Chapter 3: Protocols and client/Server Applications

Protocol

- Formal description of how entities communicate.
- A language system use to exchange information.

Attribute

- A variable used to support protocol operation.
- Sometimes called a field.

Value

- Assigned to an attribute at a particular point in time.
- Convey protocol state.

Simple Mail Transfer Protocol (SMTP)

- SMTP is a message transfer Agent (MTA).
- SMTP is a push protocol; it pushes the message from the client to the server.
- SMTP is a TCP/IP application layer protocol used in sending and receiving e-mail.
- It transmits simple text messages only.
  - 7-bit ASCII format
- Uses information written on envelope of mail.
  - Message header
  - Contains recipient address and other information.
- Does not look at contents
  - Message body
- SMTP usually is implemented to operate over Internet port 25.

Basic Operations

- Mail is created by user agent program (mail client).
- Messages queued and sent as input to SMTP sender program
  - Typically a server process
  - Daemon on Unix
    - sendmail or qmail

Mail message contents

Each queued message has:

- Message text
  - RFC 822 header with message envelop and list of recipients.
  - Message body, composed by user.
- A list of mail destinations
  - Derived by user agent/ SMTP server from the header.
  - May require expansion of mailing lists.

SMTP Sender

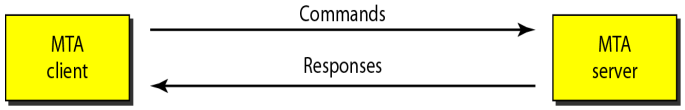
- Takes message from queue.
- Transmits to proper destination host.
  - Via SMTP transaction.
  - Over one or more TCP connections to port 25.
- When all destinations processes message is deleted.

SMTP Receiver

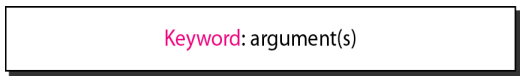
- Accepts arriving message.
- Places in user mailbox or copies to outgoing queue for forwarding.
- Receiver must:
  - Verify local mail destinations.
  - Deals with errors
    - Transmission
    - Lack of disk space

Commands and Responses

- SMTP uses commands and responses to transfer messages between an MTA client and an MTA server.
- Each command or reply is terminated by a two-character (carriage return and line feed) end-of-line token.



- Command format



- Commands

Keyword	*Argument(s)*
HELO	Sender's host name
MAIL FROM	Sender of the message
RCPT TO	Intended recipient of the message
DATA	Body of the mail
QUIT	
RSET	
VRFY	Name of recipient to be verified
NOOP	
TURN	
EXPN	Mailing list to be expanded
HELP	Command name
SEND FROM	Intended recipient of the message
SMOL FROM	Intended recipient of the message
SMAL FROM	Intended recipient of the message

- Responses

Code	*Description*
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input
Transient Negative Completion Reply	
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted: insufficient storage

Permanent Negative Completion Reply	
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

- **Example**

  Let us see how we can directly use SMTP to send an e-mail and simulate the commands and responses. We use TELNET to log into port 25 (the well-known port for SMTP). We then use the commands directly to send an e-mail. In this example, forouzanb@adelphia.net is sending an e-mail to himself. The first few lines show TELNET trying to connect to the Adelphia mail server. After connection, we can type the SMTP commands and then receive the responses.

==========

$ telnet mail.adelphia.net 25
Trying 68.168.78.100 . . .
Connected to mail.adelphia.net (68.168.78.100).

```
========== Connection Establishment ==========
  220 mta13.adelphia.net SMTP server ready Fri, 6 Aug 2004 . . .
HELO mail.adelphia.net
  250 mta13.adelphia.net
```

```
================== Mail Transfer ================
MAIL FROM: forouzanb@adelphia.net
  250 Sender <forouzanb@adelphia.net> Ok
RCPT TO: forouzanb@adelphia.net
  250 Recipient <forouzanb@adelphia.net> Ok
DATA
  354 Ok Send data ending with <CRLF>.<CRLF>
From: Forouzan
TO: Forouzan

This is a test message
to show SMTP in action.
.
```

```
================== Connection Termination ==============
   250 Message received: adelphia.net@mail.adelphia.net
QUIT
   221 mta13.adelphia.net SMTP server closing connection
Connection closed by foreign host.
```

E-mail Message (RFC22)

- Email is one of the most popular Internet services.
- Email is a method of exchanging digital messages from an author to one or more recipients.
- Email Environment
    - User Agent
        - Composing messages, reading messages, replying messages, forwarding messages, handling mailboxes.
    - Message Transfer Agent: SMTP
    - Message Access Agent: POP and IMAP
- Internet email messages consist of two major sections:

    - *Header* — Structured into fields such as From, To, CC, Subject, Date, and other information about the email.

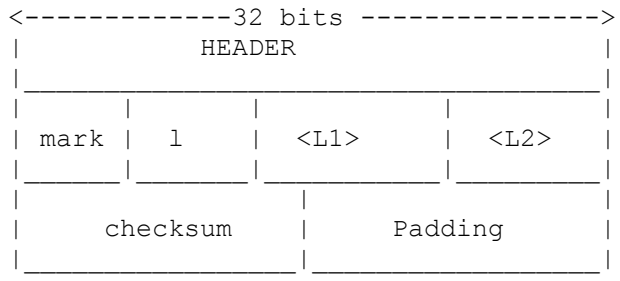
- *Body* — The basic content, as unstructured text; sometimes containing a signature block at the end. This is exactly the same as the body of a regular letter.

- RFC22 defines Host-Host Control Message Formats. It is written by Vint Cerf , October 17, 1969.
- All Host-Host control messages consist of sequences of 8-bit bytes of the form:

<control byte> <parameter byte l> ... <parameter byte n>

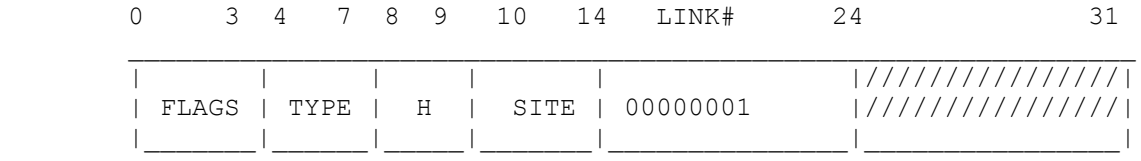
Control byte	Parameter	Interpretation
<0>	<L>	Please establish primary connection; our output link # is L
<1>	<L1> <L2>	Please establish auxiliary connection parallel to our primary output link L1. The auxiliary output link is L2.
<2>	<L1> <L2>	OK primary. Your primary output link to us was L1; our primary output link to you is L2.
<3>	<L1> <L2>	OK auxiliary. Your auxiliary output link is L1, our auxiliary output link is L2.
<4>	<L>	Not OK primary. We cannot establish a primary connection. Your primary output link number was L.
<5>	<Li> <L2>	Not OK auxiliary. We cannot establish an auxiliary connection. Your primary output link no was L2.
<6>	<L>	Please stop transmitting over link number L. This is called the CEASE directive.
<7>	<L>	We are CLOSING our output link number L. You may get this message before the last message arrives over this link since control messages are higher priority than regular data messages.
<8>	<L>	UNCEASE: that is, you may resume transmitting over output link number L.

Each control message is embedded in the appropriate message structure
e.g.:



typical control message (please
establish auxiliary link #L2
parallel to our primary link #l)

The header for all HOST-HOST control messages is given below:



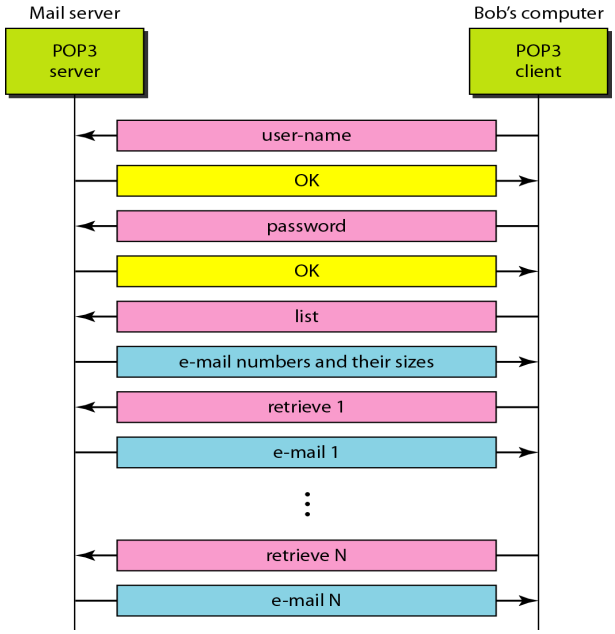
where FLAGS - 0000
TYPE - 0000 (regular message)
H - host #(0-3) at SITE (usually 0 for single HOST sites)
SITE - Site #
LINK# - 00000001 (HOST-HOST control link)

Pretty Good Privacy (PGP)

- PGP is a data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication.
- PGP is often used for signing, encrypting and decrypting texts, E-mails, files, directories and whole disk partitions to increase the security of e-mail communications.
- It was created by Phil Zimmermann in 1991.
- Provides Application layer security. It is primarily used for email security.
- **PGP's operation consists of five services:**
    - **Authentication Service**
      Sender authentication consists of the sender attaching his/her digital signature to the email and the receiver verifying the signature using public-key cryptography.
    - **Confidentiality Service**
      This service can also be used for encrypting disk files. PGP uses symmetric-key encryption for confidentiality. The user has the choice of three different block-cipher algorithms for this purpose: CAST-128, IDEA, or 3DES, with CAST-128 being the default choice.
    - **Compression Service**
      By Default PGP compresses the email message after appending the signature but before encryption. This makes long-term storage of messages and their signatures more efficient. This also decouples the encryption algorithm from the message verification procedures. Compression is carried out with the ZIP algorithm.
    - **E-Mail Compatibility Service**
      Since encryption, even when it is limited to the signature, results in arbitrary binary strings, and since network message transmission is character oriented, we must represent binary data with ASCII strings.
      PGP uses Base64 encoding for this purpose.
    - **Segmentation Service**
      For long email messages (these are generally messages with attachments), many email systems place restrictions on how much of the message will be transmitted as a unit. PGP has built-in facilities for such segmentation and re-assembly.

Post Office Protocol (POP)

- POP is a Message Access Agent.
- POP is a pull protocol; it is used to retrieve e-mail from a mail server.
- There are two versions of POP.
  - POP2- became a standard in the mid-80 and requires SMTP to send messages.
  - POP3- can be used with or without SMTP.
- A POP3 server listens on well-known port 110.
- **POP3 has two modes:**
  - **Delete mode-** mails deleted as they are read
  - **Keep mode-** mails remain in the mailbox
- Mail accessing using POP involves following steps.
  - Mail client initiates connection with client's mailbox on the mail server.
  - The client opens a connection to the server on TCP port 110.
  - It then sends its user name and password to access the mailbox.
  - The user can then list and retrieve the mail messages, one by one.
- An example of downloading Email using POP3 is shown below.



Internet Message Access Protocol (IMAP)

- IMAP is a Message Access Agent.
- IMAP4 is most widely used Massage Access Protocol.
- IMAP4 has the following additional properties than POP3.

- A user can check the email header prior to downloading.
- A user can search the contents of the email for a specific string of characters prior to downloading.
- A user can partially download email. This is especially useful if bandwidth is limited and the email contains multimedia with high bandwidth requirements.
- A user can create, delete, or rename mailboxes on the mail server.
- A user can create a hierarchy of mailboxes in a folder for e-mail storage.

What's the difference between POP and IMAP?

POP	IMAP
Post Office Protocol	Internet Messaging Access Protocol
Best if you use only one computer to check email	Best if you use many different computers to check your email
Downloads your email to the particular computer you are checking it on	Your mail is always on the server
Allows you to keep a large backlog of email messages only limited by the size of your computer.	You are limited by your mailbox size quota for how many messages you keep, although you can archive old messages and save them onto your computer manually.
Does not have a web interface (Some webmail companies, such as Yahoo, will let you check POP mail)	Has a web interface. If you are using NCF Webmail, you are using IMAP.
New messages are downloaded in their entirety, you have to wait for the message to download.	New message headers are downloaded so you see all your mail faster, the message you want to read is not downloaded to your computer until you click on it.

Hypertext Transfer Protocol (HTTP)

- HTTP is the most popular TCP/IP application layer protocol, is at the heart of the Web.
- HTTP is a protocol used mainly to access data on the World Wide Web.
- HTTP functions as a combination of FTP and SMTP.
- HTTP is implemented in two programs: a client program and server program.
- HTTP defines the structure and exchange of HTTP messages between client and server.
- HTTP uses the services of TCP on well-known port 80.

(see more about http on chapter4)

File Transfer Protocol (FTP)

- FTP used for transferring files from one computer to another in an internetworking environment.
- FTP uses the services of TCP. It needs two TCP connections.
  - The well-known port 21 is used for the control connection.
  - The well-known port 20 for the data connection.
- The FTP client has three components:
  - user interface
  - the client control process
  - the client data transfer process
- The FTP server has two components:
  - the server control process
  - the server data transfer process
- The control connection is made between the control processes.
- The data connection is made between the data transfer processes.

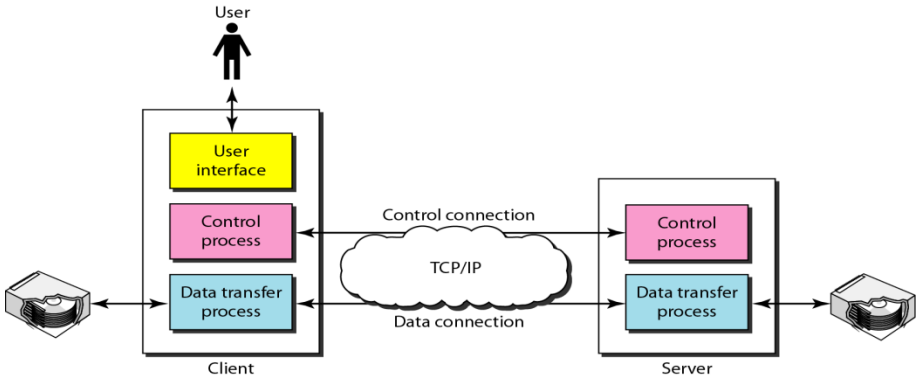


Fig. Model of FTP

- The control connection remains connected during the entire interactive FTP session.
- The data connection is opened and then closed for each file transferred.
- **Communication over Control Connection**
  - FTP uses the same approaches as SMTP to communicate across the control connection.
  - It uses the 7-bit ASCII character set.
  - Communication is achieved through commands and responses

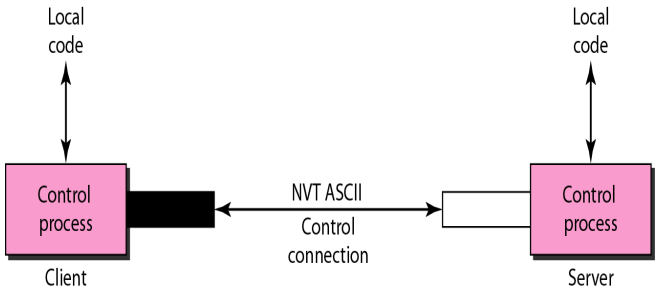


Fig. using the control connection

- **Communication over Data Connection**
  - File transfer occurs over the data connection under the control of the commands sent over the control connection.
  - File transfer in FTP means one of three things:
    - A file is to be copied from the server to the client. This is called retrieving a file. It is done under the supervision of the RETR command.
    - A file is to be copied from the client to the server. This is called storing a file. It is done under the supervision of the STOR command.
    - A list of directory or file names is to be sent from the server to the client. This is done under the supervision of the LIST command. Note that FTP treats a list of directory or file names as a file. It is sent over the data connection.

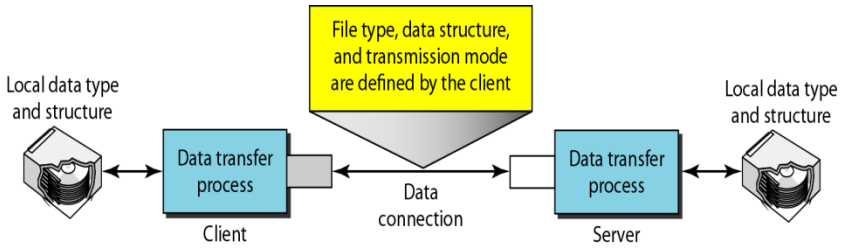
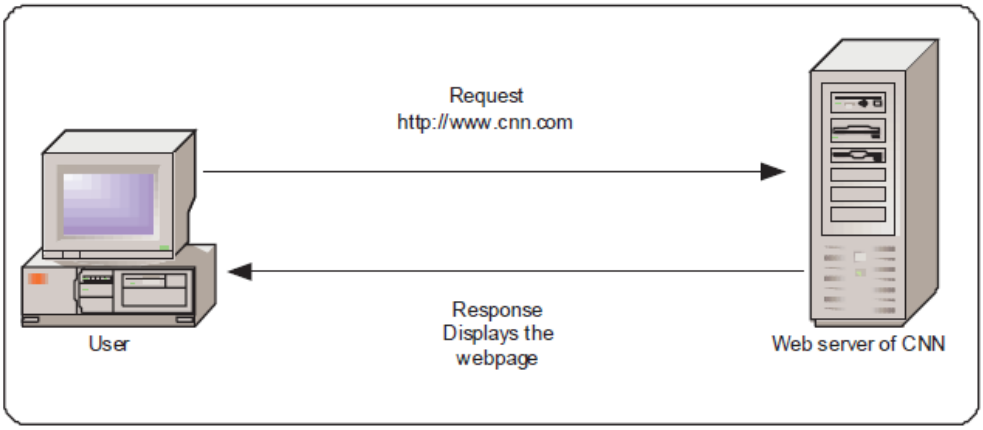


Fig. using the data connection

Client/Server Architecture

- A network architecture in which each computer or process on the network is either a client or a server.
- Servers are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network servers).
- Clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power.
- Client/server describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfills the request.
- Although the client/server idea can be used by programs within a single computer, it is a more important idea in a network.
- In a network, the client/server model provides a convenient way to interconnect programs that are distributed efficiently across different locations.

Example: to check your bank account from your computer, a client program in your computer forwards your request to a server program at the bank. That program may in turn forward the request to its own client program that sends a request to a database server at another bank computer to retrieve your account balance. The balance is returned back to the bank data client, which in turn serves it back to the client in your personal computer, which displays the information for you.



Client – The host that requests a service is called a client.
Server – The host that services the request.

Client Application (One Tier)

- The program runs on a machine.
- In most cases there are no separate application layers.
- Database programs are located on the same machine.
- At any moment only one user has the advantage of the program.

2-Tier Architecture

- 2-tier architecture is used to describe client/server systems where the client requests resources and the server responds directly to the request, using its own resources.
- This means that the server does not call on another application in order to provide part of the service.
- A two-tier (C/S) model first began to emerge with the applications developed for local area networks in the late eighties & early nineties, and was primarily based upon simple file sharing techniques implemented by X-base style products (dBase, FoxPro, Clipper, Paradox, etc.).
- Two-tier client/server architectures have 2 essential components
  - A Client PC and
  - A Database Server
-

2-Tier Considerations:

- Client program accesses database directly
  - Requires a code change to port to a different database
  - Potential bottleneck for data requests
  - High volume of traffic due to data shipping
- Client program executes application logic
  - Limited by processing capability of client workstation (memory, CPU)
  - Requires application code to be distributed to each client workstation

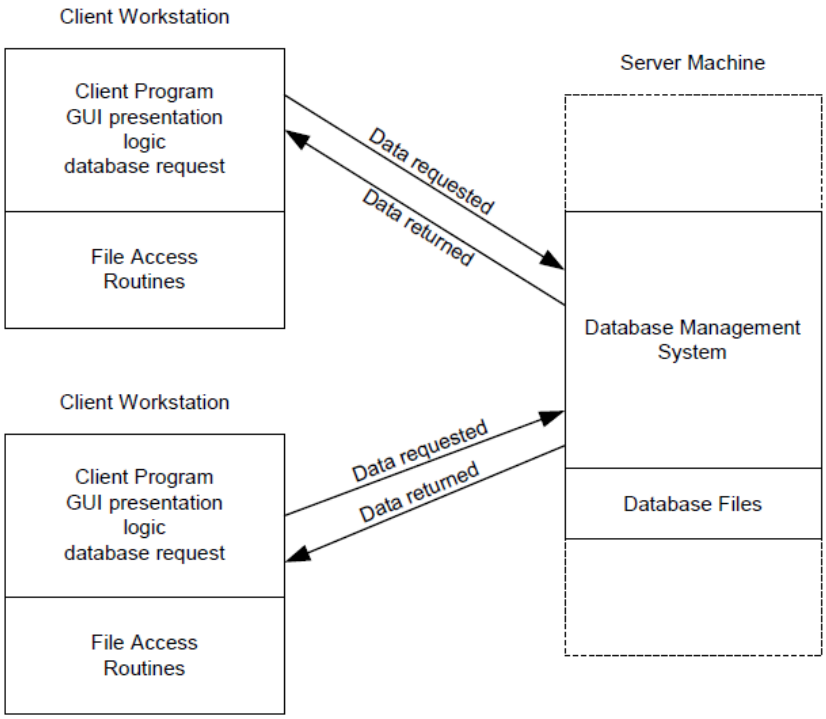


Figure 7.1 Client/Server 2-Tier Architecture

Two – Tier Pros and Cons

Advantages	Disadvantages
Development Issues: • Simple structure • Easy to setup and maintain	*Development Issues:* • Complex application rules difficult to implement in database server – requires more code for the client • Complex application rules difficult to implement in client and have poor performance • Changes to business logic not automatically enforced by a server – changes require new client side software to be distributed and installed • Not portable to other database server platforms
Performance: • Adequate performance for low to medium volume environments • Business logic and database are physically close, which provides higher performance.	*Performance:* • Inadequate performance for medium to high volume environments, since database server is required to perform business logic. This slows down database operations on database server.

3-Tier Architecture

In 3-tier architecture, there is an intermediary level, meaning the architecture is generally split up between:

- A client, i.e. the computer, which requests the resources, equipped with a user interface (usually a web browser) for presentation purposes
- The application server (also called middleware), whose task it is to provide the requested resources, but by calling on another server
- The data server, which provides the application server with the data it require

3-Tier Architecture Considerations:

- Client program contains presentation logic only
  - Less resources needed for client workstation
  - No client modification if database location changes
  - Less code to distribute to client workstations
- One server handles many client requests
  - More resources available for server program
  - Reduces data traffic on the network

Advantage of 3-tier Architecture

- Clear separation of user-interface-control and data presentation
- Re-definition of the storage strategy won't influence the client
- Trusted system
- Dynamic load balancing
- Change Management

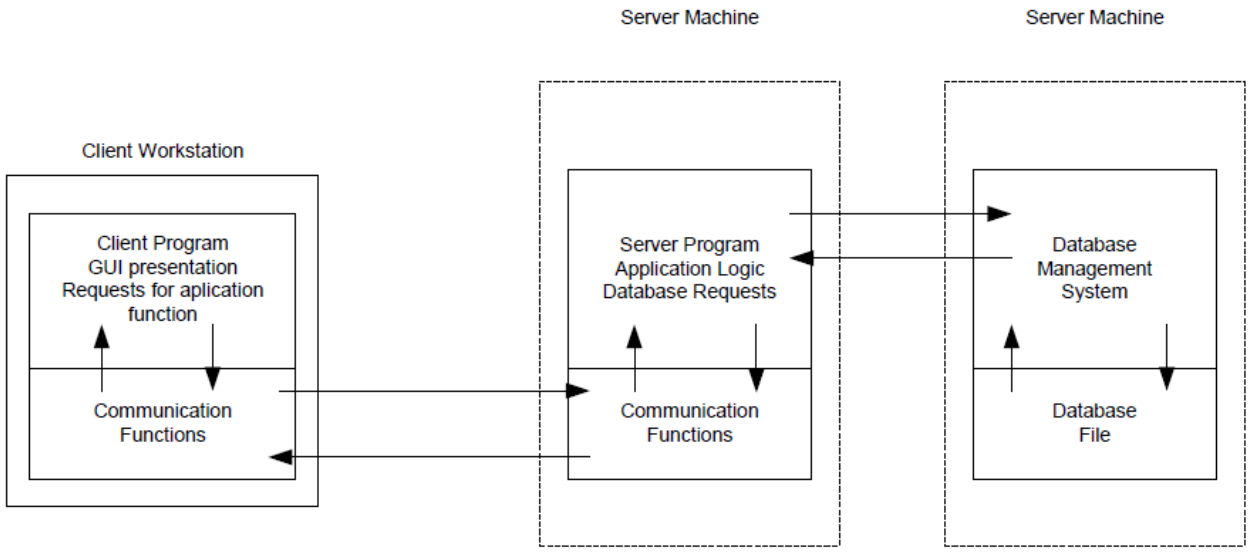


Figure 1.8. Typical 3 – Tier Architecture

3 – Tier Pros and Cons

Advantages	Disadvantages
Development Issues: • Complex application rules easy to implement in application server • Business logic off-loaded from database server and client, which improves performance • Changes to business logic automatically enforced by server – changes require only new application server software to be installed • Application server logic is portable to other database server platforms by virtue of the application software	*Development Issues:* • More complex structure • More difficult to setup and maintain.
Performance: • Superior performance for medium to high volume environments	*Performance:* • The physical separation of application servers containing business logic functions and database servers containing databases may moderately affect performance.

N-Tier Architecture

- N - Tier Client Server Network Architecture is an integrated Server System that helps to Distribute Data and Application workloads between different Web Server and Application Server.
- N-tier application architecture provides a model for developers to create a flexible and reusable application.
- Middle tier is divided into 2 or more unites
- N-Tier: An unlimited number of tiers.
- Each tier may have multiple computers.
- Advantages:
  - More powerful applications
  - Many services to many clients
  - Enhanced security, scalability and availability
- Disadvantages:
  - Much more complicated to design and model
  - Performance risks
  - Reliability is more difficult to achieve
  - More difficult to maintain software

Major Quality Attributes Effects of N-Tier architectures

- Performance
- Reliability
- Usability
- Security
- Availability
- Scalability
- Maintainability

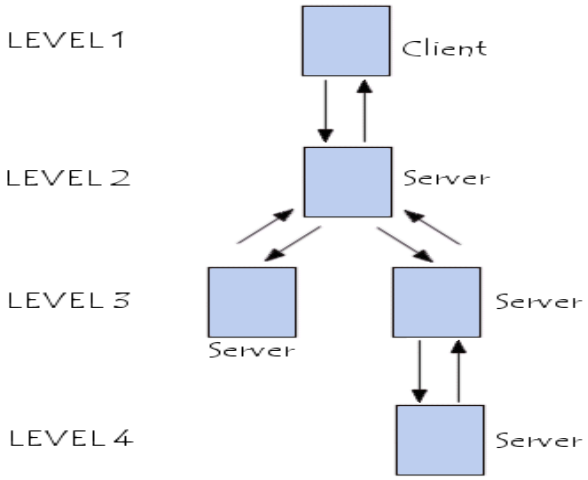


Fig. N-Tier Architecture

Benefits of the Client/Server Model

- **Divides Application Processing** across multiple machines:
  - Non-critical data and functions are processed on the client
  - Critical functions are processed on the server
- **Optimizes Client Workstations** for data input and presentation (e.g., graphics and mouse support)
- **Optimizes the Server** for data processing and storage (e.g., large amount of memory and disk space)
- **Scales Horizontally** – Multiple servers, each server having capabilities and processing power, can be added to distribute processing load.
- **Scales Vertically** - Can be moved to more powerful machines, such as minicomputer or a mainframe to take advantage of the larger system's performance
- **Reduces Data Replication** - Data stored on the servers instead of each client, reducing the amount of data replication for the application.

Comparison of Architectures

Architecture	Advantage	Disadvantage
One tier	Simple Very high performance Self-contained	No networking – can't access remote services Potential for spaghetti code
Two tiers	Clean, modular design Less network traffic Secure algorithms Can separate UI form business logic	Must design/implement protocol Must design/implement reliable data storage
Three tiers	Can separate UI, logic, and storage Reliable, replicable data Concurrent data access via transactions Efficient data access	Need to buy DB product Need to hire DBA Need to learn SQL Object-relational mapping is difficult
N tiers	Support multiple applications more easily Common protocol/API	Less efficient Must learn API (CORBA, RMI, etc.) Expensive products More complex, more faults Load balancing is hard

---

Universal Internet Browsing

…left for students

Multiprotocol Support

Complex data communication systems no longer utilize a single protocol to handle all transmission tasks. Instead, they require a set of protocol, called a protocol suite. The reason for using multiple protocols is to make them less completed; this simplifies dealing with problems that arise when machines communicate over a network. Such problems include the following:

- **Hardware failure**

- When a router or a host hardware fails, the protocol software needs to detect the failure and recover from it.

- **Network congestion**
  - The protocol software needs to detect when the network capacity has been exceeded and arrange a way to handle the congestion.
- **Packet Delay or loss**
  - The protocol software needs to adapt to long delays in order not to lose packets that were significantly delayed.
- **Data corruption**
  - The protocol software needs to detect and recover from transmission errors and corruption due to transmission impairments or hardware failures.
- **Data duplication or sequence error**
  - Networks that offer multiple routes may deliver data out of sequence or deliver duplicates of packets. The protocol software needs reorder packets and remove duplicates.

It is difficult or undesirable to write a single protocol that will handle everything in the network, particularly since many networks are heterogeneous. So the decision has been made to partition the communication problem into sub problems and organize the software into modules that handle one sub problem. The partitioning was based on the fact that the design of the network (or internet} and the organizations of the protocol software are interrelated: one cannot be designed without the other.

Two main models of protocol layering are available today. The first is based on the International Organization for Standardization (ISO). The second is based on research that led be the TCP/IP protocol suite. Layered protocols are designed so that layer n at the destination received exactly the same object sent by layer n at the source. In other words, they operate in a peer-to-peer mode.