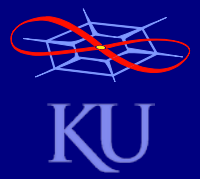
Communication Networks Laboratory
The University of Kansas EECS 780
Introduction to Socket Programming

Mohammed J.F. Alenazi, Egemen K. Çetinkaya,
and James P.G. Sterbenz



Department of Electrical Engineering & Computer Science
Information Technology & Telecommunications Research Center
The University of Kansas

ekc@ittc.ku.edu
jpgs@eecs.ku.edu

http://www.ittc.ku.edu/~jpgs/courses/nets

Socket Programming
Outline

L2.1 Motivation and overview
L2.2 Socket programming stages
L2.3 Socket programming examples
L2.4 Socket programming assignment

Socket Programming
Motivation and Overview

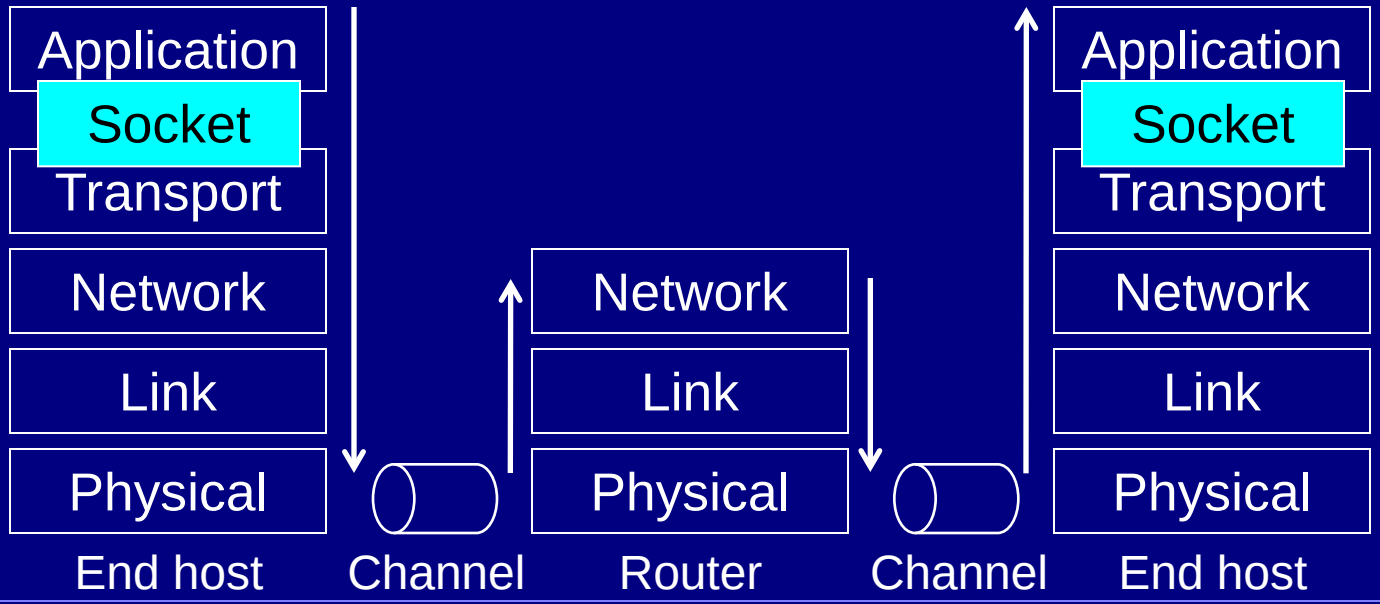
L2.1 Motivation and overview
L2.2 Socket programming stages
L2.3 Socket programming examples
L2.4 Socket programming assignment

Motivation and Overview
Socket Programming and Applications

- Also called client/server application development
- Introduced in 4.1 BSD in 1982
- Network application implementations
  - standard network application
    - based on RFCs
    - programs conforms rules
    - port numbers should be implemented per RFC
  - proprietary network application
    - they don't conform RFCs
    - code will not interoperate
    - don't use standard RFC well-known port numbers (0-1023)

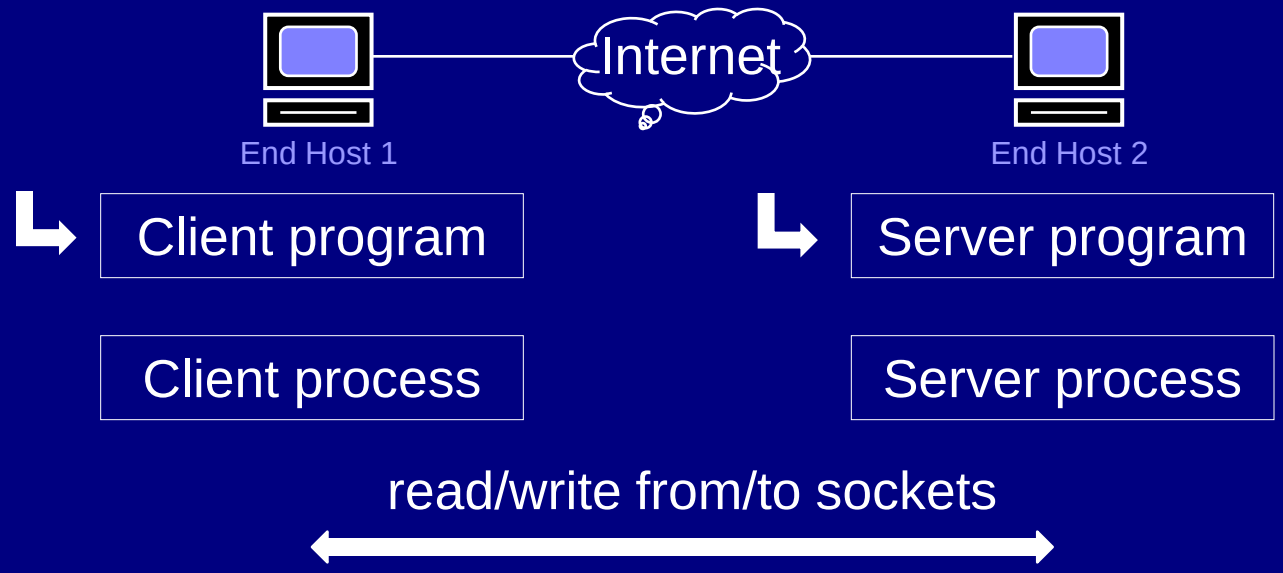
Motivation and Overview
E2E Application Data Flow and Sockets

- Application process send messages via sockets
- Application process is controlled by the developer
- TL (TCP,UDP) is controlled by the OS



Motivation and Overview
Sockets and Processes

- Socket is a method for Inter Process Communication
- Processes are created via client/server programs
- IPC can be done on a single host as well



Socket Programming
Socket Programming Stages

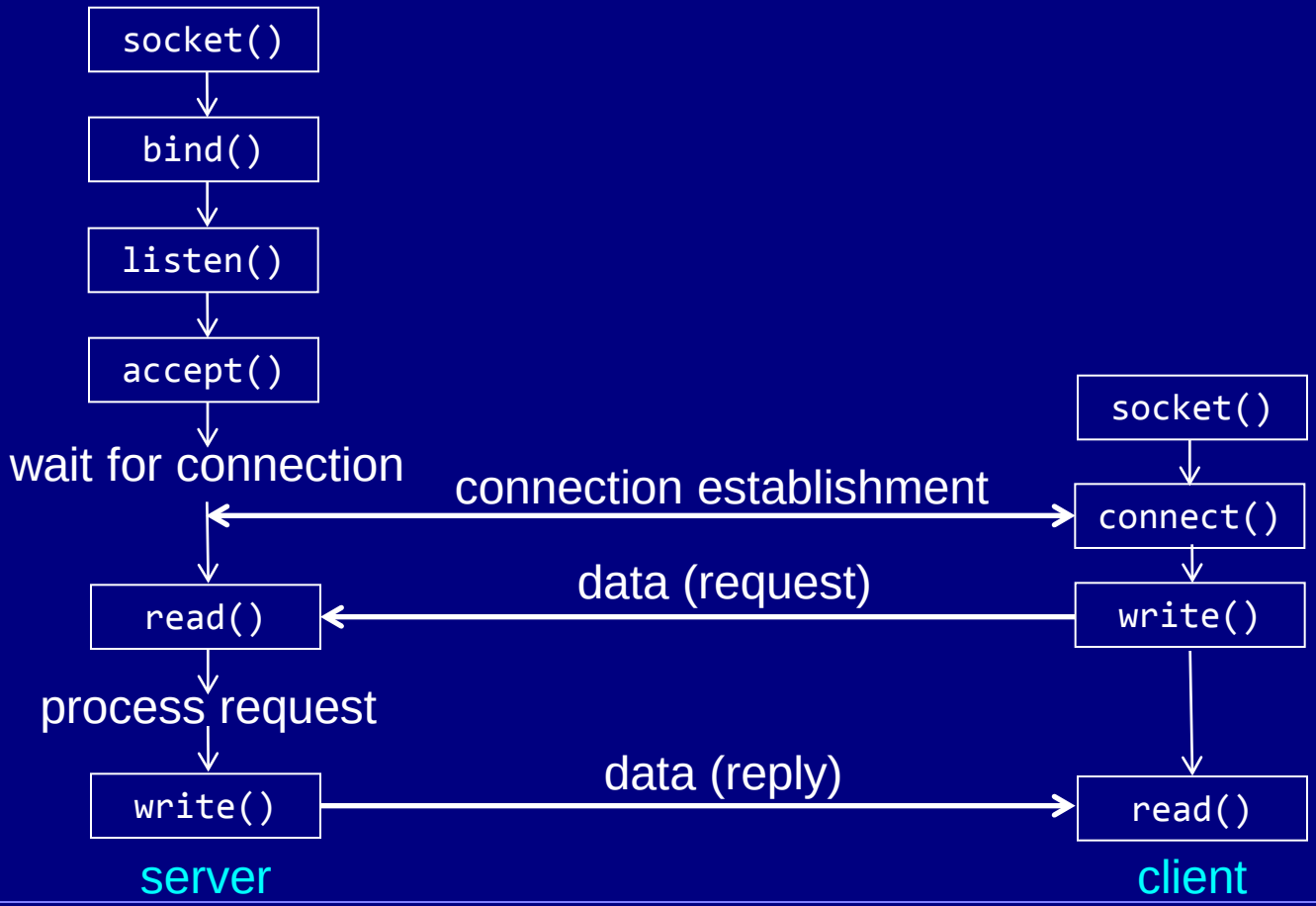
Socket Programming Stages
Planning Phase

1. Developer decides programming language and OS
   - Python , C, Java etc. and UNIX, Linux, MS etc.
2. Developer should decide:
   - to run the application on TCP
     - TCP is connection oriented, reliable byte stream channel
   - to run the application on UDP
     - UDP is connectionless service, best effort, no guarantee
   - skip transport layer to run the application
     - for hop nodes: e.g. ICMP
     - also called raw sockets
3. Developer implements the code

Socket Programming Stages
Socket Programming with TCP

- In order to establish connect. between client & server
- Server process
  - has to be ready to respond client's requests
  - server has to have a welcoming socket
- Client process
  - creates socket
  - specifies the destination
  - 3-way handshake occurs
    - client invokes server's `welcomeSocket accept()` method
    - server responds this by creating dedicated `connectionSocket`
    - TCP establishes pipe betw. `connectionSocket-clientSocket`

Socket Programming Stages
Connection-Oriented Flow Diagram

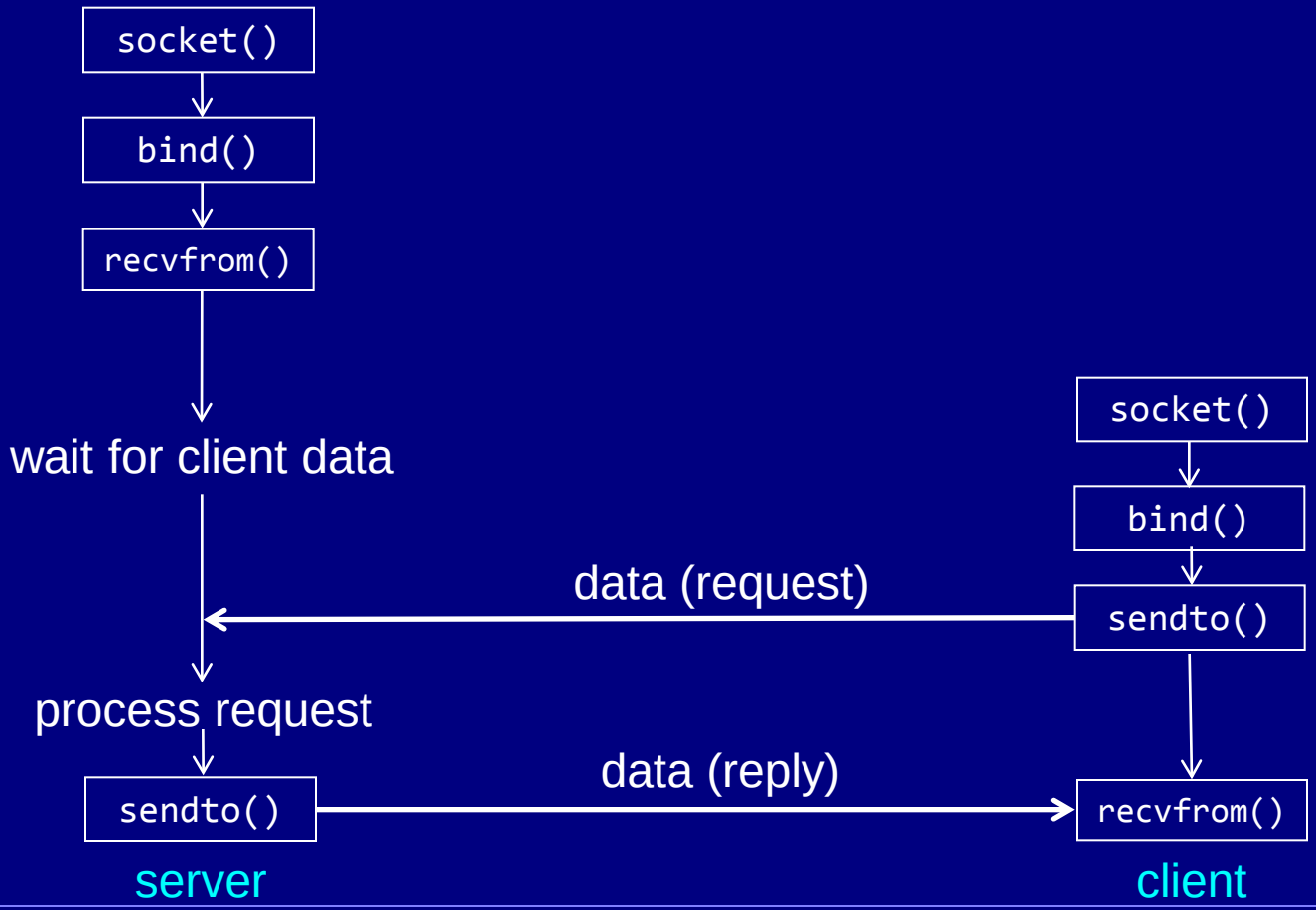


[Stevens-1990]

Socket Programming Stages
Socket Programming with UDP

- Connectionless transport between client & server
  - there is no initial handshaking
  - unlike TCP client can be started first
- Client attaches destination address to each packet
- Client process
  - creates `clientSocket` of type `DatagramSocket`
  - in TCP `clientSocket` is of type `Socket`
- Server process
  - creates `serverSocket` of type `DatagramSocket`
  - there is no `welcomeSocket` as in TCP

Socket Programming Stages
Connectionless Flow Diagram



[Stevens-1990]

Socket Programming
Socket Programming Examples

Socket Programming Examples
Python TCP Client

```
import socket
TCP_IP = '127.0.0.1'
TCP_PORT = 5005
BUFFER_SIZE = 1024
MESSAGE = "Hello, World!"

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((TCP_IP, TCP_PORT))
s.send(MESSAGE)
data = s.recv(BUFFER_SIZE)
s.close()

print "received data:", data
```

Socket Programming Examples
Python TCP Server

```
import socket
TCP_IP = '127.0.0.1'
TCP_PORT = 5005
BUFFER_SIZE = 1024
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((TCP_IP, TCP_PORT))
s.listen(1)

conn, addr = s.accept()
print 'Connection address:', addr
while 1:
    data = conn.recv(BUFFER_SIZE)
    if not data: break
    print "received data:", data
    conn.send(data)  # echo
conn.close()
```

Socket Programming Examples
Python UDP Client

```
import socket

UDP_IP = "127.0.0.1"
UDP_PORT = 5005
MESSAGE = "Hello, World!"

print "UDP target IP:", UDP_IP
print "UDP target port:", UDP_PORT
print "message:", MESSAGE

sock = socket.socket(socket.AF_INET, # Internet
                     socket.SOCK_DGRAM) # UDP
sock.sendto(MESSAGE, (UDP_IP, UDP_PORT))
```

Socket Programming Examples
Python UDP Server

```
import socket

UDP_IP = "127.0.0.1"
UDP_PORT = 5005

sock = socket.socket(socket.AF_INET, # Internet
                     socket.SOCK_DGRAM) # UDP
sock.bind((UDP_IP, UDP_PORT))

while True:
    data, addr = sock.recvfrom(1024) # buffer size is 1024 bytes
    print "received message:", data
```

Socket Programming
Lab Report Submission Requirement

L2.1 Motivation and overview
L2.2 Socket programming stages
L2.3 Socket programming examples
L2.4 Socket programming assignment

Socket Programming Assignment
Program Output Requirement

- Create an application that will
  - convert lowercase letters to uppercase
    - e.g. [a…z] to [A…Z]
    - the code will not change any special characters
    - e.g. &*!
  - if the character is in uppercase , the program will not alter it
- Create socket API both for
  - reliable-byte stream
  - datagram services
- Must take the server address and port from the CLI

Socket Programming Assignment
Example for UDP

- Server side:

  python LastName_UDP_server.py 5050
- Client side:

  python LastName_UDP_server.py 127.0.0.1 5050
  Please enter the statement: eecs780
  return text from the server: EECS780
  Please enter the statement:

- The program should keep asking the user until it forced to exit,i.e, ctrl+c

Socket Programming Assignment
Questions to Answer

- What are the example applications that use TCP and UDP? Give two example for each protocol
- What are the port numbers those applications use?
  - use examples from well-known port number range
- Can you use those port numbers in the application you developed? Why or why not? Explain.
- In the application you developed how many simultaneous client/server connections are possible for each transport protocol? Are the port numbers same for a given protocol for each connection?
  - support you answer with `netstat` command output

Socket Programming Assignment
Extra Credit

- For client/server programs, print local and foreign address using functions
- What is the largest datagram size can you send and receive using UDP socket? Verify it experimentally

Socket Programming Assignment
Submission Requirements[1]

- Follow instructions on submission requirements page
  - `http://www.ittc.ku.edu/~jpgs/courses/homework.html`
- You need to submit:
  - client file for reliable byte-stream service
  - server file for reliable byte-stream service
  - client file for datagram service
  - server file for datagram service
  - PDF file for additional questions
- Use Python

Socket Programming Assignment
Submission Requirements[2]

- Folder name: LastName_socket_programming
- Folder contains:
  - LastName_TCP_server.py
  - LastName_TCP_client.py
  - LastName_UDP_server.py
  - LastName_UDP_client.py
  - LastName_report.pdf
- Subject: "EECS780 – programming 1"
- attachment: LastName_socket_programming.zip
  - zip the whole folder

Socket Programming
Acknowledgements

Some material in these foils comes from the textbook supplementary materials:

- Kurose & Ross,
  Computer Networking:
  A Top-Down Approach, 5th ed.
  http://wps.aw.com/aw_kurose_network_5

- Stevens,
  UNIX Network Programming
  Prentice Hall, 1990
  http://www.kohala.com/start/unp.html

Socket Programming
Acknowledgements

- Donahoo & Calvert,
  TCP/IP Sockets in C: Practical Guide for Programmers, 2nd ed.
  http://cs.ecs.baylor.edu/~donahoo/practical/CSockets/

- Donahoo & Calvert,
  TCP/IP Sockets in Java: Practical Guide for Programmers, 2nd ed.
  Morgan Kaufmann, 2008
  http://cs.ecs.baylor.edu/~donahoo/practical/JavaSockets/

- Python.org http://wiki.python.org/moin/TcpCommunication
- Python.org http://wiki.python.org/moin/UdpCommunication

Socket Programming
Additional Reading

Some material in these foils comes from the textbook supplementary materials:

- http://www.cs.utsa.edu/~korkmaz/teaching/cn-resources/programs/capitalize-tcp/
- http://www.cs.utsa.edu/~korkmaz/teaching/cn-resources/programs/capitalize-udp/
- http://gaia.cs.umass.edu/ntu_socket/
- http://beej.us/guide/bgnet/
- http://java.sun.com/docs/books/tutorial/networking/sockets/
- http://www.faqs.org/faqs/unix-faq/socket/
- http://www.iana.org/assignments/port-numbers
- http://docs.freebsd.org/44doc/psd/20.ipctut/paper.pdf
- http://docs.freebsd.org/44doc/psd/21.ipc/paper.pdf