RW354

Principles of Computer Networking

A.E. Krzesinski and B.A. Bagula
Department of Computer Science
University of Stellenbosch

Last updated: 21 July 2003

The material presented in these slides is used with permission from

- *Larry L. Peterson and Bruce S. Davie. Computer Networks: A Systems Approach (Second Edition). Morgan Kaufmann Publishers. ISBN 1-55860-577-0.*
- *William Stallings. Data and Computer Communications (Sixth Edition). Prentice-Hall Inc. ISBN 0-13-571274-2.*
- *Andrew S. Tannenbaum. Computer Networks (Fourth Edition). Prentice Hall Inc. ISBN 0-13-349945-6.*

Project outline

These slides describe the RW354 practical which consists of five projects related to network applications, design and implementation

1. *Why use a client-server model?*
2. *Client-server programming.*
3. *RW354 practicals.*

Why use a client-server model?

- *Robustness: the server runs in its own address space, and so it can protect itself against poorly written clients.*
- *Communication: the client and the server do not have to run on the same machine, so a communication mechanism is needed.*
- *Load balancing: the client and the server may run on different machines, enabling load balancing.*
- *Interoperability: the client and the server may use different hardware, operating system, etc.*

Client-server examples

- *FTP/FTPD: client (FTP) and server (FTPD)*
- *TELNET/TELNETD: client (TELNET) and server (TELNETD)*
- *TALK/TALKD: client (TALK) and server (TALKD)*
- *IRC/IRCD: client (IRC) and server (IRCD)*
- *LYNX: is an information-retrieval client*
- *HTTPD: server for for hyper-text Clients such as Lynx, or graphical-based ones (Netscape, Internet Explorer, or Mosaic).*

Client-server programming

The objective is to divide the work of Internet programs into two parts

- *a server knows how to do a certain task, or how to give a certain service*
- *a client knows how to talk to a user, and connect that user to the server.*

There are two types of client server systems

- *a single-client server serves only one client*
- *a multi-client server serves many clients either simultaneously or one after the other.*

The role of clients

- *provide a convenient user interface*
- *establish a connection with the server*
- *after connection establishment*
  - *clients receive commands from the user, translate them to the server's language (protocol) and send them to the server*
  - *clients receive messages from the server, translate them into human-readable form and show them to the user.*

The role of servers

- *servers accept requests from clients*
- *servers handle requests from clients*
- *servers return results to clients.*

Inter Process Communication (IPC)

- *the Inter Process Communication (IPC) interface uses a layer above the transport layer*
- *IPC uses socket programming: message destinations are specified as socket addresses identified by a socket port number and an Internet address.*

Types of communication

IP supports three types of communication

- *datagram communication (UDP): connectionless*
- *stream communication (TCP): connection-oriented*
  - *UDP and TCP use the standard Application Program Interface (API) for sending and receiving data from the computer's hardware*
- *raw sockets bypass the standard API and give direct access to computer hardware (dangerous).*

The client algorithm

- *get the server's address (e.g. erwin.cs.sun.ac.za)*
- *form a working address (e.g. 127.96.53.21)*
- *connect to the server*
- *while (not finished) do :*
  - *wait for information from server or user*
  - *if (information from server) do*

    parse information, show to user, update local state information, etc.
  - *else if (a user command) do*

    parse command, send to server, or deal with locally.

Single Client-Server algorithm

- *bind a port on the computer*
- *forever do*
  - *listen to the port for connection requests*
  - *accept an incoming connection request*
  - *if (authorized client) do*
    while (connection still alive) do
    - *receive request from client*
    - *handle request*
    - *send results of request, or error messages*
  - *else abort connection*

Multi Client-Server algorithm

- *bind a port on the computer*
- *forever do*
  - *listen on the port for connection requests from new or existing clients*
  - *if (new request) accept connection*
  - *if (un-authorized client) close connection*
  - *else if (close request) close connection*
  - *else if (request from existing client) do*
    - *receive request from client*
    - *handle request*
    - *send results, or error message*

Elementary TCP Client-Server interaction

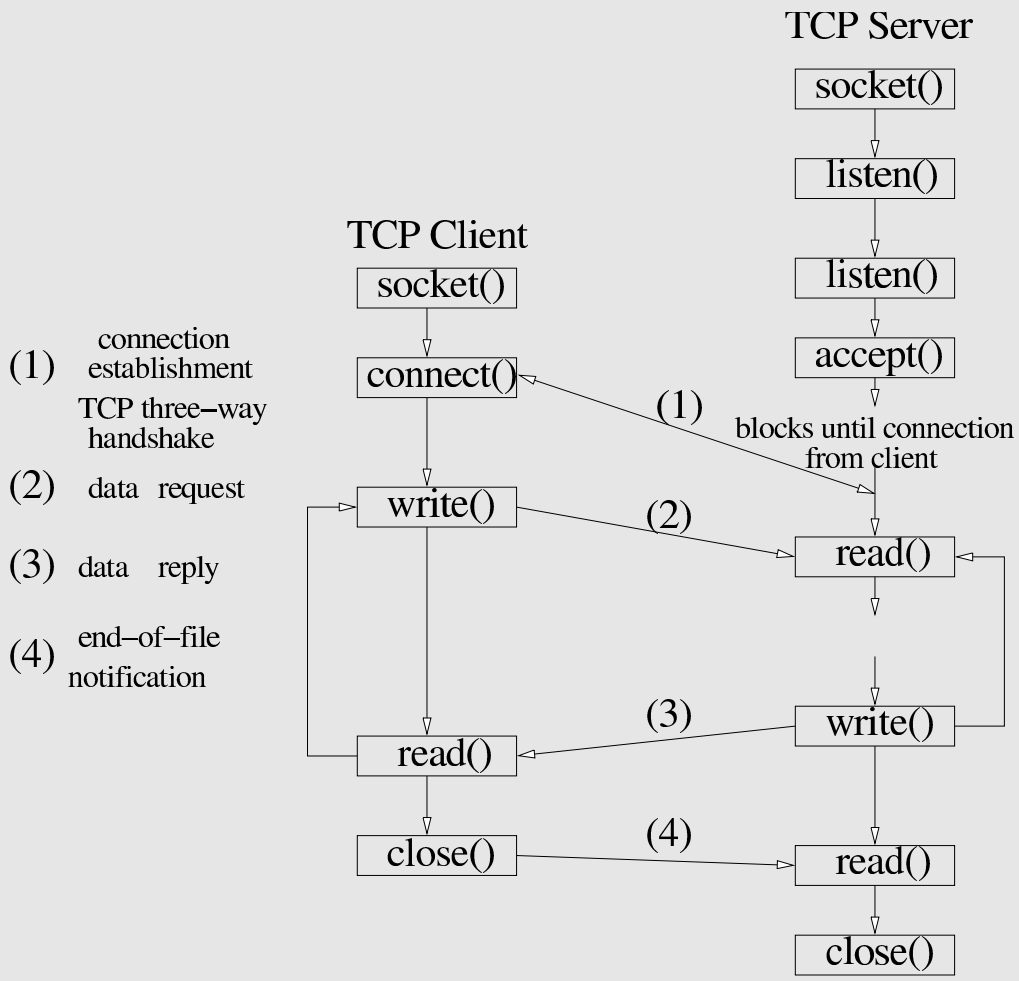


Figure 1: The TCP socket model

Elementary UDP Client-Server interaction

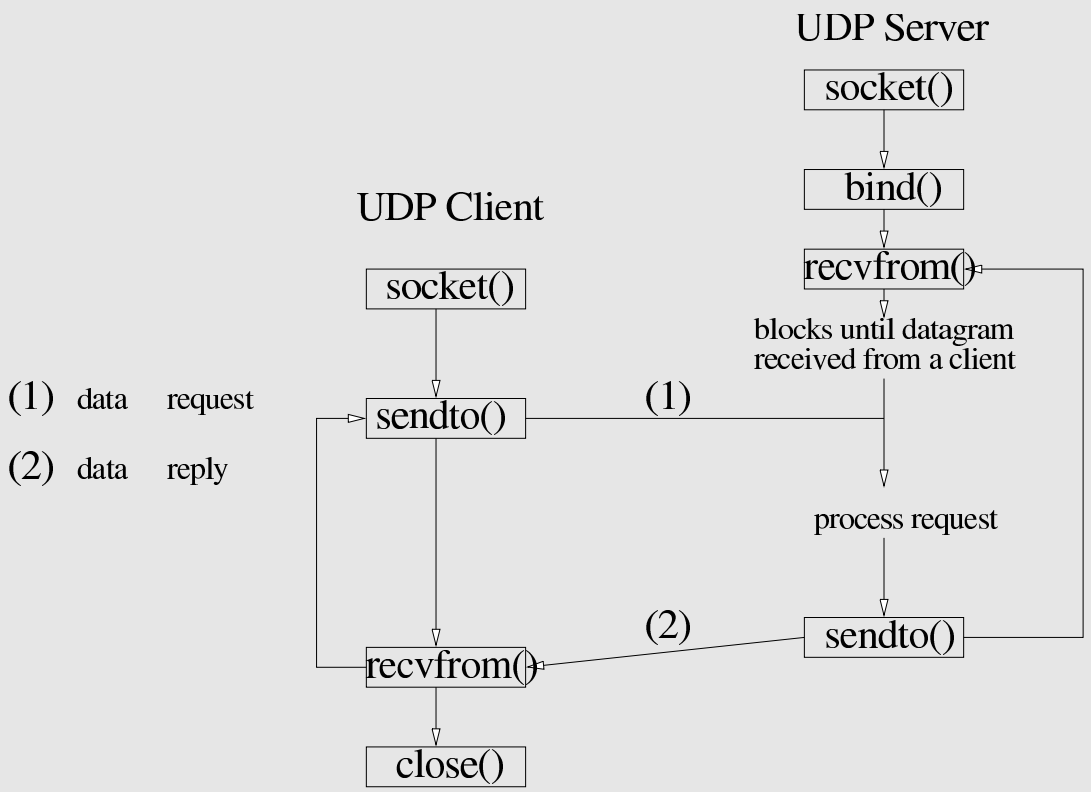


Figure 2: The UDP socket model

Elementary socket functions

- `socket` *creates an Internet socket*
- `bind` *binds the server's port to the socket*
- `listen` *converts a socket to listening socket*
- `accept` *accepts a client connection*
- `connect` *establishes a TCP connection with the server*
- `read`
- `write`
- `exit` *terminates the program*
- `close` *terminates the program.*

RW354 practical

The RW354 practical consists of five projects related to network application, design and implementation.

- *The educational objective: to implement concepts related to socket programming and multithread-based operations using UDP, TCP and raw sockets.*
- *The design and programming objectives are specific to the following network concepts: (1) IP addressing, (2) IP routing, (3) IP traffic Engineering, (4) IP security and (5) Voice over IP.*

Where applicable, IETF Draft/RFC compliant design and programming are encouraged.

Report

For each project, the following items must be submitted

1. *the project report*
2. *the executable file*
3. *all source codes.*

The report, executable code and source files must be submitted as a single zip file whose name has the following form: `GROUP.ProjectNUMBER.zip` *where* `GROUP` *is the group name and* `NUMBER` *is the project number.*

Assessment

The project assessment is based on the following factors for each project

- ***15 points*** *for completeness and quality of the project report*
- ***15 points*** *for completeness of the test procedures*
- ***50 points*** *for correct operation of the client and server processes*
- ***10 points*** *for implementation of optional features*
- ***10 points*** *for IETF Draft/RFC compliance.*

Project 0: elementary Client-Server interaction

- *The educational objective: to become familiar with client-server protocols including UDP sockets, TCP sockets.*
- *The programming aims: no design is required for this project.*

  Use the examples provided by the TCP and UDP `echo` *and* `daytime` *programs to learn more about how sockets work.*

Client-Server examples

TCP/UDP `echo`

- *the client reads a line of text from its standard input and writes the line to the server*
- *the server reads the line from its network input and echoes the line back to the client*
- *the client reads the echoed line and prints it on its standard output.*

TCP/UDP `daytime`

- *the client establishes a connection to the server*
- *the server sends back the current time and date in a human-readable format.*

Project 1: elementary socket programming

- *The educational objective: to become familiar with client-server protocols including UDP sockets, TCP sockets, raw sockets and multi-thread operation.*
- *The design and programming aims: to design, implement and test client and server programs that implement the Simple Registration Protocol (SRP) protocol and a SYN Denial of Service (DoS) attack as defined below.*

The Simple Registration Protocol

The SRP allows

- *client processes to register their activities to a server*
- *the server process to update a database located on the server side of the client-server application using messages received from clients*
- *the client messages are related to simple activities such as registration at a conference, booking a classroom for a lecture, finding information concerning a known activity, etc.*

A SYN Denial of Service Attack

A typical initiation of a communication between two computers using the TCP/IP protocol is as follows

- *the client sends a packet with a SYN (synchronous idle character) flag to the host*
- *the host replies with a SYN ACK (acknowledge) packet*
- *the client receives the SYN ACK*
- *the client sends a final ACK packet to the host.*

A DoS attack

A DoS attack uses raw sockets to create a fake return address for launching a DoS

- *the client (intruder) sends a SYN packet to the host with a fake return address*
- *the host receives the SYN packet from the intruder, reads the return address and sends a SYN-ACK packet to the fake return address*
- *the host waits for several minutes before realizing that no one was connecting to it*
- *the intruder quickly opens many ports on the host to consume all of its resources.*

Project 2: IP addressing

- *The educational objective: to reinforce concepts related to client-server protocols, socket calls, multi-threading and the operation of streams.*
- *The design and programming aims: to design, implement and test client and server programs that implement the message exchange used by the IP addressing protocols and concepts including*
  - *an IP data communication model*
  - *Network Address Translation (NAT)*
  - *Dynamic Host Configuration (DHCP).*

IP data communication

IP data communication is achieved by using the ARP and RARP protocols to send packets from an IP host to another host located either on the same network or on a remote network

- *RARP (Reverse Address Resolution Protocol) translates a physical address into a logical address*
- *ARP (Address Resolution Protocol) translates a logical address into a physical address.*

The algorithm

The algorithm implemented by a host H_1 to find another host H_2 is as follows

- *H_2 uses the netmask to check if H_1 is on the same local network*
- *if H_2 is on the same local network as H_1 then H_1 uses ARP to obtain the MAC (physical) address of H_2: H_1 sends the message to H_2 using the Ethernet protocol*
- *else H_1 uses ARP to obtain the MAC address of the gateway: H_1 sends the message to the gateway: the gateway sends the message to H_2.*

Network Address Translation (NAT)

Many hosts on a LAN may need to communicate with each other but not with hosts outside the LAN. The LAN hosts do not need globally unique IP addresses – they need private IP addresses that are unique within the LAN.

- *The objective: to solve the problem of address space consumption by combining the use of private addresses defined by RFC1918 on the LAN side and public Internet addresses on the Internet side.*
- *The requirement: the router connecting a LAN to the Internet has two addresses: a private address defined by RFC1918 on the LAN side and a public IP address assigned by an ISP at the Internet side.*

The algorithm

- *A host H_1 sending a packet to a host H_2 includes the source IP and the source port together with the destination IP address and port number.*
- *The router adds an entry into a table it keeps.*
- *The table maps each private IP address into a temporarily assigned globally unique IP address.*

  The globally unique IP address is written into the packet header and the packet is sent to H_2.

  When packets are returned from H_2, the NAT table is consulted: the locally unique IP address is written into the packet header and the packet is sent to H_1.

Dynamic Host Configuration (DHCP)

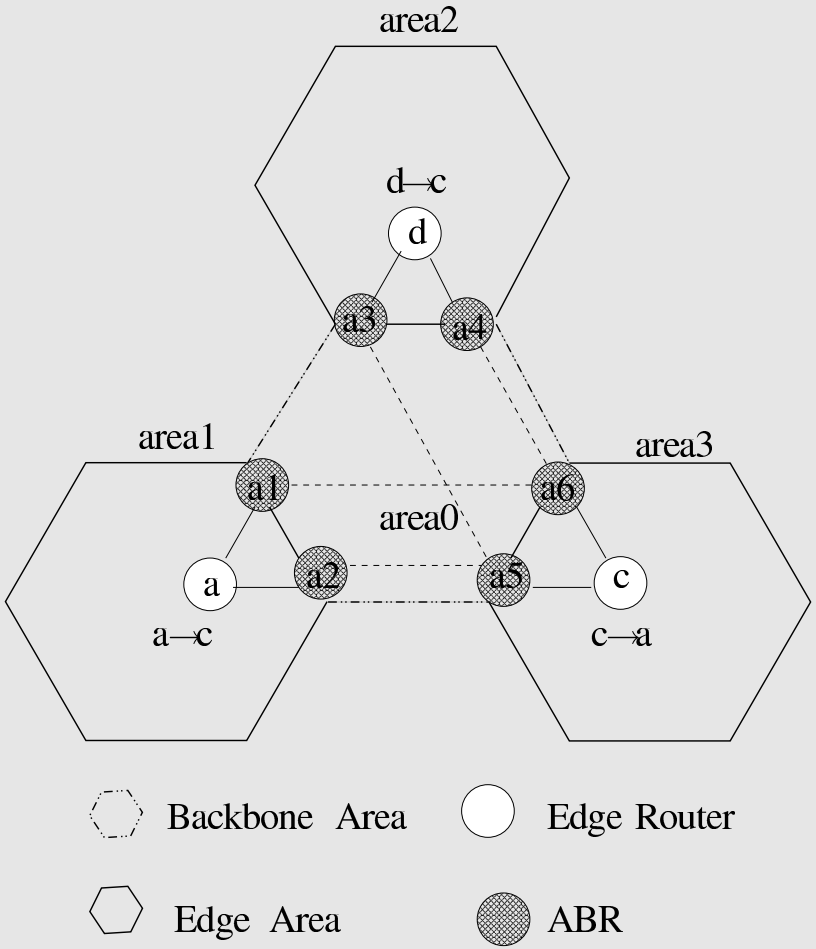
DHCP simplifies the task of defining an addressing system by

- *defining a DHCP server that allocates IP addresses and maintains a database and DHCP clients*
- *defining pools of TCP/ IP addresses which are then allocated to clients by the server*
- *setting configuration parameters like the subnet mask, default router, DNS server that are required to make TCP/ IP work correctly*
- *working across most TCP/ IP routers and allocating IPs according to the subnet the request came from*
- *leasing IP addresses for periods of time.*

Project 3: Virtual Link Advertisement Protocol (VLAP)

- *The educational objective: to reinforce client-server concepts, socket calls , multi-threading and the operation of streams.*
- *The design and programming aims: to design, implement and test client and server programs that implement the VLAP protocol defined below.*

VLAP illustration



The VLAP protocol is implemented between the Path Computer Servers (PCSs) and the Path Computation Clients (PCCs) of a multi-area routing domain.

VLAP Messages

Each routing area is provided a PCS that computes virtual links for that area. The virtual links are advertised to neighbor areas for inter-area path computation.

The VLAP protocol includes three virtual link computation messages types illustrated by figure **??** *for a path from source* s_1 *to the destination* d_3

- *virtual link request (VLQ)*
- *virtual link reply (VLP)*
- *virtual request forwarding (VRF)*

A path from source s_1 *to the destination* d_3 *is provided by virtual links* $v_1 : s_1 - r_1$, $v_2 : r_1 - r_2$, *and* $v_3 : r_3 - d_3$ *provided by the PCS of the OSPF areas* a_1, a_2 *and* a_3.

Project 4: The Secure Simple Registration Protocol

The SSRP extends the simple Registration protocol (SRP) to include security features

- *The educational objective: to reinforce security concepts by extending the client-server protocols designed for the simple registration protocol to include IP security features.*
- *The design and programming aims: to design, implement and test client and server programs that implement the SSRP protocol defined below.*

Project 5: Voice Over IP

- *The educational objective: to reinforce concepts related to Voice over IP (VoIP), the Session Inititiation Protocol (SIP), client-server protocols, socket calls, multi-threading and the operation of streams.*
- *the design and programming aims: to add services to the SIP protocol using NIST SIP and to test these services.*

The SIP protocol (RFC3261) describes how to set up Internet telephone calls, video conferences, and other multimedia connections.

SIP communication

SIP provides 6 communication methods

- `INVITE` requests initiation of a session
- `ACK` confirms that as session has been initiated
- `BYE` requests termination of a session
- `OPTIONS` queries a host about its capabilities
- `CANCEL` cancels a pending request
- `REGISTER` informs a redirection server about the user's current location

The NIST SIP kit

NIST SIP is

- *a kit for research and testing VoIP methodologies and tools*
- *enables the measurement of voice quality for existing VoIP end-systems and networks*
- *facilitates the analysis of new VoIP coding and representation technologies and network transport services*
- *expedites the research and development of protocols and platforms for programmable telephony services.*