

CSE 461

# Section 1: UDP, TCP, and addresses

# Addressing

- Project 0 requires sending your own IP address to another client
- Problem: getting own IP address can be hard
- How can we do it?



# Getting own IP address: hostname method

## Steps

- Get the computer's host name
- Resolve it into an IP address

## Python example

```
import socket  
name = socket.gethostname()  
ip = socket.gethostbyname(name)
```

- Sometimes doesn't work depending on hostname configuration and/or will just return local host (127.0.0.1)



# Getting own IP address: connect to server method

## Steps

- Create a socket
- Connect to known server on internet
- Get socket address

## Python example

```
import socket  
  
s = socket.socket(socket.AF_INET,  
socket.SOCK_DGRAM)  
  
s.connect(('gmail.com',80))  
  
ip = s.getsockname()[0]
```

- Useful when previous method doesn't work
- Similar methods exist for other languages as well

# Other Methods

- `getaddrinfo()`
  - E.g., `socket.getaddrinfo(name, o)`
  - Can return other network interfaces that you don't want (e.g., IPv6)
- Querying outside URL
  - E.g., `urllib2.urlopen('http://abstract.cs.washington.edu/~zahorjan/ip.cgi').read()`
  - Best method (if you have a dedicated server to tell you your IP)



# Ports

- Addresses specific to applications/services on a system
- 16-bit numbers (from 0 to 65535)



# Well-Known Ports

- Many applications/services have designated ports
- Examples:
  - ftp: 21
  - ssh: 22
  - telnet: 23
  - http: 80
- Ports from 0 to 1023 are “well-known ports” (don’t use them for protocols you make up!)
- Can see a list of your system’s well known ports in /etc/services (Linux/Unix)



# Hostnames map to IP addresses

- Hosts contact DNS (Domain Name System) servers to get IP address of a given name
- E.g., 'www.gmail.com' maps to 173.194.33.118
- nslookup demo