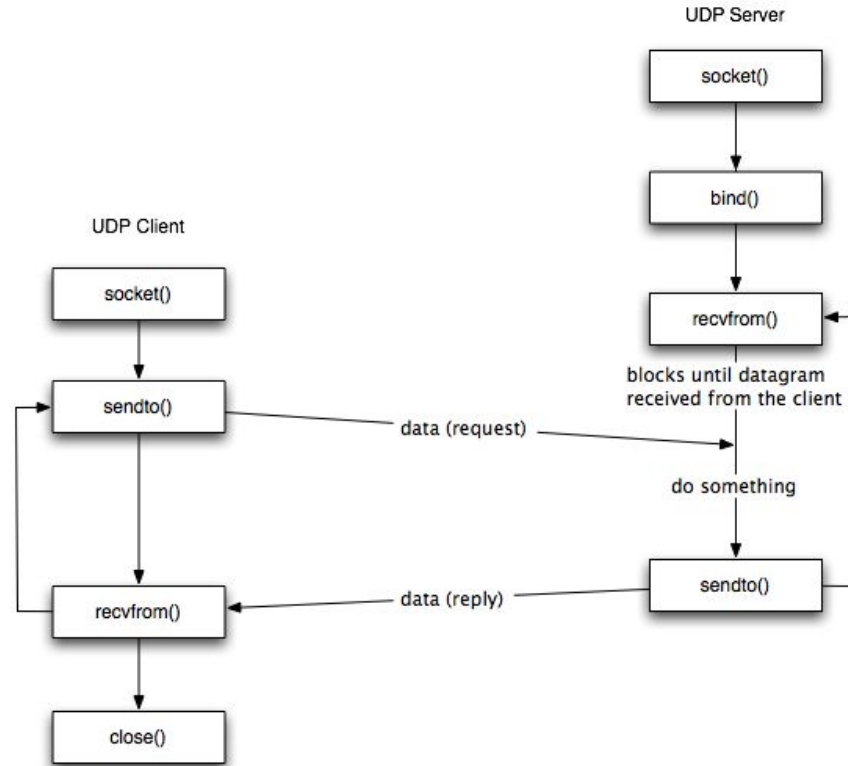


# Recitation 2

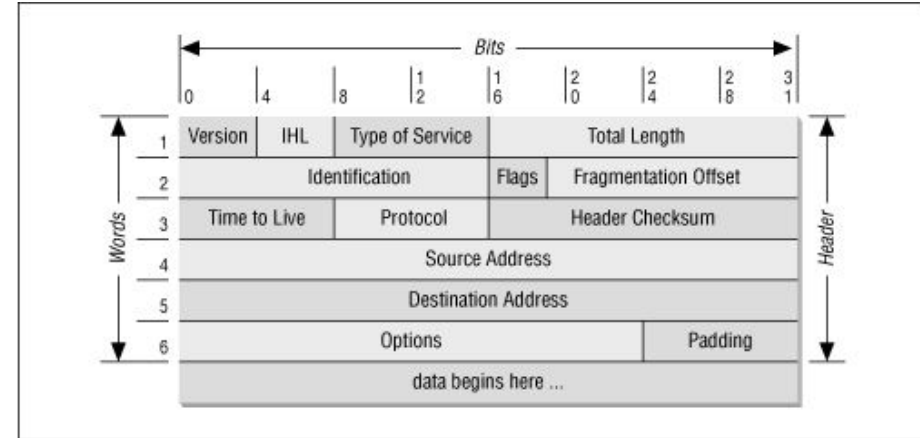
## Internet Technology (Section 01)

# Socket Workflow (UDP)



# Network Packets

- A packet is a **formatted** chunk of data
  - Typically holds information for **control** and **data**



# Structs in Python

- Pack formatted binary data
  - Needs a format string

<https://docs.python.org/3/library/struct.html>

```
import struct
'''
!: Byte order: network (= big-endian)
I: unsigned int
?: bool
'''
fs = "!3I?"
b = struct.pack(fs, 123, 456, 789, True)
print(b)
print(struct.calcsize(fs))
arr = struct.unpack(fs, b)
print(arr)
```

# A Toy Example - Student Class Registration

Server

```
import socket, struct
HOST, PORT="127.0.0.1", 1234
sock = socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)
sock.setsockopt(socket.SOL_SOCKET,
socket.SO_REUSEADDR, 1)
sock.bind((HOST,PORT))
roster = {}
while True:
    pkt, addr = sock.recvfrom(1024)
    if not pkt: continue
    id,course,val = struct.unpack("!II?",pkt)
    if val: #add course for student
        if id in roster:
            roster[id] |= {course} #set union
        else: roster[id] = {course}
    else: #remove course for student
        if id in roster:
            try: roster[id].remove(course)
            except KeyError: pass
    sock.sendto(bytes(str(roster),'utf-8'),addr)
sock.close()
```

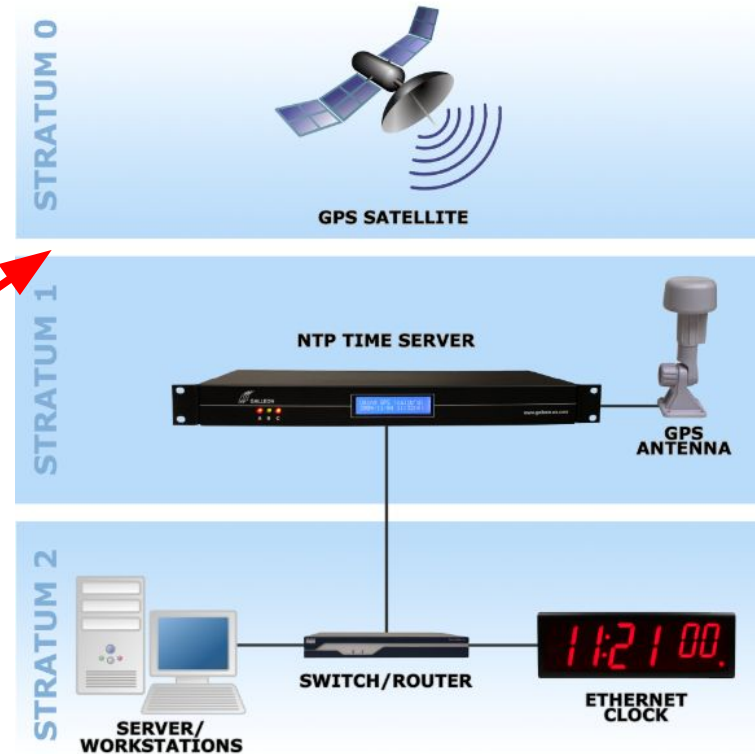
Client

```
import socket, struct
HOST="127.0.0.1"
PORT=1234
ADDR = (HOST,PORT)
sock = socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)
fs = "!II?" #(Student ID, Course Number,
Enroll Boolean)
pkt = struct.pack(fs,186395729,352,True)
sock.sendto(pkt,ADDR)
data,addr = sock.recvfrom(2048)
print(data)
sock.close()
```

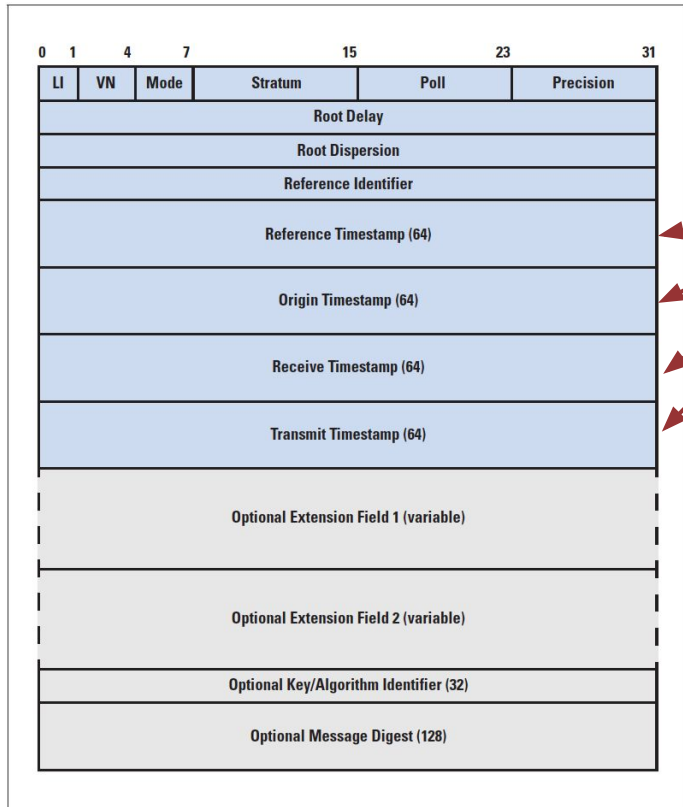
# Network Time Protocol

- An application layer protocol for computer clock synchronization over networks
- Uses **UDP** on port **123**
- Each layer of time sources is called a **stratum**

**Try calling (202)-762-1401**



# NTP Packet



Epoch: January 1st, 1900  
*Y2038 problem*

**NTP timestamp format (64 bits) :**

Seconds (32-unsigned) | Fraction (32)

Seconds and Fractions since 01.01.1900

Seconds: 0x00 | 0x00 | 0x00 | 0x07 → **7**

Fraction: 0x00 | 0x00 | 0x00 | 0x03 →  **$3/2^{32}$**

7.0000000006985

# Unix Time Format

- Most used time format in computing
- Epoch: January 1st, 1970
  - *Y2038 problem*
- 32-bit signed integer

Unix  $\longleftrightarrow$  NTP

- 2,208,988,800 seconds from 1900 to 1970

$$\text{Unix} = \text{NTP} - 2208988800$$

$$\text{NTP} = \text{Unix} + 2208988800$$

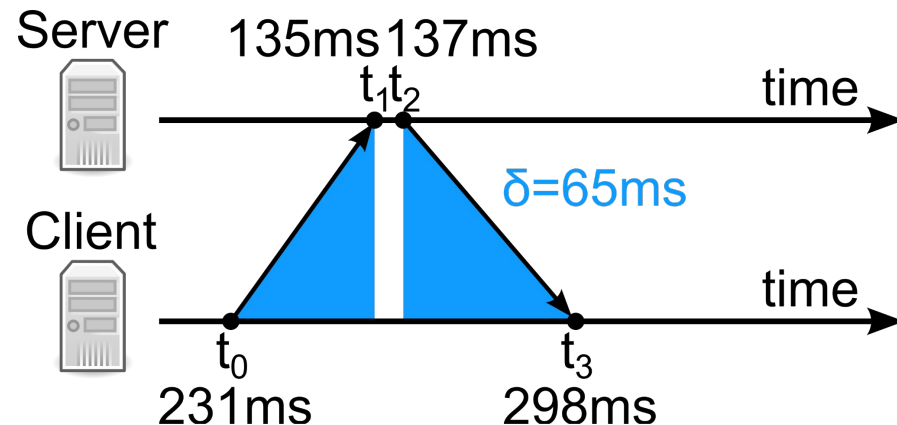


# NTP Clock Synchronisation Algorithm

- $\text{delay} = (T3 - T0) - (T2 - T1)$   
 -> **65ms**
- $\text{offset} = ((T1 - T0) + (T2 - T3)) / 2$   
 -> **-128.5ms**

## Simple Clock Update

$\text{time} = \text{time} + \text{offset}$



# Project 1

```
#!/usr/bin/env python
'''
CS352 Assignment 1: Network Time Protocol
You can work with 1 other CS352 student
DO NOT CHANGE ANY OF THE FUNCTION SIGNATURES BELOW
'''
from socket import socket, AF_INET, SOCK_DGRAM
import struct
from datetime import datetime
def getNTPTimeValue(server="time.apple.com", port=123) -> (bytes, float, float):
    return (pkt, T1, T4)
def ntpPktToRTTandOffset(pkt: bytes, T1: float, T4: float) -> (float, float):
    return (rtt, offset)
def getCurrentTime(server="time.apple.com", port=123, iters=20) -> float:
    return currentTime
if name == " main ":
    print(getCurrentTime())
```

# Extra Slides

Common time reference identifiers (refid) codes

Refid <sup>[32]</sup> ↕	Clock Source ↕
GOES	Geosynchronous Orbit Environment Satellite
GPS	<a href="#">Global Positioning System</a>
GAL	<a href="#">Galileo</a> Positioning System
PPS	Generic pulse-per-second
IRIG	Inter-Range Instrumentation Group
WWVB	LF Radio <a href="#">WWVB</a> Fort Collins, Colorado 60 kHz
DCF	LF Radio <a href="#">DCF77</a> Mainflingen, DE 77.5 kHz
HBG	LF Radio <a href="#">HBG</a> Prangins, HB 75 kHz (ceased operation)
MSF	LF Radio <a href="#">MSF</a> Anthorn, UK 60 kHz
JJY	LF Radio <a href="#">JJY</a> Fukushima, JP 40 kHz, Saga, JP 60 kHz
LORC	MF Radio <a href="#">Loran-C</a> station, 100 kHz
TDF	<a href="#">MF Radio Allouis</a> , FR 162 kHz
CHU	HF Radio <a href="#">CHU</a> Ottawa, Ontario
WWV	HF Radio <a href="#">WWV</a> Fort Collins, Colorado
WWVH	HF Radio <a href="#">WWVH</a> Kauai, Hawaii
NIST	<a href="#">NIST</a> telephone modem
ACTS	NIST telephone modem
USNO	USNO telephone modem
PTB	German PTB time standard telephone modem
MRS	(Informal) Multi Reference Sources
GOOG	(Unofficial) Google Refid used by google NTP servers as time4.google.com