Assignment 1

Tweet Inquirer

Due Date: Monday, February 13, 2017 @ 23:55

ECE 4564 - Network Application Design

Learning Objectives

Client Sockets

- socket()
- connect()
- send()
- recv()

Server Sockets

- bind()
- listen()
- accept()

Twitter API

WolframAlpha API

Encapsulation/Decapsulation

Checksums

Sockets

Sockets

- Low level API for opening a connection to another device and exchanging data
- Socket protocols: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)
- Protocols act as a transport mechanism: bits travel from sender to receiver
- Higher level applications need to establish what data to transmit and how data is ordered

TCP is a reliable mechanism

- TCP will ensure that the bits sent from sender to receiver arrive in order and without error or TCP will notify the user of a problem

UDP is a best effort service

- Messages may not get to destination and the sender won’t be warned of a lost message

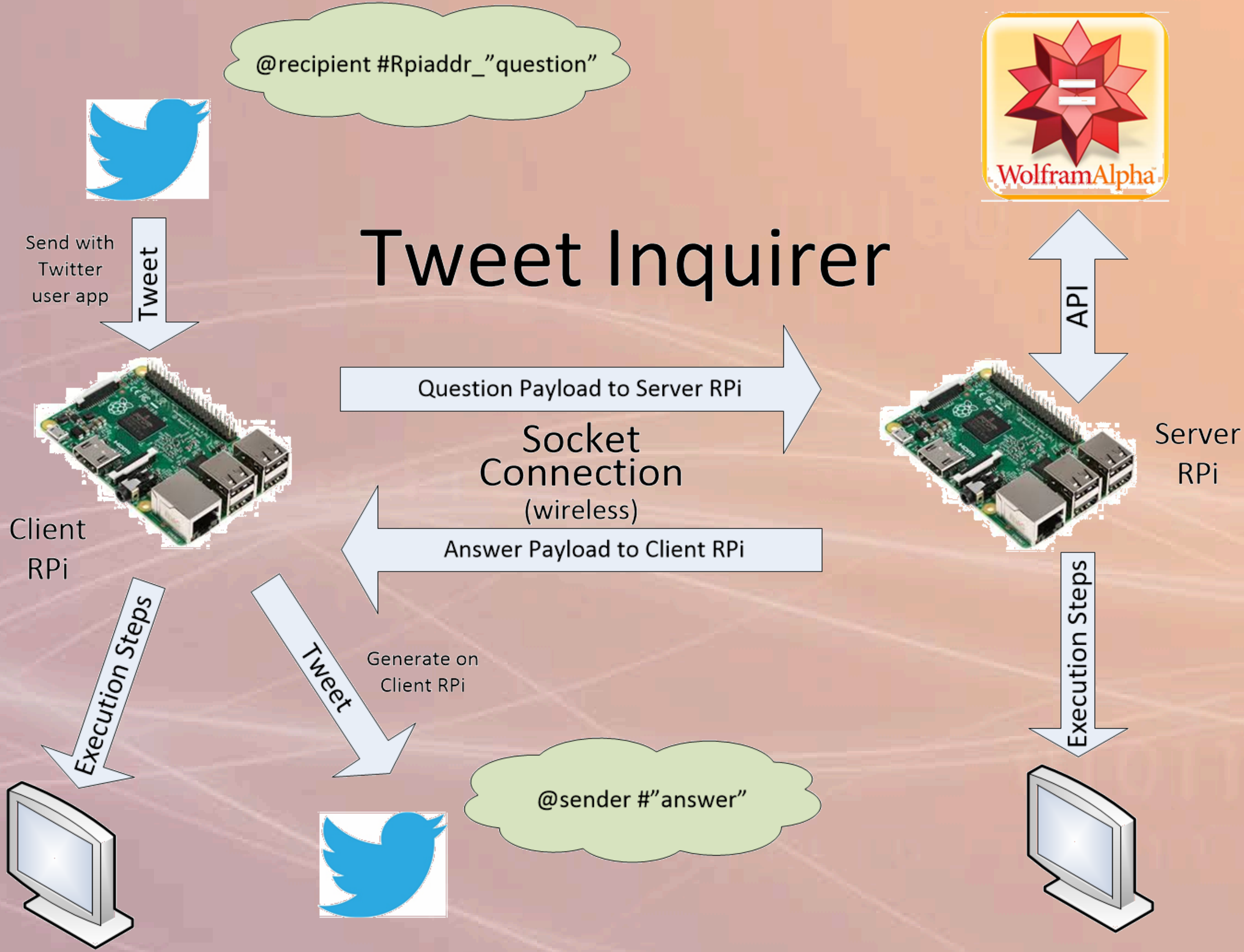
Assignment Overview

Assignment 1 is a text-based question and answer system using WolframAlpha's computational knowledge engine. Questions and answers of your Q&A system are expressed as Twitter Tweets

The system uses two Rpi's following the client/server model discussed in class. The server is iterative and connection-oriented. Communication between client and server is handled via sockets.

The client Rpi receives the Tweet containing the question. The client builds and sends a question "payload" to the server Rpi via sockets. The server sends the question to the WolframAlpha engine and receives the answer. The server builds and sends an answer "payload" back to the client Rpi. The client tweets the answer.

System Overview


@recipient #Rpiaddr_"question"
Send with Twitter user app
Tweet
Tweet Inquirer
API
Question Payload to Server RPi
Socket Connection (wireless)
Answer Payload to Client RPi
Client RPi
Server RPi
WolframAlpha
Execution Steps
Tweet
Generate on Client RPi
@sender #"answer"
Execution Steps

Client / Server Model

Client Rpi

- Captures “question” Tweet
- Builds question payload
- Sends question payload to server Rpi via socket interface
- Receives and parses answer payload
- Sends “answer” Tweet

Server Rpi

- Receives question payload from client Rpi
- Parses question payload
- Sends question to WolframAlpha engine via API call
- Receives answer from WolframAlpha engine
- Builds answer payload
- Sends answer payload to client Rpi via socket interface

Client Requirements

Client Rpi

- Captures a Twitter status object (Tweet)
- Parses the Tweet
  - Address and port of target Raspberry Pi (Rpi)
  - “question”
- Computes checksum of “question”
- Builds question payload (see payload slide)
- Sends question payload to server via socket call
- Waits for answer payload
- Deconstructs answer payload
  - Verifies checksum
- Sends a Tweet containing “answer” to original Twitter account

Status Object Format

Format of “question” Tweet:

@Twitter_User_Name   #server_host_addr:port_”question”

Example:

@tomswift #192.168.1.32:23456_”What is a Hokie?”

Format of “answer” Tweet:

@Twitter_User_Name   #”answer”

Example:

@tomswift #”a student, alumnus, or supporter of Virginia Tech University”

Server Requirements

Server Rpi

- Waits for question payload from client
- Receives and deconstructs question payload
  - Verifies checksum
- Sends question to WolframAlpha engine via API call
- Receives answer from WolframAlpha engine
- Builds answer payload
  - Generates checksum on answer
  - Assembles payload
- Sends answer payload to client via socket call

Payloads

Question Payload (Python tuple):

- Question text
- MD5 hash of question text

Answer Payload (Python tuple):

- Answer text
- MD5 hash of answer text

Required to Pickle payload when sending over socket connection

Tuples

Immutable list

Syntax: (elem1, elem2, …)

A tuple cannot be changed.

Example:

```
>>> tuple1 = (1, 5, 10)
>>> tuple1[2] = 2
Traceback (most recent call last):
    File "<pyshell#136>", line 1, in ?
     tuple1[2] = 2
TypeError: object doesn't support item assignment
```

Pickle

Pickling is the native form of serialization included in Python’s Standard Library.

A *pickle* stores the contents of a Python data structure that can be reconstructed later on a different machine.

```
>>> import pickle
>>> pickle.dumps([5,6,7])
'(lp0\nI5\naI6\naI7\na.'
>>> pickle.loads('(lp0\nI5\naI6\naI7\na.')
[5, 6, 7]
>>>
```

https://docs.python.org/3.5/library/pickle.html

Twitter

- Requires a Twiter account
- Applications accessing Twitter require an API key. This key is obtained by first signing up for a development account here. The API key can be created once logged into your account.
- Beware of API rate limits
- Implement as Streaming API
- Suggest implementing with Tweepy

WolframAlpha

Applications accessing the WolframAlpha engine require an API key. This key is obtained by first signing up for a development account here. The API key can be created once logged into your account.

Note: Access to this service is limited to 2000 non-commercial API calls per month.

Given this, each team member should set up their own development account and generate an API key. This provides 3 to 4 keys per team allowing 6000 to 8000 service queries. Please to not intentionally try to exceed these limits.

The Validation Process

1. Each team will get 15 minutes slot for validation. This timing deadline has to be followed strictly. This includes the time for setting up your hardware and presenting your assignment. Sign-up for slots will be available on Canvas soon.

2. Each team member **must** be present during the validation. Inform Dr. Plymale and GTA's prior to your validation time by email in case you won't be able to make it. Team members absent from validation will receive a 10 point reduction in score.

3. Each team should bring their hardware kits and setup for validation (Raspberry Pi, power adapter & related hardware for that HW). Kindly come 10 minutes before your appointed time slot.

4. During validation, you will be provided with sets of monitors, keyboards, mouse, HDMI cables and a LAN cable if needed.

Grading

- Report: 10%
  - 1 to 2 pages
  - Single-spaced
  - Submit as PDF

- Validation with GTAs: 90%
  - Parsing question from Twitter feed correctly - 10 points
  - Payload checksum compute and validation - 10 points
  - Passing question and generating answers from Wolfram API - 20 points
  - Communicating Q&A via socket - 20 points
  - Tweeting Answer(s) correctly - 10 points
  - All of the above over Eduroam - 20 points

Grading

- There may be multiple answers to a question, Tweeting all of the answers is considered a correct demonstration

- Missing answers will result in partial scoring

- Should demonstrate application over Eduroam, since your validations will take place in the Durham building, make sure your application works over Eduroam in the Durham building

Report

You must document the design, and outcomes in a brief written report (1-2 pages). One report submitted by each team. Your report should contain the following items.

- At the top of the first page of your report, include: your names (as recorded by the university); your email address; and the assignment name (e.g., “ECE 4564, Assignment 1”). Do not include your Virginia Tech ID number or your social security number.

- The body of the report must contain the following sections. Use section numbers and headings to organize your report.
  - Section 1 – Objectives: Provide a description of the design objectives and general approach to the design.
  - Section 2 – Team member responsibilities
  - Section 3 – Conclusions: Discuss the outcome of your design and any problems encountered and resolutions; what you learned by doing this project; and any experiences that were particularly good or bad.

Python Style

Follow style guide PEP0008 when writing and commenting your code

https://www.python.org/dev/peps/pep-0008/

Coding for all assignments in Python 3

What You Turn In

All assignments must be submitted through Canvas, no later than the due date of 2017 Feb 13 @ 23:55

Your assignment should be a single zip or tarball (i.e. tar.gz, tar.bz) which contains the following:

- All source code for this assignment
  - Python code running on client and server Rpi's
    - Client code identified as "client.py"
    - Server code identified as "server.py"

- A ReadMe file describing client and server initialization procedures and any extra libraries used.

- Report (PDF file)

Assignment References

The official documentation from Python

- The Official Python Documentation. https://docs.python.org/3.4/index.html

A short and quick introduction to Python when programming on a Raspberry π (Ch. 2-6)

- Bradbury, Alex, and Ben Everard. Learning Python with Raspberry Pi. John Wiley & Sons, 2014.

A Python reference guide for programmers who understand software engineering/computer science concepts such as object-oriented programming*

- Martelli, Alex. Python in a Nutshell. 2nd Ed. " O'Reilly Media, Inc.", 2006. http://proquest.safaribooksonline.com/book/programming/python/0596100469

A more comprehensive book on Python that includes examples*

- Lutz, Mark. Programming python. 4th Ed. " O'Reilly Media, Inc.", 2010. http://proquest.safaribooksonline.com/book/programming/python/9781449398712

Assignment References

“Beej's Guide to Network Programming Using Internet Sockets” (PDF)

“Foundations of Python Network Programming”, 2nd Ed. (Full text – VT Library)

The official WolframAlpha API

- http://products.wolframalpha.com/api/

The official Twitter Developers documentation

- https://dev.twitter.com/

A Twitter API

- Tweepy

Streaming Tweets from Twitter

- Tutorial

Academic Integrity

- For this assignment, it is expected that a team's work is their own
- The code you turn in must be your own (i.e. you need to have written your assignment)
- You are allowed to copy and paste example code from other websites, but you must include a comment in your code that attributes the website you copied the code from (i.e. original author's name and URL to the original code)
- You can discuss the assignment with other teams
- However, you cannot just tell another team the answer to a particular problem

Final Thoughts

In many cases, engineers are expected to just make things work given a particular design constraint (e.g. software package to use or are limited to a particular hardware platform).

You will likely run into similar situations in this class while designing and implementing your assignments and project.

.

When you're stuck, try searching online for a solution. Many times others have tried something similar and documented their experiences for others to learn and benefit from

Do not publically post answers to assignments, or your code until after the assignment due date.

Contact your instructor or GTA's as soon as you encounter a problem you're unable to solve. Don't wait to begin right before the assignment is due.