, type in: helloworld	Server prints out includes exactly the same line: helloworld No score if there are other characters in the line such as b'helloworld'
Server ACKs for data received *	3	Use the output above	Output on rubric document.

Rubric: Part 1 Flow Control

Description	Points	Example Commands	Criteria
Client resends for data not ACKed	6	Kill the server: Repeat ctrl-c until server quits On the client side type in: tobelost	3 points - Client repeats for every second (no score if not repeating with constant 1-second interval) something like: DEBUG: LLP sent: b'D\x00\x00\x00\x02tobelost\n' DEBUG: Sent: DATA 2 b'tobelost\n' 3 point - seq num is the Sent Data in above test case + 1
Client blocks upon full sliding window	6	Kill the client, and restart the client only ./fc/client.py -p 8000 -h 127.0.0.1 first copy to notepad then copy again 7 lines of text. line1 … line7 Paste to client (ctrl-shift-v in typical Linux console)	Only lines 1-5 are repeatedly sent
Overall testing+	9	Restart both sides as the first test case first copy to notepad then copy again 17 lines of text. line1 … line17 Paste to client (ctrl-shift-v in typical Linux console)	3 points - Server prints lines in order and no skipping (can have debug info in between) 3 points - Server prints all lines 3 points - no error/exception and both sides returns to normal

Rubric: Part 2 Simple DNS Server

Description	Points	Example Commands	Criteria
Lookup with recursion works	3	dig -p 8053 @localhost A wisc.edu	dig output has answer with correct answer section. Look for ANSWER SECTION: in dig output.
Lookup for A record includes authority and additional section from the final response	3	dig -p 8053 @localhost A wisc.edu	dig output has answer along with an authority and additional section
Lookup for A record with CNAME induced recursion works	4	dig -p 8053 @localhost A www.pinterest.com	dig output has answer section with A and CNAME(s)
Lookup for CNAME record works	3	dig -p 8053 @localhost CNAME www.pinterest.com	dig output has answer section has type CNAME record
Lookup for NS record provides correct answer	3	dig -p 8053 @localhost NS wisc.edu	dig output has answer section has type NS records
Lookup for domain hosted in EC2 provides correct TXT record	4	dig -p 8053 @localhost A www.code.org	dig output has answer with IP in EC2 (with TXT record)
Non-recursive lookups are not resolved recursively	4	dig +norecurse -p 8053 @localhost A wisc.edu	dig output has no answer section and authority section with *.edu-servers.net & additional section with IP for those
Lookup for A record when an intermediate response contains only authority section and no additional section works (your code needs to recursively request A record for one of the name servers in the authority section)	4	dig -p 8053 @localhost A www.youtube.com	dig output (Answer section) has a CNAME and many A records

Thank You. Questions?