Online ISSN: 0975-4172
Print ISSN: 0975-4350

Global Journal of Computer Science and Technology

DISCOVERING THOUGHTS AND INVENTING FUTURE

Technology Reforming Ideas

April 2011

Pinnacles

Analysis of Routing Protocols

Knowledge Penetration Process

Three Dimensional Clustering

Fingerprint Identification

The Volume 11
Issue 6
Version 1.0

ENG

Global Journal of Computer Science and Technology

Global Journal of Computer Science and Technology

Volume 11 Issue 6 (Ver. 1.0)

Global Association of Research

Incorporation No.: 0423089
License No.: 42125/022010/1186
Registration No.: 430374
Import-Export Code: 1109007027
Employer Identification Number (EIN):
USA Tax ID: 98-0673427

Global Journals Inc.

*(A Delaware USA Incorporation with "Good Standing"; **Reg. Number: 0423089**)*

Sponsors: Global Association of Research

Open Scientific Standards

Publisher's Headquarters office

Global Journals Inc., Headquarters Corporate Office, Cambridge Office Center, II Canal Park, Floor No. 5th, ***Cambridge (Massachusetts)***, Pin: MA 02141
United States
USA Toll Free: +001-888-839-7392
USA Toll Free Fax: +001-888-839-7392

Offset Typesetting

Global Association of Research, Marsh Road, Rainham, Essex, London RM13 8EU
United Kingdom.

Packaging & Continental Dispatching

Global Journals, India

Find a correspondence nodal officer near you

To find nodal officer of your country, please email us at *local@globaljournals.org*

eContacts

Press Inquiries: *press@globaljournals.org*
Investor Inquiries: *investers@globaljournals.org*
Technical Support: *technology@globaljournals.org*
Media & Releases: *media@globaljournals.org*

Pricing (Including by Air Parcel Charges):

For Authors:

22 USD (B/W) & 50 USD (Color)

Yearly Subscription (Personal & Institutional):

200 USD (B/W) & 500 USD (Color)

Editorial Board Members (HON.)

Contents of the Volume

Global Journal of Computer Science & Technology
Volume 11 Issue 6 Version 1.0 April 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

A Quick Review of On-Disk Layout of Some Popular Disk File Systems

By Wasim Ahmad Bhat , S. M. K. Quadri
Kashmir University

Abstract- : Disk file systems are being researched since the inception of first magnetic disk in 1956 by IBM. As such, many good disk file system designs have been drafted and implemented. Every file system design addressed a problem at the time of its development and efficiently mitigated it. The augmented or new designs rectified the flaws in previous designs or provided a new concept in file system design. As such, there are many file systems that have been successfully d in operating systems. Among these designs, some file systems have made an influential impact on the file system design because of their capability to cope up with change in hardware technology and/or user requirements or because of their innovation in file system ign or because time favored them which allowed them to find space in popular operating systems. In this paper, we vide a quick review of on-disk layout of some popular disk file systems across many popular platforms like Windows, Linux & Macintosh. The goal of this paper is to explore the on-disk layout of these file systems to identify the various layout policies and data structures they exploit which made them to be adapted by their native and other operating systems.

Keywords: *File System, On-Disk, Design, Popular, Review.*

Classification: *GJCST Classification: FOR Code: 100699,100604*

A QUICK REVIEW OF ON-DISK LAYOUT OF SOME POPULAR DISK FILE SYSTEMS

Strictly as per the compliance and regulations of:

A Quick Review of On-Disk Layout of Some Popular Disk File Systems

Wasim Ahmad Bhat[α], S. M. K. Quadri[Ω]

Abstract- Disk file systems are being researched since the inception of first magnetic disk in 1956 by IBM. As such, many good disk file system designs have been drafted and implemented. Every file system design addressed a problem at the time of its development and efficiently mitigated it. The augmented or new designs rectified the flaws in previous designs or provided a new concept in file system design. As such, there are many file systems that have been successfully implemented and incorporated in operating systems. Among these designs, some file systems have made an influential impact on the file system design because of their capability to cope up with change in hardware technology and/or user requirements or because of their innovation in file system design or because time favored them which allowed them to find space in popular operating systems. In this paper, we provide a quick review of on-disk layout of some popular disk file systems across many popular platforms like Windows, Linux & Macintosh. The goal of this paper is to explore the on-disk layout of these file systems to identify the various layout policies and data structures they exploit which made them to be adapted by their native and other operating systems.

Keywords- File System, On-Disk, Design, Popular, Review.

I. Introduction

Since the advent of computers a mechanism for persistent storage of data and/or programs was needed. On the time line, magnetic disks are the primitive [1] (introduced in 1956 as data storage for an IBM accounting computer) and still widely used secondary storage device. Magnetic disk drive is the most primitive and cost effective storage device. There has been continuous improvement in its hardware technology to increase its performance and capacity [2]. Although performance has seen less improvement with respect to capacity, but the tremendous drop in cost per unit byte, reliability over solid state storage and increase in capacity have made disk drives every body's choice [3]. And hence, disk file systems have attracted researchers over the globe to exploit its pros and minimize its cons.

A File System is a way to organize, store, retrieve, and manage information on a permanent storage medium such as a disk [4]. File system is an important part of an operating system as it provides a way by which data can be stored, organized, navigated, accessed and retrieved in form of files and directories from storage sub system. It is generally a kernel module which consists of algorithms to maintain the logical data structures residing on the storage subsystems. The basic key functions that every file system incorporates are basic file operations like copy, move, create, delete and rename, efficient organization of data for quick storage and retrieval and efficient use of disk space. Apart from these basic functions some file systems also provide additional functions such as compression, encryption, file streams and others. Keeping all the hardware parameters and workload constant, the performance of a hard disk will all depend upon the type of file system used. In general file systems were developed in an incremental fashion by individual efforts of researchers and software industry with high cohesion with the hardware limitation and requirements at that time. Later, refinement of existing file systems [5] and new file systems were developed to keep pace with hardware enhancement and off course need.

To understand the file system design in general and on-disk layout specifically, we need to review the history of its invention a bit so that we can get some overview of the environment and situations in which the first file systems were drafted and implemented. Further, this history will give us some idea about the incremental file system design that has been followed since the inception of first file system. In the early days of computers, file systems were simply considered part of the operating system that ran the computer, and in those days operating systems themselves were rather new and fancy. One of the first file systems to have a name was DEC Tape [6], named after the company that made it (Digital Equipment Corporation) and the physical system the files were stored on (reel-to-reel tape recorders). The tapes acted like very slow disk drives. DEC Tape stored an astoundingly small 184 kilobytes of data per tape on the PDP-8 [7], DEC's popular early minicomputer. It was called a minicomputer only because, while the size of a refrigerator, it was still smaller than IBM's mainframes that took up entire rooms. Of course, the invention of the transistor and integrated circuit allowed another

About[α] - Research scholar in P. G. Department of Computer Sciences, Kashmir University, India. He did his Bachelor's degree in Computer Applications from Islamia College of Science & Commerce, India and Master's degree in Computer Applications from Kashmir University, India.
E-mail- wasim.ahmed.bhat@gmail.com
About[Ω] - Head, P. G. Department of Computer Sciences, Kashmir University, India. He did his M. Tech. in Computer Applications from Indian School of Mines, India and Ph. D. in Computer Sciences from Kashmir University.
E-mail-quadrismk@hotmail.com

whole round of miniaturization. DEC slowly became extinct while the rest of the world moved to microcomputers.

In 1972, Gary Kildall [8] got interested in working with Microprocessors and got involved with Intel. His research was related to compilers and code optimization. While working as a consultant in Intel, Kildall developed the Programming Language/Microprocessor (PL/M) [9] and the Control Language/Microprocessor (CP/M) [10]. He wrote CP/M to test out PL/M compiler. CP/M allowed him to store files and retrieve them from 8-inch floppy. He was able to run and test programs from it, modify them and check their portability by putting floppy in other machine's drive. CP/M got very popular because it used small amount of memory required to run it, approximately 3 ½ K and had a file system, but it does not have a name. It was very simple, as it stored files in a completely flat hierarchy with no directories. File names were limited to eight characters plus a three-character "extension" that determined the file's type. This was perfectly sensible because it was exactly the same limitation as the computer Kildall was working with. Gary Kildall and the company he founded to sell CP/M, Digital Research, soon became very wealthy and the usage of CP/M was tripling every year. It turned out that a lot of microcomputer companies needed an operating system, and Gary had designed it in a way that separated all the BIOS from the rest of the OS. Unfortunately for Kildall, other people soon got the same idea he had.

A programmer named Tim Patterson [11] wrote his own OS called "QDOS" (Quick and Dirty Operating System) [12] that was a quick and dirty clone of everything CP/M did, because he needed to run an OS on a new 16-bit computer, and Gary hadn't bothered to write a 16-bit version of CP/M yet. QDOS had a slightly different file system than CP/M, although it did basically the same thing and didn't have directories either. Patterson's file system was based on a 1977 Microsoft program called Microsoft Disk Basic, which was basically a version of Basic that could write its files to floppy disks. It used an organization method called the File Allocation Table.

Bill Gates bought Tim Patterson's QDOS for $50,000, and renamed it MS-DOS [13]. He now was able to sell it to IBM and every company making an IBM clone, and Gary found himself quickly escorted from the personal computing stage. As it was originally a quick and dirty clone of a file system designed for 8-bit microcomputers in the 1970s that was itself a quick-and-dirty hack that mimicked the minicomputers of a decade earlier, FAT was not really up for very much. It retained CP/M's "8 and 3" file name limit, and the way it stored files was designed around the physical structure of the floppy disk drive, the primary storage device of the day. The introduction of hard disks soon made FAT-12 obsolete but file systems got attention and every individual researcher and software industry professional recognized its importance and started either enhancing and augmenting the older designs or re-designing some new file systems from scratch.

In this paper, we will look at some most popular file systems' on-disk layout. The popularity of the file systems selected is solely based on the popularity of the operating systems that support them natively. The goal of this paper is to look at the layout policies they exploit and data structures they use to mitigate the challenges for which they were designed. In this paper, we will review the native file systems of Windows, Linux and Macintosh operating systems.

II. FAT File Systems

The design of FAT [14] file system is very simple as it uses simple data structures. This simplicity in design has made FAT file system popular and supported by almost every operating system. In today's world, several digital devices, such as mini mp3 players, smart phones, digital cameras, etc. are becoming part of our life. These devices exchange data frequently with desktop computers. The PC discovers these devices as standard USB mass storage devices and automatically mounts the file system inside them. This is possible only if the file system used in device is supported by the PC's operating system. That is why; conventional FAT file system is a useful format for solid state memory cards as it provides a convenient way to share data by being supported by almost all operating systems [15]. As mentioned before, FAT12 was the first FAT file system but was able to address only limited number of sectors as it was developed for floppy disks. Later, with the introduction of hard disk drive, FAT16 was introduced and with higher capacity drives, FAT32 and now exFAT [16] (unofficially called FAT64). Almost all the flavors of FAT file system follow same design with the exception of pointer width in bits that is used to access the sectors (or Clusters) and which gives the FAT suffix 12, 16, 32 and 64. FAT12 and FAT16 are obsolete now whereas exFAT is not widely used yet, in contrast to FAT32 which is supported by almost every operating system.

The FAT32 file system consists of 4 different data structures to allow semantics of hierarchical file systems to be implemented on volume.

a) *BOOT Sector*

Boot Sector is located at the beginning of the volume. It includes an area called BPB (Bios Parameter Block) at offset 11 of length 49 bytes and contains some basic file system information. The rest of the sector usually contains boot code with boot signature word (0x55AA) at offset 509.

BPB is a one dimensional table that contains variable length entries. Each entry in BPB stores file system layout information except one (BPB_Reserved) which is kept reserved for future extension. Different versions of FAT file systems have size difference in BPB and contain different entries. Table 1 shows the BPB for FAT32 file system. Each entry has been given a name to identify its role along with entry offset and size.

Name	Offset (byte)	Size (bytes)
BS_jmpBoot	0	3
BS_OEMName	3	8
BPB_BytsPerSec	11	2
BPB_SecPerClus	13	1
BPB_RsvdSecCnt	14	2
BPB_NumFATs	16	1
BPB_RootEntCnt	17	2
BPB_TotSec16	19	2
BPB_Media	21	1
BPB_FATSz16	22	2
BPB_SecPerTrk	24	2
BPB_NumHeads	26	2
BPB_HiddSec	28	4
BPB_TotSec32	32	4
BPB_FATSz32	36	4
BPB_ExtFlags	40	2
BPB_FSVer	42	2
BPB_RootClus	44	4
BPB_FSInfo	48	2
BPB_BkBootSec	50	2
BPB_Reserved	52	12
BS_DrvNum	64	1
BS_Reserved1	65	1
BS_BootSig	66	1
BS_VolID	67	4
BS_VolLab	71	11
BS_FilSysType	82	8

Table 1. Description of FAT32 BPB

Reserved Sectors immediately follow Boot Sector. The number of reserved for volume includes Boot Sector and is indicated by BPB at offset 14 of Boot Sector. Typically, reserved sectors include FSInfo sector at sector 1 and BkBoot sector at sector 6 of the volume. FSinfo sector further qualifies the FAT32 volume, while BkBoot is replica of boot sector and is used for recovery purposes.

b) *File Allocation Table (FAT)*

The File Allocation Table (FAT) is an array of n-bit wide entries and spans over a number of sectors indicated by BPB at offset 36 of Boot Sector. FAT32 volume has generally 2 consecutive copies of FAT data structure and is called FAT Mirroring. Mirroring is done for recovering from FAT corruption in case one copy of FAT gets corrupt. In case of solid state storage devices, FAT is not mirrored to prolong the life of solid state device by reducing the write cycles. Bit 7 of BPB offset 40 of boot sector indicates whether FAT is mirrored or not. This data structure of FAT file system gives it the name and is the heart of the file system. The suffixes used by various FAT file systems indicate the bit width of entries in FAT data structure. Thus, in FAT32, the FAT entries are 32-bit wide.

FAT data structure is a table that stores the information about which clusters are free, used or possibly unusable. A cluster is a fixed length group of consecutive data sectors which are located immediately after FAT data structure and occupy rest of the volume. The number of sectors per cluster is indicated by BPB at offset 13 of boot sector. FAT file system always allocates space on storage device in terms of clusters. This is done to increase the performance of the file system by avoiding individual multiple accesses to disk. Thus, the file system may suffer from high internal fragmentation if cluster is too large and there are many small sized files; and may degrade the performance if it is small and the volume has large sized files. Depending upon the type of file system and size of the volume, the cluster size varies but the number of sectors per cluster is restricted to a value that is power of 2 i.e. 1,2,4,8,16,32,64, etc. In addition to keep track of used and unused clusters, FAT data structure also keeps track of chain of clusters allocated to a file. The technique used by FAT32 file system is simple. Every file and directory except the root directory of volume has an entry in its parent directory that contains its name, attributes & 32 bit wide entry that indicates the first cluster number allocated to it. The FAT data structure entries are 32 bit wide and each entry uniquely corresponds to the cluster on the volume sequentially i.e. the first entry corresponds to cluster 0, second entry corresponds to the cluster 1, etc. The formula used to locate the cluster entry in FAT data structure for any valid cluster number N is

$$FATOffset = N * 4$$

$$ThisFATSecNum = BPB\_ResvdSecCnt + \frac{FATOffset}{BPB\_BytsPerSec}$$

$$ThisFATEntOffset = FATOffset \% BPB\_BytsPerSec$$

where ThisFATSecNum is the logical sector number of the volume and ThisFATEntOffset is the offset in the sector where 32-bit FAT entry corresponding to cluster number N exists. The contents of any valid cluster entry in FAT can have values as shown in Table 2.

FAT32 Cluster Entry Values	Description
0x00000000	Is Free Cluster
0x00000001	Reserved value
0x00000002 – 0x0FFFFFEF	Is Used Cluster and value points to next cluster in the chain allocated to file/directory
0x0FFFFFF0 – 0x0FFFFFF6	Reserved values
0x0FFFFFF7	Some Bad sector in Cluster,

Global Journal of Computer Science and Technology Volume XI Issue VI Version I April 2011

	Unusable
0x0FFFFFF8 – 0x0FFFFFFF	Is Last Cluster in file/directory or EOC (End Of Cluster chain) marker

Table 2. Description of Valid FAT Entries

Let's suppose two files, say MYFILE1.TXT and MYFILE2.TXT are currently residing on a FAT32 volume such that the former is fragmented and is 3 clusters long while the latter is not fragmented and is 2 clusters long as shown in figure 1. MYFILE1.TXT has first cluster allocated 0x00000029, FAT contents against that cluster shows another cluster 0x0000002A, then 0x0000002D whose FAT contents show that this cluster is the last cluster in chain. Similarly, for MYFILE2.TXT the first cluster allocated is 0x0000002B whose FAT contents point to next cluster in chain, 0x0000002C, which is the last cluster in chain as pointed by its FAT content.

Each file/directory may occupy one or more clusters depending upon its size. Thus, a file/directory is represented by a chain of these clusters. However, these clusters are not necessarily to be stored adjacent to one another on the disk's surface but are often fragmented throughout the volume as shown in figure 1 where MYFILE1.TXT is fragmented while MYFILE2.TXT is not.

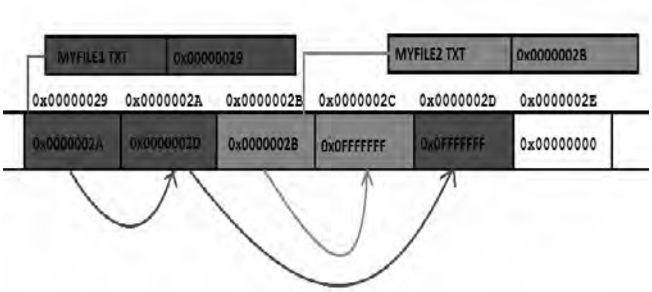


Figure 1. A Snapshot of FAT Data Structure.

As memory cost per unit capacity is dramatically decreasing every year and storage size is increasing, the maximum number of clusters have increased dramatically and also the cluster size. In FAT32, the FAT entry is 32 bit wide which points to next cluster in chain but it only uses lower 28 bits to address clusters. Thus, FAT entry values say 0xA0000000 and 0xB0000000 point to same cluster on volume. As such, 2^{28} clusters can exist on FAT32 volume. As mentioned before, the successive major versions of FAT file systems are named after the number of table entry bits; FAT12, FAT16, FAT32 & FAT64, the goal of every new version is to address large volume and large file size. Although, KFAT [17], TFAT [18] and FATTY [19] versions of FAT file system have also been designed but the goal was reliability. Because the number of bytes per sector as indicated by BPB at offset 11 of Boot sector is always divisible by 4, a FAT32 FAT entry never spans over a sector boundary.

The first two entries in FAT store special values:

- The first entry contains a copy of BPB at offset 21 of Boot Sector which is 8 bit long which indicates the type of storage media. The remaining 20 bits between high 4 and low 8 of this entry are set to 1.
- The second entry stores the EOC marker. The high order two bits of this entry are sometimes, used for dirty volume management: high order bit if set to 1 indicates that last shutdown was clean otherwise abnormal. The next highest bit, if set to 1 indicates that during the previous mount no disk I/O errors were detected else there were.

Because the first two FAT entries store special values, there is no cluster 0 or 1. The first addressable cluster in FAT32 FAT data structure is cluster 2, which is the reason why BPB value at offset 44 of Boot Sector which indicates the Root Directory cluster number cannot be less than 2 and is usually 2, i.e., the Root Directory is at the start of file/directory region.

c) *Directory Structure*

The semantics of hierarchical file system lives on the notion of files and directories. The hierarchical file system is like a tree where every non-leaf node is a subdirectory containing any number of non-leaf nodes (sub-directories) or leaf nodes (files) or both. The tree begins at a root node called root directory. In FAT32, the root directory is of variable size and is assigned the first cluster, whose address is indicated in BPB at offset 44. Among all the files and directories that may reside on FAT32 volume, root directory is the only directory that does not have filename and attributes; more precisely does not have any entry like other files and directories have. In case of FAT12 and FAT16, root directory is located at fixed location after FAT copy and is of fixed size indicated in BPB.

A directory is an array of 32 byte wide structures where each structure represents a file or directory either existing or deleted and in case of long name support, the remaining the parts of long name. The structure of 32 byte wide entry of directory is shown in Table 3.

Name	Offset (byte)	Size (bytes)
DIR_Name	0	11
DIR_Attr	11	1
DIR_NTRes	12	1
DIR_CrtTimeTenth	13	1
DIR_CrtTime	14	2
DIR_CrtDate	16	2
DIR_LstAccDate	18	2
DIR_FstClusHI	20	2
DIR_WrtTime	22	2
DIR_WrtDate	24	2
DIR_FstClusLO	26	2
DIR_FileSize	28	4

Table 3. Description of FAT32 Directory Entry Structure

The name and other metadata about a file are all stored in the 32-byte directory entry for file. The list of characters that cannot be used in a file name, “. “ / \ [ ] ; : | = or 0x20 is really an operating system issue, not a file system issue. Linux, via its FAT support, can create files with some of these characters in their names. This may cause problems with portability if that disk is later read in a Windows environment. Dating back to the creation of the first FAT12 volumes in the 70’s, all files were given a name in the 8.3 naming convention. That is, eight characters for the name and three characters for an extension that identified the type of file; ‘dot’ is never saved. Long file name support was later introduced but not in any semblance of an elegant way.

Usually, FAT32 places the root directory in the first available cluster, which places it right behind the FAT area. All other directories in all the FAT file systems will be allocated clusters as they need them and can reside anywhere on the disk.

III. NT File System

NTFS was designed to quickly perform standard file operations such as read, write & search. It was developed from scratch although some concepts were borrowed from OS/2’s HPFS [20]. The design of NTFS file system is bit complex but very nicely drafted and crafted. It includes many new features of modern file system like transparent compression and encryption, sparse files, multiple data streams, reliability, fast recovery, security features, privileges and permissions, and representation of everything as file and everything belonging to a file as collection of attribute/value pairs from filename attribute to data attribute [21]. The design of NTFS file system is such that every sector of volume belongs to some file unlike FAT. Even the file system metadata that describes the file system is part of some file.

When a volume is formatted with NTFS file system, it leads to the creation of several system files used by file system to store volume metadata and implement the file system. These files are not accessible to user directly. These system files have entry just like other regular volume files and directories have, and have been given some reserved names prefixed by $ sign. The standard configuration of NTFS file system has 16 system files out of which last 4 entries are reserved [22]. Table 4 lists these system files along with their $MFT name, $MFT entry offset (explained later) and purpose of the file.

System File	File Name	MFT Record	Purpose of the File
Master file table	$Mft	0	Contains one base file record for each file and folder on an NTFS volume.
Master file table 2	$MftMirr	1	A duplicate image of the first four records of the $MFT.
Log file	$LogFile	2	Contains a list of transaction steps used by NTFS for recoverability.
Volume	$Volume	3	Contains information about the volume.
Attribute definitions	$AttrDef	4	A table of attribute names, numbers, and descriptions.
Root file name index	$	5	The root folder.
Cluster bitmap	$Bitmap	6	A representation of the volume showing which clusters are in use.
Boot sector	$Boot	7	Includes the BPB used to mount the volume and additional bootstrap loader code used if the volume is bootable.
Bad cluster file	$BadClus	8	Contains bad clusters for the volume.
Security file	$Secure	9	Contains unique security descriptors for all files within a volume.
Upcase table	$Upcase	10	Converts lowercase characters to matching Unicode uppercase characters.
NTFS extension file	$Extend	11	Used for various optional extensions such as quotas, reparse point data, and object identifiers.
		12-15	Reserved for future use.

Table 4. $MFT Entry name, & Offset & Purpose of NTFS System Files

a) *$BOOT*

The location of $BOOT file is fixed and resides on first 16 sectors of NTFS volume. The first sector is called Boot Sector as it contains the boot strap code and following 15 sectors are boot sector’s IPL (Initial

Program Loader). The boot sector is duplicated at last sector of the volume. The boot sector of $BOOT file contains two data structures; BPB followed by Extended BPB. Table 5 describes the BPB and Extended BPB of NTFS boot sector (Offset, Length & Field Name).

Byte Offset	Field Length	Field Name
0x0B	WORD	Bytes Per Sector
0x0D	BYTE	Sectors Per Cluster
0x0E	WORD	Reserved Sectors
0x10	3 BYTES	*always 0*
0x13	WORD	*not used by NTFS*
0x15	BYTE	Media Descriptor
0x16	WORD	*always 0*
0x18	WORD	Sectors Per Track
0x1A	WORD	Number Of Heads
0x1C	DWORD	Hidden Sectors
0x20	DWORD	*not used by NTFS*
0x24	DWORD	*not used by NTFS*
0x28	LONGLONG	Total Sectors
0x30	LONGLONG	Logical Cluster Number for the file $MFT
0x38	LONGLONG	Logical Cluster Number for the file $MFTMirr
0x40	DWORD	Clusters Per File Record Segment
0x44	DWORD	Clusters Per Index Block
0x48	LONGLONG	Volume Serial Number
0x50	DWORD	Checksum

Table 5. BPB & Extended BPB of NTFS file system

Among other things, the two data structures contain sectors per cluster, bytes per sector, total sectors, logical cluster number of $MFT file, logical cluster number of $MFTMirr file, clusters per file record segment and clusters per index block.

b) *$MFT*

$MFT file or Master File Table file is an array of fixed records where each record represents uniquely every file or directory of the volume even the system files including the $MFT file. The first 16 records are reserved for system files. Table 4 shows the list of first 16 records ordered as per their position and corresponding system files they represent along with short description. The first entry represents the $MFT file itself while second entry represents the mirrored copy of $MFT file named $MFTMirr whose first record is identical to first record of $MFT. Actually, $MFTMirr duplicates first 4 records of $MFT for recovery purpose. In case the first record of $MFT that defines $MFT, is corrupted the file system code should read the second record of $MFT to locate $MFTMirr and read its first record to build $MFT or should directly read the $MFTMirr file's first record by locating its position from logical cluster number in BPB to build $MFT. As $MFT actually defines the NTFS layout, logical cluster number of $MFT is kept in BPB so that file system driver can locate $MFT at boot time. $MFT is not fixed like FAT and hence can be relocated in case it is damaged; same is true for other system files.

A record in $MFT is a 1 KB structure that stores attributes of file/directory to which it corresponds. NTFS stores everything belonging to file or directory as a collection of attribute/value pairs including filename, security information, time stamps, data, etc [23]. Each $MFT record corresponds to a unique file. If a file has large number of attributes, more than one record is allocated to a file. In this case, the first record that stores the location of others in Attribute List attribute is called Base File Record. Whether a file consumes one or more $MFT records, if the value for any particular attribute is completely stored in record, such an attribute is called Resident Attribute. Several attributes are defined as always being resident so that NTFS can locate non-resident attributes for e.g. $STANDARD_INFORMATION, $INDEX_ROOT, $ATTRIBUTE_LIST, etc. A non-resident attribute is one whose value cannot be completely stored in an $MFT record. In such case, NTFS allocates clusters for the attribute's data separate from $MFT. This area is called a *run* or technically an *extent*. If resident attribute's value grows, it is converted to non-resident attribute and allocated a run. $DATA attribute for files greater than 1 KB, $BOOT, $MFTMirr and $LogFile is always non-resident. Table 6 shows the standard attribute names and their description [24]. Actually attributes correspond to numeric codes which NTFS uses to order (in ascending order) the attributes within an $MFT record with same attribute types appearing more than once in case a file has multiple values for that attribute. Most attributes never have names, though Index related attributes and $DATA attribute often does. Names distinguish among multiple attributes of same type that a file can include. The value of an attribute is a byte stream and is stored as a separate stream in a file. NTFS does not read and write files instead attribute streams. The read and write APIs exported by file system driver normally operate on file's unnamed $DATA attribute.

Attribute Type	Description
Standard Information	Includes information such as timestamp and link count.
Attribute List	Lists the location of all attribute records that do not fit in the base MFT record.
File Name	A repeatable attribute for both long and short file names. The long name of the file can be up to 255 Unicode characters. The short name is the 8.3, case-insensitive name for the file. Additional names, or hard links, required by POSIX can be included as additional file name attributes.
Security Descriptor	Describes who owns the file and who can access it.

Data	Contains file data. NTFS allows multiple data attributes per file. Each file typically has one unnamed data attribute. A file can also have one or more named data attributes, each using a particular syntax.
Object ID	A volume-unique file identifier. Used by the distributed link tracking service. Not all files have object identifiers.
Logged Utility Stream	Similar to a data stream, but operations are logged to the NTFS log file just like NTFS metadata changes. This is used by EFS.
Reparse Point	Used for volume mount points. They are also used by Installable File System (IFS) filter drivers to mark certain files as special to that driver.
Index Root	Used to implement folders and other indexes.
Index Allocation	Used to implement folders and other indexes.
Bitmap	Used to implement folders and other indexes.
Volume Information	Used only in the $Volume system file. Contains the volume version.
Volume Name	Used only in the $Volume system file. Contains the volume label.

Table 6. Standard Attribute Types & their Description

Each $MFT record begins with an entry header which is 42 bytes long. This standard header contains a magic number "FILE", number of entries in fix up array, $Log File sequence number, Sequence number, Hard Link count, offset to first attribute, flags that indicate whether record is in use or not, used and allocated size of MFT entry, file reference to base file record in case it is not base record, attributes and fix up values. Each attribute begins with a standard header containing information about the attribute like type and length of attribute, length of name and offset to name, non-resident flag, etc. The header of every attribute is always resident and records whether the value is resident or non-resident.

For resident attributes, the header also contains the offset from the header to attribute's value and length of attribute's value. Figure 2 shows the typical structure of a $MFT entry record [25].

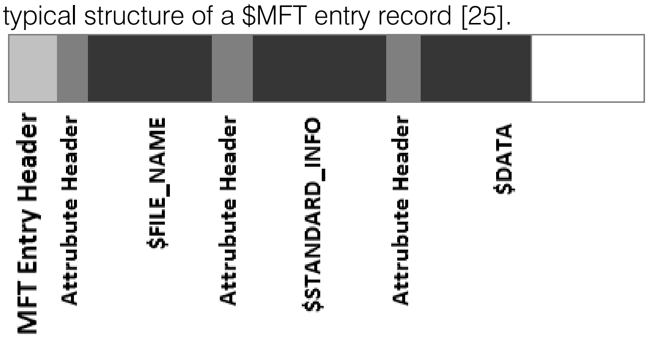


Figure 2. Typical MFT Entry Record

NTFS refers to physical locations on a disk by means of logical cluster numbers (LCNs). LCNs are simply the numbering of all clusters from the beginning of the volume to the end. To convert an LCN to a physical disk address, NTFS multiplies the LCN by the cluster factor (i.e. number of sectors per cluster) to get the physical byte offset on the volume. NTFS refers to the data within a file by means of virtual cluster numbers (VCNs). VCNs number the clusters belonging to a particular file from 0 through m. VCNs aren't necessarily physically contiguous, however; they can be mapped to any number of LCNs on the volume. When an attribute is nonresident, as the data attribute for a large file might be, its header contains the information NTFS needs to locate the attribute's value on the disk. This information is typically the VCN-to-LCN mapping pairs. Figure 3 shows the data attribute header containing VCN-to-LCN mappings for the two runs, which allows NTFS to easily find the allocations on the disk. Other attributes can be stored in runs if there isn't enough room in the $MFT file record to contain them.

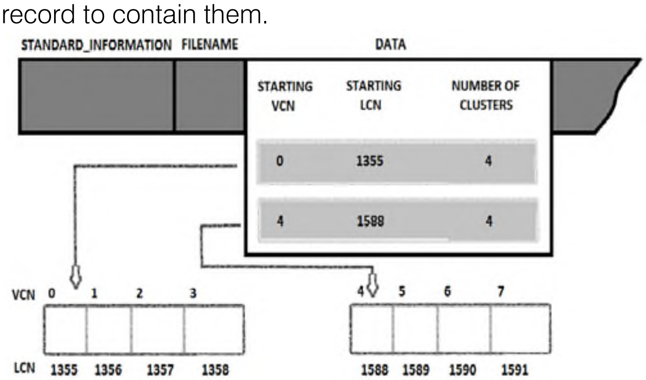


Figure 3. Non-Resident $DATA attribute of File.

A file on an NTFS volume is identified by a 64-bit value called a *File Reference*. The file reference consists of a file number and a sequence number. The file number corresponds to the file's $MFT record entry offset (or to that of base file record if the file has more than one file record entries). The file reference sequence number, which is incremented each time an $MFT file record position is reused, enables NTFS to perform internal consistency checks. If a particular file has too many attributes to fit in the $MFT record, a second $MFT record is used to contain the additional attributes (or attribute headers for nonresident attributes). In this case, an attribute called the *Attribute List* is added to file in base record. The attribute list attribute contains the name and type code of each of the file's attributes and the file reference of the $MFT record where the attribute is located. The attribute list attribute is also provided for those cases in which a file grows so large or so fragmented that a single $MFT record can't contain the multitude of VCN-to-LCN

Global Journal of Computer Science and Technology Volume XI Issue VI Version I April 2011

mappings needed to find all its runs. Files with more than 200 runs typically require an attribute list.

In NTFS, a file directory is simply an index of filenames, i.e., a collection of filenames along with their file references organized in a particular way (B-tree) for quick access [26]. To create a directory, NTFS indexes the filename attributes of the files in the directory. Conceptually, an $MFT entry for a directory contains in its Index Root attribute a sorted list of the files and/or directories in the directory. It also contains the file reference in the MFT where the file/directory is described and time stamp and size information for the file/directory. A large directory can also have nonresident attributes (or parts of attributes), as Figure 4 shows.

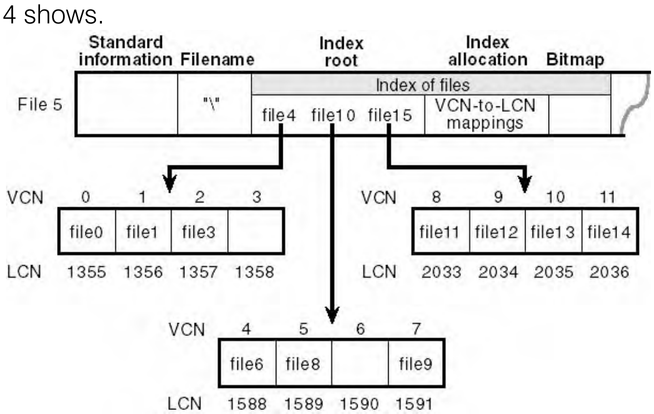


Figure 4. Root Directory [21]

In this example, the $MFT file record doesn't have enough room to store the index of files that make up this large directory. A part of the index is stored in the Index Root attribute, and the rest of the index is stored in non-resident runs called Index Buffers. For large directories, however, the filenames are actually stored in 4-KB fixed-size index buffers that contain and organize the filenames. Index Buffers implement a B+ tree data structure, which minimizes the number of disk accesses needed to find a particular file, especially for large directories. The index root attribute contains the first level of the B+ tree (root subdirectories) and points to index buffers containing the next level (more subdirectories, perhaps, or files). Figure 4 shows only filenames in the index root attribute and the index buffers (file6, for example), but each entry in an index also contains the file reference in the $MFT where the file is described and time stamp and file size information for the file. NTFS duplicates the time stamp and file size information from the file's $MFT record. This technique, which is used by FAT and NTFS, requires updated information to be written in two places. Even so, it's a significant speed optimization for directory browsing because it enables the file system to display each file's time stamps and size without opening every file in the directory.

The index allocation attribute maps the VCNs of the index buffer runs to the LCNs that indicate where the index buffers reside on the disk, and the bitmap attribute keeps track of which VCNs in the index buffers are in use and which are free. Figure 4 shows one file entry per VCN (that is, per cluster), but filename entries are actually packed into each cluster. Each 4-KB index buffer can contain about 20 to 30 filename entries. The B+ tree data structure is a type of balanced tree that is ideal for organizing sorted data stored on a disk because it minimizes the number of disk accesses needed to find an entry. In the $MFT, a directory's index root attribute contains several filenames that act as indexes into the second level of the B+ tree. Each filename in the index root attribute has an optional pointer associated with it that points to an index buffer. The index buffer contains filenames with lexicographic values less than its own. In Figure 4, for example, file4 is a first-level entry in the B+ tree. It points to an index buffer containing filenames that are (lexicographically) less than itself—the filenames file0, file1, and file3. Note that the names file1, file2, and so on that are used in this example are not literal filenames but names intended to show the relative placement of files that are lexicographically ordered according to the displayed sequence.

c) *$LogFile*

The internal structure of the $LogFile is not well understood. Once the log is full, the first entry is overwritten with the next new entry. What get logged are the individual transactions that make up each file access or file write or whatever. For instance, when modifying a file the following steps might occur:

- read $MFT entry for directory entry file is in
- read directory entry file is in
- read $MFT record for file
- write file
- update Atime in file's MFT record
- update Mtime in file's MFT record
- update Atime in directory entry for that file
- update Mtime in directory entry for that file

This list gets considerably longer if the file is encrypted or compressed. If the command fails before the entire string of transactions are completed, due to system crash or whatever other reason, the file system has to have a way to change each of the transactions involved back to their previous values in order to maintain consistency of the file system. The file system provides a reliable, crash-resilient environment.

d) *$Volume*

The file $Volume contains the name of the volume. That is its most important function. There is also volume information data in this file that contains a version number and a set of flags. The version number will be broken into two pieces, a major and a minor version number.

e) *$AttrDef*

This file contains the list of attributes available to the file system in this version of NTFS. It is because of this file that we know the catchy names for the attributes that we are using. The entry for the attribute also contains some information about the allowable sizes and location (resident or not) of the attribute can be.

f) *$Bitmap*

The $BitMap is a special file within the NTFS file system. This file keeps track of all of the used and unused clusters on an NTFS volume. When a file takes up space on the NTFS volume the location is uses is marked out in the $BitMap. The method of keeping track of cluster allocation is relatively simple. Each bit in the Bitmap represents 1 cluster; if that bit is "1" then the cluster is in use.

g) *$BadClus*

This file is the size of the NTFS volume, but is a sparse file of all zeros. Since zeros in sparse files are counted instead of saved, this file takes up no space on the disk. If a cluster is ever deemed 'bad', data will be written to this file at the same offset into this file as the offset the bad cluster is into the volume. This will causes this file to allocate clusters in the $bitmap file, which in turn prevents other files from trying to use the bad cluster in the future.

h) *$Secure*

In Windows NT, every file had a $Security_Descriptor attribute that did this job. Since many files had the same values in that attribute it was moved to this file so that data wasn't repeated.

i) *$UpCase*

Case in the file name is preserved, but is converted to all uppercase for sorting as the directory entry is created. This file contains the uppercase characters of 'every' UNICODE alphabet so that NTFS knows the proper alphabetical order of each code page of UNICODE without having to inherently knows every code page of UNICODE.

j) *$Extend*

$Extend is a directory that contains other system files. This allows for more system files to be added but without pushing the limit of the 16 I-nodes reserved for system files.

IV. Extended File Systems

In response to these problems, two new file systems were developed "Xia" and "Second Extended File System" [31]. Xia file system was based on Minix file system and provided long filenames, support for large volume size and 3 timestamps; while Ext2 file system was based on Ext file system with many reorganizations and improvements. It was designed with evolution in mind and contained space for future extension. Due to minimal design, Xia was more stable than Ext2 file system. Later, bugs were fixed in Ext2 file system and lots of improvements and new features were integrated. Ext2 file system became stable and de facto standard Linux file system. Ext2 uses VFS to extend the maximum volume from 2 GB to 4 TB. It allows root user to recover from incidents where other users overfill the file system. It uses variable length directory entries while filename length could be extended to 1012. Ext2 file system may use synchronous updates like BSD FFS [32]. This is the maximum reliability support provided by Ext2 file system. In synchronous updates, any modification to file system metadata like I-node, bitmap blocks, indirect blocks and directory blocks are synchronously written to the disk. Although this mechanism provides bit reliability, it leads to poor performance. Ext2 file system allows administrator to choose logical block size when creating file system. Block sizes can typically be 1024, 2048 and 4096 bytes. Ext2 implements fast symbolic links which does not use any data block on file system by not storing the target name in a data block but in I-node itself.

Andrew S. Tanenbaum wrote the Minix operating system in 1987 [27]. Tanenbaum created it for teaching purpose. Later, he published a textbook that included source code of Minix. This code was taken and published on Usenet where thousands of readers were able to examine and further develop Minix. As Minix was simple and bug free, Torvalds decided to incorporate its architecture into the operating system he was developing. Torvalds named his operating system Linux. One shortcoming of Torvalds first Linux kernel was that it only supported Minix file system. Minix file system was an efficient and relatively bug free piece of software. However, the restrictions in design of Minix file system were too limiting, so people started thinking and working on the implementation of new file system in Linux [28]. In order to add more file systems to Linux operating system, Torvalds modified a VFS written by Chris Provenzano and integrated it into the kernel [29]. After integration, a new file system called "Extended File System" was implemented which removed two big Minix limitations; maximum volume size and maximum filename length, but still there were some problems; no support for separate access, I-node and data modification timestamps. This file system used linked lists to keep track of free blocks and I-nodes and thus

Global Journal of Computer Science and Technology Volume XI Issue VI Version I April 2011

resulted in bad performance with aging [30].

Ext3 file system was designed to eliminate enormously long file system recovery times after the crash. Ext3 is a journaling file system [33]. A journaling file system differs from a traditional file system in that it keeps transient data in new location, independent of the permanent data and metadata on disk. Because of this, such a file system does not dictate that the permanent data has to be stored in any particular way. As such, it is quite possible for Ext2 file system on disk structure influenced by the layout of the BSD file system to be used in this file system. The layout of journaled Ext2 (or Ext3) file system on disk is entirely compatible with existing Ext2 file system. Ext2 file system design already includes a number of reserved I-node numbers; one among them is used for the file system journal. The features that separate Ext3 from being a valid ext2 system are journaling, h-tree indexing, and file system growth while the system is online. Ext4 [34] is the most recent version of the extended file system. This latest release hosts many new features such as a maximum volume size of one Exabyte, backwards compatibility with ext2 and ext3, online defragmentation, and nanosecond timestamps. The nanosecond timestamp is unique to Ext4 and allows applications that utilize file creation and modification times to track their timing in nanoseconds rather than seconds.

As there has been a large drift in the on-disk layout of Linux file systems from Extended file system to Extended 2 file system while later versions have support Ext2 on-disk layout, we will review only Ext and Ext2 on disk layout in detail.

Extended File System is based on the concepts derived from UNIX operating system. In Extended File Systems, every file is represented by an I-node (Index Node), everything is a file, directory which is a special file contains list of entries pertaining to files it contains along with corresponding I-node. When a volume is formatted with Extended File System, 4 data structures are created as shown in figure 5.

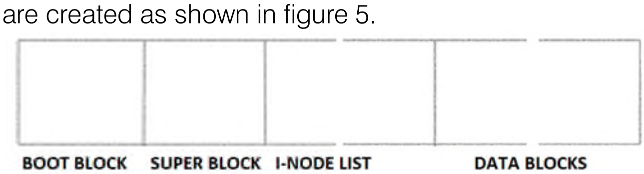


Figure 5. Data Structures of Extended File System

a) *Data Blocks*

Data Blocks immediately follow the I-node list and occupy rest of the volume. A data block is a set of consecutive sectors which is allocated to a file in its entirety. They are internally represented by numbers corresponding to their position in the volume. A file may be allocated one or more data blocks, consecutive or fragmented over the volume.

b) *I-node List*

I-node list structure immediately follows the Super block. The size of I-node list depends upon the volume size and is calculated at initial format and punched in Super block. I-node is the basic building block; every file and directory in the file system is described by one and only one I-node. Each I-node contains the description of the file it represents; file type, access permissions, owner, access times, link count, file size and table of pointers to data blocks. I-nodes are internally represented by I-node number enumerated by their position in the I-node list. The numbering begins from 1, I-node 0 does not exist on newly formatted volume. An I-node of Type=0 and number of links=0, is free otherwise represents a file.

The table of pointers to data blocks is an array of entries where first 9 direct entries contain the address (index number) of data blocks containing data of the file while the next single indirect entry contains the address of data block that contains the direct entries for data blocks containing the data of the file. The next entry in table is a double indirect entry that points to a data block which contains single indirect entries. Similarly, a triple indirect entry in table points to a data block that contains double indirect entries. This level of indirection is used to allow the structure of I-node to be small but at the same time allows large file size to be addressed. This scheme is shown in figure 6.

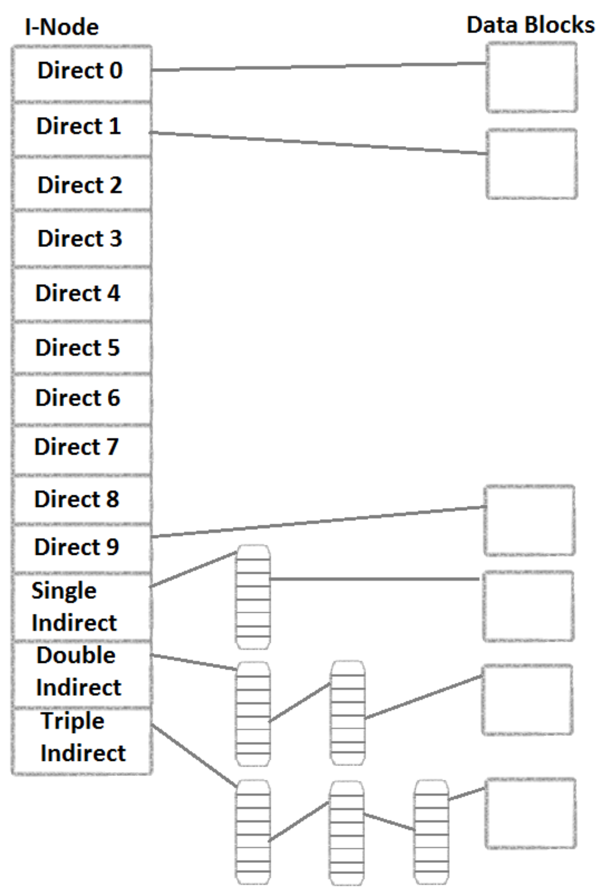


Figure 6. Levels of indirection to address data blocks.

Several block entries in I-node can be 0 meaning that logical block entries contain no data. This happens if no process ever wrote data into the file offsets corresponding to those blocks and hence block numbers remain at their initial value 0. This way Extended File System supports Sparse files.

Byte Offset in Directory	I-Node Number	File Names
0	83	.
16	2	..
32	1798	init
48	1276	fsck
64	85	mount
80	1268	passwd

Figure 7. A typical directory file content.

Directories are implemented as special files containing a list of fixed sized entries. Each entry contains I-node number and fixed length filename it represents. Any entry that contains 0 in I-node but has some valid filename represents a deleted file that existed previously on the volume. Every directory file has first 2 entries containing '.' and '..' entry representing its I-node number and parent directory's I-node number respectively. For root '/' directory both entries have same value. A typical directory file content is shown in figure 7.

c) *Boot Block & Super Block*

The Boot block is located at first sector of volume and contains the boot strap code. The Super block immediately follows the Boot block and contains the information that describes the state of a file system. The information contained in Super block includes:

- Size of the file system,
- Number of free blocks in the file system,
- A list of free blocks in the file system,
- Index of next free block in the free block list,
- Size of I-node list,
- Number of free I-nodes in file system,
- List of free I-nodes in file system,
- Index of next free I-node in free I-node list,
- Lock fields for free block and free I-node list, and
- Flag indicating that Super block has been modified.

Extended file system stores in Super block information that is needed to maintain I-nodes and data blocks. When the volume is created Super block list of free I-nodes is empty and kernel searches the I-node list structure for those I-nodes where the Type=0 and populates the list to its full capacity remembering the highest numbered I-node it finds. The next time the kernel searches the disk for free I-nodes, it uses this remembered I-node as its starting I-node. Keeping track of I-nodes is easy but the list is used to avoid the I-node list search every time an I-node is needed as free I-nodes can be located in I-node list any time by searching for type field. The data blocks are necessarily to be maintained in their entirety because there is no way for kernel to know on the basis of the content they contain that whether the data block is free or allocated.

The Super block contains the list of free blocks populated at the time of volume creation. The data blocks are organized in a linked list fashion. The Super block list contains the list of free blocks to its capacity. One entry in the list points to a block that contains such kind of a list to its capacity. During volume creation, the kernel tries to organize the list in such a manner such that block numbers allocated to a file are nearby but later on no such effort is made. The structure of metadata about the free data blocks is shown in figure 8.

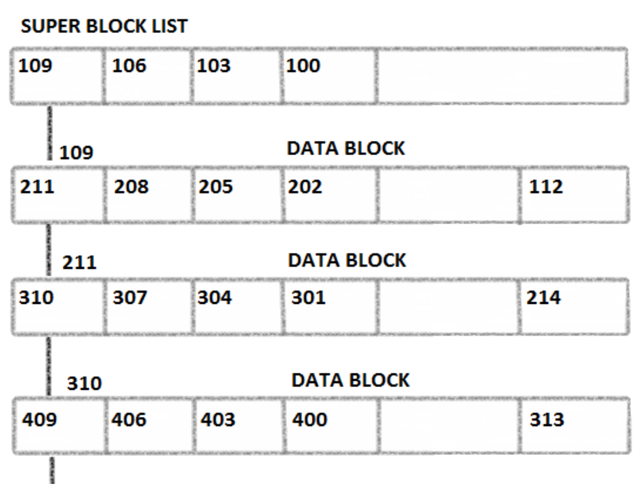


Figure 8. Free data block management.

Global Journal of Computer Science and Technology Volume XI Issue VI Version I April 2011

Extended 2 file system on-disk layout is strongly influenced by BSD file system and is almost similar to Extended file system. Ext2 file system is divided into block groups, which contain a fixed number of blocks where blocks are fixed sized number of sectors. Block groups immediately follow the boot sector and are numbered from 0 onwards. Every block group contains a Super block (1 block in size), Group descriptors (n blocks in size), Data block bitmap (1 block in size), I-node bitmap (1 block in size), I-node table (n blocks in size) and data blocks (n blocks in size). The typical structure of Ext2 file system is shown in figure 9.

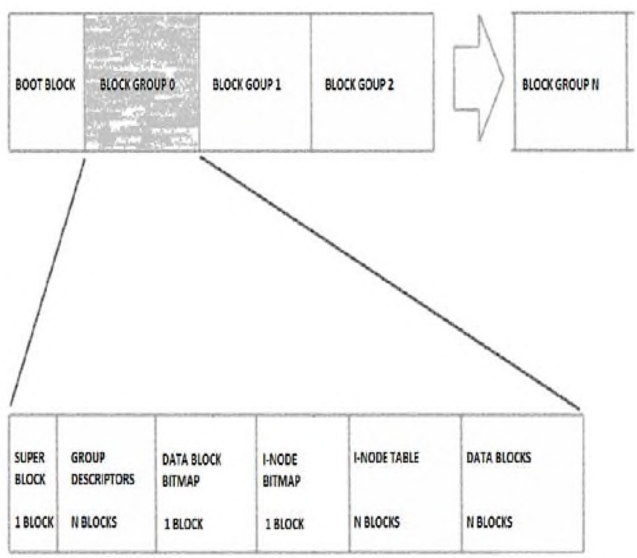


Figure 9. Ext2 data structures.

Using block groups has 3 advantages:

- Each block group contains a redundant copy of Super block and block group descriptors that actually define the file system. As such, it is easy to recover if any Super block gets corrupted.
- This arrangement gives good performance by reducing the distance between the I-node table and the data blocks which reduces the head seeks during file I/O.
- It reduces fragmentation by keeping the data blocks belonging to a file in same block group.

d) *Super Block*

The Super block of Ext2 contains following information:

- Magic number which validates whether the block is Super block or not.
- Revision level which indicates features it supports.
- Mount count and maximum mount count.
- Block group number that holds this copy of Super block.
- Block size fixed at volume creation.
- Blocks per group fixed at volume creation.
- Free blocks which indicates number of free blocks.
- Free I-nodes which indicates number of free I-nodes.
- First I-node which indicates the root '/' I-node.

Ext2 Super block does not contain information regarding the list of free data blocks and I-nodes. This information is individually maintained by Block bitmap and I-node bitmap of block group.

e) *Block Group Descriptor*

Block group descriptor consumes one block and contains following information:

- The block number of block allocation bitmap for this block group used during block allocation and de allocation.
- The block number of I-node allocation bitmap for this block group used during I-node allocation and de allocation.
- I-node table which contains the starting block number of I-node table for this block group.
- Number of free blocks in group.
- Number of free I-nodes in group.
- Number of directories in group.

Only the first copy of Super block and group descriptors is updated by Ext2 file system while for other block groups it is left untouched. When a consistency check is executed, the information is copied on other block groups.

f) *Block & I-node Bitmaps*

Both of these bitmaps occupy one block each and number of blocks they address depends upon fixed number of blocks per group. In these bitmaps, each bit corresponds to a block (or I-node) of group and its state indicates whether it is allocated or not.

g) *I-node Table*

I-node table is an array of fixed sized I-nodes and occupy many blocks depending upon the size of I-node, total number of I-nodes in a group and block size all indicated by Super block.

Ext2 I-node is almost same as that of Extended I-node in that it uses multiple levels of indirection but Ext2 directories contain variable length entries unlike Ext file system directory. Each directory entry contains I-node number, name length and name of file.

Ext3 on-disk data structures are identical to those of an Ext2 file system. As a matter of fact, if an Ext3 file system has been cleanly un-mounted, it can be remounted as an Ext2 file system, conversely, creating a journal of Ext2 file system and remounting it as Ext3 is simple and fast operation.

V. Hierarchical File Systems

Macintosh File System (MFS) was introduced around 1983 with first Mac computer. MFS was optimized to be used on very small and slow media [35]. With the introduction of larger media, the time taken to display the contents of a folder was a concern as MFS used a single flat file to store all of the file and directory listing information. As such, the system had to do a complete search of this file in order to build a list of files stored in a particular folder.

Hierarchical File System (HFS), also called Mac OS Standard, was introduced in 1985 to mitigate this problem. HFS replaced the flat file of MFS with Catalog File which uses B-tree structure that could be searched very quickly regardless of size. HFS was introduced with 20 MB hard disk drive and was hard coded into 128 KB ROM. HFS file system divides the volume in 512 bytes long sectors and allocates to files allocation blocks which contain one or more consecutive sectors. HFS contains 5 data structures that make up the volume:

- Boot blocks occupy sector 0 and 1 of system and contain system startup information.
- Master Directory Block (MDB) occupies sector 2 and defines the volume layout and other information like location and size of other structures. MDB is duplicated at opposite end of the volume in second to last sector. This is used to recover the volume in case of corruption and is only updated only when either Catalog file or Extent Overflow file size increases.
- Volume Bitmap starts at sector 3 and keeps track of which allocation blocks are free. The size of Volume bitmap depends upon the size of the volume.
- Catalog file is a B-tree that contains records for all files and folders which exist on the volume. Files and folders are uniquely identified in Catalog file by Catalog Node ID (CNID). Each node represents a file or folder and may contain any 2 types of records among the 4 possible types. For a file node, a File Thread Record stores filename and CNID of its parent directory and a File Record stores 16 byte attributes used by Finder, timestamps, its CNID, first 3 extents of file for both data and resource fork, and pointer to first data and resource fork extent records in Extent Overflow file (in case it has any). For a directory node, a Directory Thread Record stores name of directory and CNID of parent directory and a Directory Record stores 16 byte attributes used by Finder, timestamps, its CNID and number of files stored in it.
- Extent Overflow file is a B-tree structure file that contains extra extents pertaining to any file if the initial 3 extents of that file record in Catalog file are used up. Later versions allowed bad blocks to be recorded as extents.

An extent is a contiguous range of allocation blocks allocated to some fork, represented by a pair of numbers; the first allocation block number and number of allocation blocks.

The general on disk layout of HFS file system is shown in figure 10.

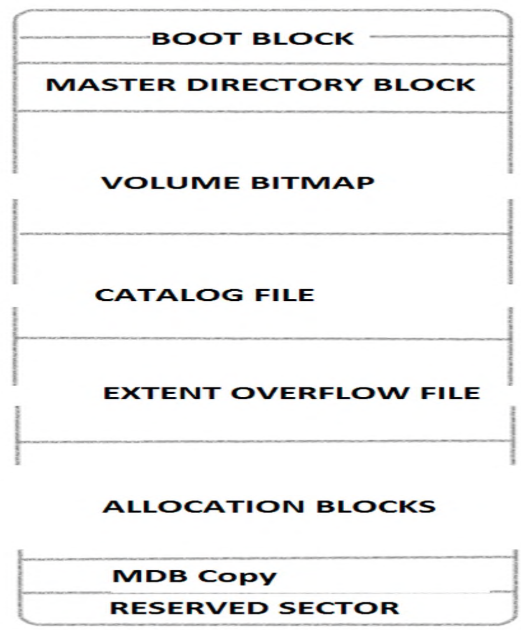


Figure 10. HFS On-Disk Layout.

Under HFS (also in HFS+) files are not monolithic and do not consist of one single element [36]. They may be composed of two or more pieces, called Forks. NTFS also supports this concept by supporting multiple data streams in general and multiple values for same attribute types identified by names. HFS files have 2 named forks (Data & Resource) and can have logically any number of un-named forks. A Data fork contains the actual data pertaining to the file like text for word processor, etc. A Resource fork contains metadata pertaining to the file like icon, preview picture, etc. In other words, Data fork is used to store the unstructured data while Resource fork is used to store the structured data. The Resource fork was designed to store metadata that would be used by GUI. HFS+ supports any arbitrary number of custom named forks in addition to data & resource forks.

As the Catalog file stores all the file and directory records in single data structure, only one program can write to this structure at a time, forcing other programs to wait in a queue to get their turn. This raises both a performance and reliability issue. Also, due to 16 bit pointers used to address allocation blocks, HFS is able to address only 65535 allocation

April 2011
13
Global Journal of Computer Science and Technology Volume XI Issue VI Version I

blocks. This means, a minimum size of allocation block can be 1/65535th of volume size. This means only 65535 files are possible and high internal fragmentation on large volumes.

Hierarchical File System Plus (HFS+), also called Mac OS Extended, was introduced in 1998 to overcome problems of HFS and has become the primary file system used in Mac computers [37]. HFS+ is an improved version of HFS supporting larger files and volumes by using 32-bit allocation block addresses and Unicode for filenames. It also supports multiple named forks for files, Journaling, inline attribute data records, access control list based file security and compatibility with file permission models on other platforms such as Windows.

Like HFS, HFS+ divides volume into 512 byte sectors and groups them into allocation blocks (usually 8) to be allocated to a file. Allocation blocks are addressed by 32-bit pointers [38]. In HFS+ volume everything is a part of one or more allocation blocks with possible exception Alternate Volume Header, unlike HFS were Boot blocks, Master Directory Block and Volume Bitmap are not part of any allocation block. To reduce file fragmentation, contiguous allocation blocks called Clumps are allocated to files. The number of allocation blocks per Clump is fixed and is specified in Volume Header. The first 1024 bytes and last 512 bytes of volume are reserved. The Volume Header is located immediately after first 1024 bytes and is fixed. The Alternate Volume Header which is replica of Volume Header is located at 1024 bytes before the end of volume and is also fixed. The on-disk layout of HFS+ volume is shown in figure 11.

Volume Header is equivalent of Master Directory Block of HFS. It stores timestamps, number of files on volume, location of other structures on volume, size of allocation blocks, size of clumps, etc. When a volume is formatted with HFS+ file system, it leads to the creation of 5 special files in addition to reserved allocation blocks, Volume Header and Alternate Volume Header.

a) *Allocation File*

Allocation file keeps track of which allocation locks are free and which are in use by representing every block by bit. It is equivalent to Volume Bitmap of HFS. The main difference between Volume Bitmap and Allocation File is that Allocation file is a regular file which can exist anywhere on volume, shrink or grow in size and need not to be contiguous while Volume Bitmap always resides in reserved area and its size is fixed. The location of first extent of Allocation file is stored in Volume Header. This architecture of Allocation file induces flexibility in HFS+ file system not found in HFS.

b) *Catalog File*

Catalog file describes every file and folder of the volume including the special files and the hierarchy in the volume. It is similar to Catalog file of HFS. The Catalog file is organized as a B-tree to allow quick and efficient searches through a large hierarchy. This file contains vital information about every file and folder along with the catalog information. The main difference between the records in HFS and HFS+ Catalog file is that in HFS+ the nodes of B-tree pertaining to files and folders contain more information and can have varying size unlike HFS. The location of first extent of Catalog file is stored in Volume Header. Catalog file contains Header node, Index nodes, Leaf nodes and if necessary Map nodes. Each file or folder in Catalog file is given a unique Catalog Node ID (CNID). For folder, CNID is called FolderID and for files FileID. Like HFS Catalog nodes, HFS+ Catalog nodes also store File Record and File Thread Record for files and Folder Record and Folder Thread Record for folders in addition to some more additional information. The main difference between HFS File Record and Directory Record and HFS+ File Record and Folder Record is that in HFS the records contain information about first 3 extents belonging to the file or folder while in HFS+ it is 8.

c) *Extent Overflow File*

Special files only have one fork i.e. Data fork. The Catalog file does not store any extent for special files rather first 8 extents of special files are stored in Volume Header. User files can have both data and resource fork and if necessary other named forks. The first 8 extents of both data and resource forks for user files are stored in Catalog file. In both types of files, if there is need for additional extents for data and resource fork and/or for named forks, the extents are recorded in Extent Overflow file. It is a B-tree structured file that stores standard additional forks' extents and named forks' extents for user files. It does not store for itself any additional data fork extent.

d) *Bad Block File*

Bad Block file is used to mark and record the areas of the volume that contain bad blocks. The Extent Overflow file is used to hold information about the Bad Block file extents.

e) *Attributes File*

An Attributes file is a special file which does not have an entry in Catalog file. An Attributes file is a complex file. A volume can have no Attributes file in which case its description in Volume Header for allocation blocks is 0. Attributes file is a B-tree structured file where nodes can contain records known as Attributes. An Attributes file can have 3 types of attributes:

- Inline Data Attributes which contain small attributes.
- Fork Data Attributes which contain references to a maximum of 8 extents.

- Extended Attributes which contain references to 8 more extents for data attributes.

f) *Startup File*

Startup file is a special file used to hold information needed when booting a system that does not have built-in ROM support for HFS plus. The boot loader can find the location of Startup file from Volume Header which contains the first 8 extents of Startup file. Startup file should not have any additional extents for data fork as it will complicate things for boot loader.

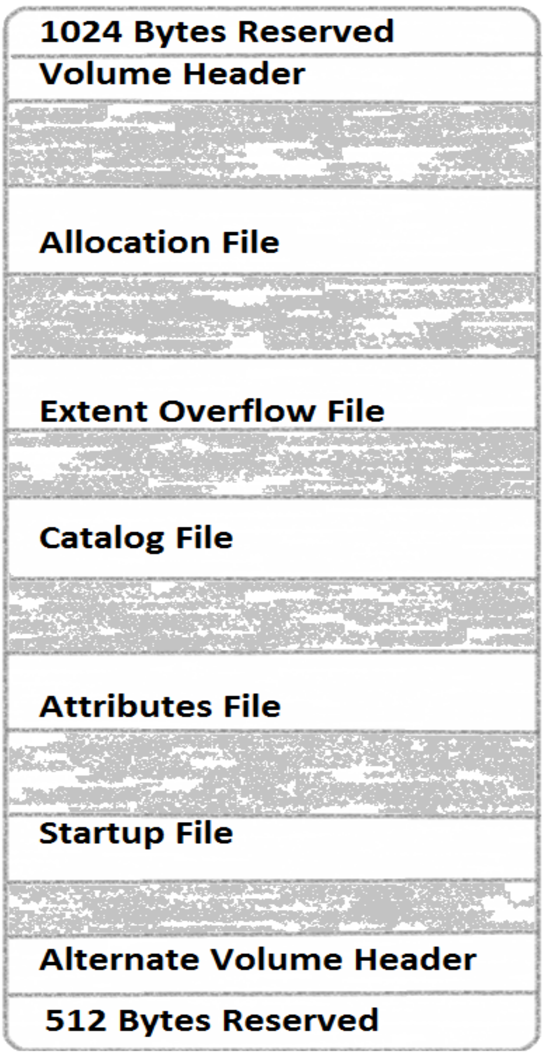


Figure 11. HFS+ On-Disk Layout.

VI. Discussion And Conclusion

We observed that the on-disk layout of file systems reviewed in this paper were objective specific. In case of FAT file systems, the new versions were developed to address the issue of large file size and large volume size support. Similarly in case of Hierarchical file systems; the augmented versions addressed Unicode support in filenames, relocatable system metadata structures and large file and volume size. In both cases, the actual design remained the same. We also observed, in case of NTFS that the design was drafted from scratch which yielded into an elegant file system having almost all features which a modern file system should have. Further, in case of Extended file systems, we observed large drift in on-disk layout from Extended file system to Extended 2 file system to increase performance and reliability. Again, the design of Extended 3 file system which is mount compatible with Extended 2 file system is an excellent example of flexibility in design of Extended 2 file system. We also observed some similarity in heterogeneous file systems. The concept of treating everything residing on the volume as a file is the basic building block of both NTFS and Hierarchical file systems.

References Références Referencias

1. Grochowski, E. (1998), "Emerging trends in data storage on magnetic hard disk drives", Datatech, pages 11–16, Sep 1998.
2. Dahlin, M.D. (1996), "The Impact of Trends in Technology on File System Design", University of California, Berkeley.
3. Gibson, G.A. (1992), "Redundant Disk Arrays: Reliable, Parallel Secondary Storage", ACM Distinguished Dissertations. MIT Press, Cambridge, Massachusetts.
4. Giampaolo, D., "Practical File System Design with the Be File System", Be, Inc.
5. Zadok, E., Iyer, R., Joukov, N., Sivathanu, G. and Wright, C.P. (2006), "On Incremental FileSystem Development", ACM Transactions onStorage (TOS),2(2):161–196, May 2006
6. DEC Tape, http://en.wikipedia.org/wiki/DECtape, Accessed on November 2010.
7. http://www.pdp8.net, Accessed on November 2010
8. http://en.wikipedia.org/wiki/Gary_Kildall, Accessed on November 2010.
9. http://en.wikipedia.org/wiki/PL/M, Accessed on November 2010.
10. http://www.digitalresearch.biz/cpm.htm, Accessed on November 2010.
11. http://en.wikipedia.org/wiki/Tim_Paterson, Accessed on November 2010.
12. http://en.wikipedia.org/wiki/86-DOS, Accessed on November 2010.
13. "The Man Who Could Have Been Bill Gates", http://www.businessweek.com/magazine/content/04_43/b3905109_mz063.htm, Accessed on November 2010.
14. FAT32 File System Specification, http://microsoft.com/whdc/system/platform/firmware/fatgen.mspx, Accessed in 2009.
15. Bhat, W.A., Quadri, S.M.K., (2010), "Review of FAT Data Structure of FAT32 file system",

Oriental Journal of Computer Science & Technology, Volume 3, No 1.
16. Extended FAT File System, http://msdn.microsoft.com/enus/library/aa914353.aspx, Accessed on November 2010.
17. Kwon, M.S., Bae, S.H., Jung, S.S., Seo, D.Y. and Kim, C.K., (2005), “KFAT: Log-based Transactional FAT File system for Embedded Mobile Systems”, In Proceedings of 2005 US-Korea Conference, ZCTS-142, 2005.
18. Alei, L., Kejia, L., Xiaoyong, L., (2007), “FATTY : A reliable FAT File System”, In Proceedings of the 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, Pages: 390-395, 2007.
19. Microsoft Corporation, “Transaction-Safe FAT File System”,http://msdn2.microsoft.c0m/en-us/library/aa911939.aspx, Accessed in 2010.
20. Duncan, R., (1989), “Design goals and implementation of the new High Performance File System”, Microsoft Systems Journal, September 1989 v4 n5 p1 (13).
21. Russinovich, M., Solomon, D.A. and Ionescu, A. (2009), “File Systems”, Windows Internals (5th edition), Microsoft Press. ISBN 0735625301.
22. Nagar, R., (1997), “Windows NT File System Internals : A Developer's Guide”, O'Reilly. ISBN 9781565922495.
23. NTFS Concepts http://www.priscilla.com/Courses/ComputerForensics/pdfslides/03-NTFSConcepts.pdf, Accessed on November 2010
24. NTFS Documentation, http://www.scribd.com/doc/2187280/NTFS-Documentation, Accessed on November 2010
25. http://www.ntfs.com, Accessed on November 2010.
26. Probert, D.B., “Windows Kernel Development”, Microsoft Corporation, http://i-web.i.u-tokyo.ac.jp/edu/training/ss/lecture/new-documents/Lectures/08-NTFS/NTFS.ppt, Accessed on November 2010
27. http://www.minix3.org/, Accessed on November 2010.
28. http://en.wikipedia.org/wiki/MINIX_file_system, Accessed on November 2010.
29. The Virtual File System in Linux, http://www.linux.it/~rubini/docs/vfs/vfs.html, Accessed on November 2010
30. Bach, M.J. (1986), “The Design of the UNIX Operating System”, Prentice Hall, 1986.
31. Card, R., Ts'o, T. and Tweedie, S., (1994), “Design and Implementation of the Second Extended Filesystem”, In Proceedings of the First Dutch International Symposium on Linux, Amsterdam, Holland, 1994.
32. McKusick, M.K., Joy, W.N., Leffler, S.J. and Fabry, R.S., (1984), “A Fast File System for UNIX”, In ACM Transactions of computer Systems, Vol 2, No. 3, 1984.
33. Tweedie, S. (1998), “Journaling the Linux ext2fs filesystem”, In LinuxExpo '98,1998.
34. Ts'o, T. (2006), “Proposal and plan for ext2/3 future development work”. Linux kernel mailing list. http://lkml.org/lkml/2006/6/28/454
35. HFS,http://en.wikipedia.org/wiki/Hierarchical_File_System, Accessed on November 2010.
36. HFS Plus, http://en.wikipedia.org/wiki/HFS_Plus, Accessed on November 2010.
37. TN1150, “HFS Plus Volume Format”, http://developer.apple.com/library/mac/#technotes/tn/tn1150.html, Accessed on November 2010
38. Mac OS X: Mac OS Extended format (HFS Plus) volume and file limits, http://support.apple.com/kb/HT2422, Accessed on November 2010
39. Overview of FAT, HPFS, and NTFS File Systems, http://support.microsoft.com/kb/100108, , Accessed on November 2010.
40. Daily, S. (1996), “NTFS vs. FAT.” Windows NT Magazine October 1996: 95.
41. NTFS Directories and Files, http://www.pcguide.com/ref/hdd/file/ntfs/files.htm, Accessed on November 2010.
42. Janes, M., “Progression of Linux File Systems”, http://mjanes.public.iastate.edu/Engl314/indiv_doc.pdf
43. Anjoy, R.G., Chakraborty, S.K., (2009), “Feature Based Comparison of Modern File Systems”, http://www.idt.mdh.se/kurser/ct3340/ht09/ADMINISTRATION/IRCSE09_submissions/ircse09_submission_16.pdf
44. Mitchell, S. (1997), “Inside the Windows 95 File System”, O'Reilly. ISBN 156592200X.
45. Tanenbaum, A.S., Woodhull, A.S., (2006). “File Systems”, Operating Systems: Design and Implementation (3rd edition.). Prentice Hall. ISBN 0131429388.
46. Pate, S.D. (2003). “UNIX Filesystems: Evolution, Design, and Implementation”, Wiley. ISBN 0471164836.
47. Leffler, S.J. and McKusick, M.K., “The design and implementation of the 4.3BSD UNIX operating System Answer Book”, Addison-Wesley, ISBN 0201546299
48. Bar, M., “Linux File Systems”, McGraw-Hill, ISBN 0072129557

49. Bovet, D.P. and Cesati, M. (2005), "Understanding the Linux Kernel",O´Reilly Media, 3rd edition, 2005. ISBN 0596005652.
50. Silberschatz, A., Galvin, P.B. and Gagne, G., (2004), "Storage Management", Operating System Concepts , 7th Edition, Wiley. ISBN 0471694665.

This page is intentionally left blank

Global Journal of Computer Science & Technology
Volume 11 Issue 6 Version 1.0 April 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

A Survey on Security Analysis of Routing Protocols

By J.Viji Gripsy , Dr. Anna Saro Vijendran

Abstract- : Mobile ad hoc networking (MANET) is gradually emerging to be very important in the growth of wireless technology. This is anticipated to offer a range of flexible services to mobile and nomadic users by means of integrated homogeneous architecture. The proper routing protocol is necessary for better communication in MANET. One of the existing reliable protocols is Ad Hoc On-Demand Vector Routing (AODV) protocol which is a reactive routing protocol for ad hoc and mobile networks that maintains routes only between nodes that wants to communicate. There are various security issues to be considered in this protocol. In order to provide security for AODV protocol, Secure Ad Hoc On-Demand Vector Routing (SAODV) can be used. SAODV is an extension of the AODV routing protocol that can be used to shield the route discovery process by providing security characteristics like integrity and authentication. For secure protocol, digital signature, hash chains, etc., can be used in routing. This paper surveys on various techniques available for securing the mobile ad hoc network.

Keywords: Mobile Ad-hoc Network, Routing Protocols,AODV protocol, SA-AODV protocol

Classification: GJCST Classification: C.2.2

A SURVEY ON SECURITY ANALYSIS OF ROUTING PROTOCOLS

Strictly as per the compliance and regulations of:

A Survey on Security Analysis of Routing Protocols

J.Viji Gripsy[α], Dr. Anna Saro Vijendran[Ω]

Abstract- Mobile ad hoc networking (MANET) is gradually emerging to be very important in the growth of wireless technology. This is anticipated to offer a range of flexible services to mobile and nomadic users by means of integrated homogeneous architecture. The proper routing protocol is necessary for better communication in MANET. One of the existing reliable protocols is Ad Hoc On-Demand Vector Routing (AODV) protocol which is a reactive routing protocol for ad hoc and mobile networks that maintains routes only between nodes that wants to communicate. There are various security issues to be considered in this protocol. In order to provide security for AODV protocol, Secure Ad Hoc On-Demand Vector Routing (SAODV) can be used. SAODV is an extension of the AODV routing protocol that can be used to shield the route discovery process by providing security characteristics like integrity and authentication. For secure protocol, digital signature, hash chains, etc., can be used in routing. This paper surveys on various techniques available for securing the mobile ad hoc network.

Keywords: Mobile Ad-hoc Network, Routing Protocols, AODV protocol, SA-AODV protocol

I. Introduction

An ad hoc network is usually defined as an infrastructureless network. This means that a network is lacking the standard routing infrastructure like fixed routers and routing backbones. Usually, the ad hoc nodes are mobile and the fundamental communication medium is wireless. Every ad hoc node possibly will be able to of act as a router. Such ad hoc networks may arise in personal area networking, meeting rooms and conferences, disaster relief and rescue operations, battlefield operations, etc.

By considering the special characteristics of MANET, designing a well-organized and dependable routing protocol strategy is a huge challenge. Currently, various ad hoc routing protocols have been proposed and developed by various researchers like DSDV, OLSR, TBRPF, AODV, DSR and ZRP. From all these, Ad-hoc On-demand Distance Vector (AODV) is recognized as one of the main IETF standards for MANET routing. On the other hand, AODV aims on improving routing performance, but provides only slight consideration to routing security, which indicates that it is susceptible to various attacks from malicious, compromised and selfish nodes.

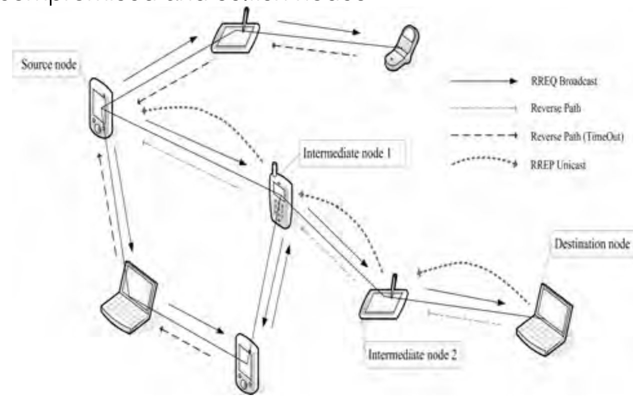


Figure 1: Route Discovery Procedure of AODV Protocol

AODV protocol is a reactive routing protocol for ad hoc and mobile networks. This represents that AODV will not perform any action until a node requires broadcasting a packet to a node for which it does not know a route. In addition, it only maintains routes among nodes that require communicating. Its routing messages do not enclose information about the entire route path, but simply regarding the source and the destination. Hence, routing messages have a constant size, independently of the number of hops of the route. It utilizes destination sequence numbers to indicate how fresh a route is that is used to grant loop freedom.

In AODV, a node performs route identification by flooding the network with a 'Route Request' message (RREQ). When it arrives a node that knows the requested route, it reply with a 'Route Reply' message (RREP) that goes back to the creator of the RREQ. Next, all the nodes of the identified path have routes to both ends of the path. Beside these routing messages, 'Route Error' messages (RERR) are utilized to alert the other nodes that several nodes are not any longer reachable because of link breakage. The route discovery procedure of AODV protocol is provided in figure 1.

But AODV lacks security features which lead to great vulnerability for attacking. To provide security for AODV, Secure Ad Hoc On-Demand Vector Routing (SAODV) s used which focuses on using various techniques like digital signature, hash chains, etc., This paper focuses on analyzing various security enabled protocols for providing better security for MANET.

About[α]- Assistant Professor, Ph.D Research Scholar , Department of Computer Science, PSGR Krishnammal College for Women,Pursuing my Phd in SNR SONS COLLEGE ,COIMBATORE,INDIA.
Email : gripsyjebs@gmail.com
About[Ω]- Director, Department of Computer Applications SNR SONS College, Coimbatore, India.

II. Literature Survey

As AODV lacks security mechanisms, malicious nodes can carry out several attacks just by not behaving based on the AODV rules. Therefore, to guarantee the entire security of the network, it is important to create security mechanisms that can withstand malicious attacks from insiders who have entire control of several nodes. For the purpose of protection against insider attacks, it is required to realize how an insider can attack a wireless ad-hoc network. Various attacks have been discussed in various literatures. According to the composition of operations for carrying out attack as mentioned in above article, misuses of AODV have been divided into two categories: atomic misuses and compound misuses. Intuitively, atomic misuses are carrying out by controlling a single routing message that cannot be any more separable. On the contrary, compound misuses are composed of multiple atomic misuses, and possibly normal uses of the routing protocol. Initially, it is required to determine a number of misuse goals that an inside attacker may require to achieve and are listed as follows.

Route Disruption (RD): Route Disruption is nothing but either breaking down an existing route or preventing a new route from being created.

Node Isolation (NI): Node isolation indicates the preventing of a provided node from communicating with any other node in the network. It varies from Route Disruption in that Route Disruption is targeting at a route with two provided endpoints, while node isolation is intended at every possible routes.

Route Invasion (RI): Route invasion means that an inside attacker adds itself into a route between two endpoints of a communication channel.

Resource Consumption (RC): Resource consumption is nothing but consuming the communication bandwidth in the network or storage space at every nodes. For example, an inside attacker may consume the network bandwidth by either forming a loop in the network. As an example, route disruption, route invasion and node isolation has been shown diagrammatically using figure 2, 3 and 4 respectively.

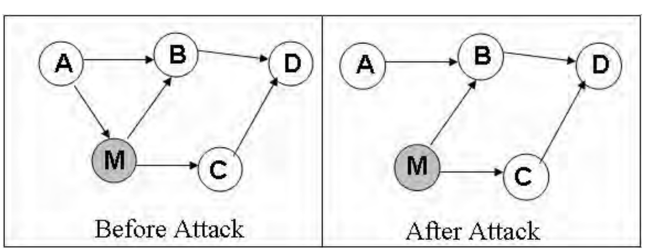


Figure 2: Node M performing Route Disruption for path A-C

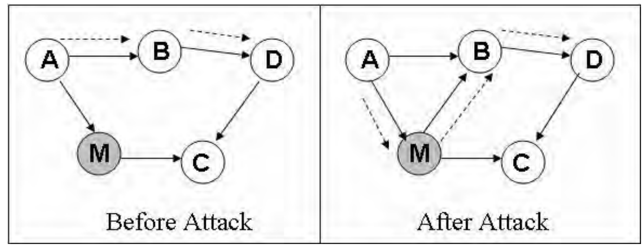


Figure 3: Route invasion

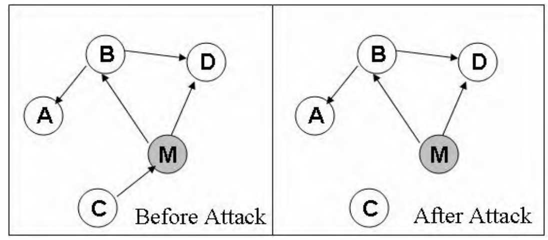


Figure 4: Node isolation

Investigation of atomic misuses can be carried out in an effective manner by means of understanding the causes of probable atomic misuse actions. All atomic misuse action is an inseparable manipulation of one routing message. In particular, the atomic misuse actions in AODV have been divided into the following four categories:

Drop (DR): Here, the attacker just drops the received routing message.

Modify and Forward (MF): Once the routing message is received, the attacker alters one or more fields in the message and then forwards the message to its neighbors through unicast or broadcast.

Forge Reply (FR): The attacker sends a faked message in reply to the received routing message. Forge Reply is generally related to the misuse of RREP messages that are in response of RREQ messages.

Active Forge (AF): The attacker sends a faked routing message without receiving any associated message.

The most interesting and complex one is that an attacker can merge several atomic misuses in a planned way and launch them.

Perlman [8] provides a link state routing protocol that attains Byzantine Robustness. Even though the protocol is greatly robust, it needs a very high overhead linked with public key encryption. Secure BGP [9] aims to protect the Border Gateway Protocol by using PKI (Public Key Infrastructure) and IPsec.

Zhou *et al.,* [10] primarily discuss key management for securing ad hoc networks. The author offers a section to secure routing, but basically conclude that nodes can shield routing data in the same manner they shield data traffic. They also examine that denial-of-service attacks against routing will be considered as damage and routed around.

Dahill *et al.,* [11] presented ARAN, a routing protocol for ad hoc networks that utilizes authentication and needs the utilization of a trusted certificate server. In ARAN, each node that forwards a route discovery or a route reply message should also sign it. Additionally, it is implemented to reply attacks by means of error messages if not the nodes contain time synchronization.

Papadimitratos *et al.,* [12] put forth a protocol (SRP) that can be used to various available routing protocols particularly DSR and IERP. SRP needs that, for all the route discovery, source and destination should contain a security association among them. In addition, the author does not even refer to route error messages. Hence, they are not sheltered, and any attackable node can just create error messages by considering other nodes as source.

Hash chains are utilized in better manner for obtaining good authentication in various techniques that attempts to protect routing protocols. In [13], [14] and [15], hash chains are used for the purpose of providing delayed key disclosure. Whereas in [16], hash chains are utilized to generate single-time signatures that can be checked instantly.

In SEAD, hash chains are utilizes in grouping with DSDV-SQ [18] by Hu *et al.,* [17]. For each provided time all node contains its individual hash chain. The hash chain is separated into segments; elements in a segment are helpful in securing hop counts in a same manner as it is performed in SAODV. The size of the hash chain is identified when it is created. Following the usage of every elements of the hash chain a new one should be calculated. SEAD can be utilized with some appropriate authentication and key distribution techniques. However determining such a technique is not simple. Brijesh [19] discusses attacks against distance vector routing protocols and describes techniques to secure them using Message Authentication Codes.

Songbai *et al.,* [1] proposed a SAODV which is a MANET routing protocol that can withstand black hole attack. AODV is a broadly used network routing protocol for MANETs. The propose of AODV shows some concentration to security issues, therefore consequential in the defenselessness of such MANET to the black hole attack. Based on AODV, the author suggested and realizes AODV suffering black hole attack - BAODV (Bad Ad Hoc On-demand Distance Vector Routing suffering black hole attack) that can imitate black hole attack to MANET by one of nodes as a malicious one in network. BAODV can be considered as AODV that is utilized in MANET which suffers from black hole attack. According BAODV, author also presents a secure and effective MANET routing protocol, the SAODV protocol that focuses on dealing with the security flaws of the AODV protocol and is accomplished to overcome the black hole attack.

Papadimitratos *et al.,* [2] deal with the security issues of route discovery in mobile ad hoc networks, providing a lightweight, however robust, routing protocol, the distance-vector secure routing protocol (DV-SRP). DV-SRP identifies on-demand multiple routes that are utilized among the network, devoid of clearly offering network connectivity. DV-SRP merges the merits of the kind of route discovery initially provided by AODV with security and therefore flexible to opposition that interrupt route discovery.

Pirzada *et al.,* [3] proposed a secure routing with the AODV Protocol. Because of their enhanced nature, ad-hoc networks are often utilized in non-secure situation that formulates them vulnerable to attacks. These attacks are offered by chipping in the malicious nodes adjacent to various network services. Routing protocols that work as the necessary force in these networks are a general target of these nodes. AODV is the commonly utilized routing protocols that are presently experiencing extensive research and development. AODV is in accordance with the distance vector routing, excluding the updates are shared not based on a periodic origin but on an as per accordance with the needs. The control packets enclose a hop-count and sequence number field that finds the freshness of routing updates. Since these fields are changeable, it generates a possible weakness that is often exploited by malicious nodes to advertise good routes. Likewise, broadcasting of routing updates in clear text also reveals vital data about the network topology that is once more a possible security hazard. The author provides a novel and pragmatic technique for securing the ad-hoc on-demand distance vector routing protocol that guards against a number of attacks performed against mobile ad-hoc wireless networks.

Sanzgiri *et al.,* [4] proposed authenticated routing for ad hoc networks. Initially, only the issue of offering effective techniques for finding paths in very dynamic networks was considered, without considering security. Since security is not considered, there are a various treats that can be used to influence the routing in an ad hoc network. The author describes these threats in this paper, particularly explaining their effects on ad hoc on-demand distance vector and dynamic source routing. Authenticated Routing for Ad hoc Networks (ARAN) protocol is proposed in this approach which uses public-key cryptographic techniques to counter all the attacks. ARAN can provide secure routing in environments where nodes are authorized to participate but in situations where participants are not to be authorized, it does not respond. The simulation and experimentation of the proposed ARAN clearly shows that the performance of the proposed approach is very significant in finding secure routes within an ad hoc network.

Khan *et al.,* [5] provided a security Adaptive Protocol Suite: Ranked Neighbor Discovery (RND) and Security Adaptive AODV (SA-AODV). Due to the raise in popularity and demand of mobility and ad hoc networking, weakness of wireless networks is also becoming a crucial issue. This study focuses on the security aspects of wireless communication, and proposes a technique with an enhanced security features. The proposed RND and SA-AODV routing protocol provides best solution for the security problems the neighbor discovery and the routing protocol for transmission are also included in this proposed approach. Based on distance metrics, the neighbor discovery phase contains the determination of trusted neighbors, which leads to trust ranking. This routing protocol provides a security adapted route from the source to its destination based on the trusted neighbors, and the required security level. The key benefit of this proposed approach is that a route is obtained with a user-defined level of security for a specific application. Thus the tow routing protocols provides a total solution for a secured environment for wireless transmission with security features.

Gurrero Zapata *et al.,* [6] suggested Securing Adhoc Routing Protocols. The problem of integrating security methods into routing protocols for ad hoc networks is considered in this paper. Security solutions like IPSec are not appropriate. A security mechanism for AODV to protect its routing information is considered in this paper. The author also discusses about the application of the proposed approaches to other similar routing protocols. Moreover, how a key management method could be used in combination with the proposed solution is also discussed in this paper.

Davide *et al.,* [7] proposed a securing AODV: the A-SAODV secure routing prototype. Mobile ad hoc networks create new type of security issues, resulted by their characteristics of collaborative and open systems and by restricted accessibility of resources. In this paper, the author considers a Wi-Fi connectivity data link layer as a basis and deals with routing security. The author elaborates the implementation of the secure AODV protocol extension that includes tuning approach which is intended at enhancing its performance. The author provides an adaptive method that enhances SAODV behavior. In addition, the author examines the adaptive strategy and another method which delays the confirmation of digital signatures.

Table 1. Overview of Existing Techniques

Method	Overview
[1]	Withstand Black Hole attack.
[2]	More robust and efficient in eliminating disrupt route discovery.
[3]	Guards against a number of attacks performed against mobile ad-hoc wireless networks.
[4]	Uses public-key cryptographic techniques to counter all the attacks. Very significant in finding secure routes within an ad hoc network.
[5]	Based on distance metrics, the neighbor discovery phase contains the determination of trusted neighbors, which leads to trust ranking. Route is obtained with a user-defined level of security for a specific application.
[6]	Usage of Key Management.
[7]	Some enhancement is performed in SAODV to improve the performance
[11]	ARAN, a routing protocol for ad hoc networks that utilizes authentication and needs the utilization of a trusted certificate server.
[13], [14], [15]	Usage of Hack chains for security.
[19]	Message Authentication Codes used for Security.

The overview of existing secure routing protocol is provided in table 1. These available techniques will helps in understanding the actual problems existing in developing the secure protocol. By analyzing those existing protocols, some techniques like digital signature, hash chains, etc., can be used together to achieve better secure routing protocol.

III. Conclusion

Mobile Ad Hoc Network is a multi-hop wireless network of mobile nodes, structuring a temporary network with no help from several recognized infrastructure or centralized administration. Because of the lack of some committed routers, each node needs to donate towards the configuration and protection of the routing framework. As there are no centrally administered secure routers, attackers can attack the network with ease. To overcome this better routing protocol must be used. AODV is the widely used routing protocol for MANET. But this protocol fails to deliver security benefits. For providing security to MANET, SAODV is used as routing protocol for MANET. This involves the usage of digital signature, hash chains, etc., In this paper, a survey is performed on the existing routing protocols for MANET. Mainly their security support is analyzed which helps for developing better security enabled routing protocol.

References Références Referencias

1. Songbai Lu, Longxuan Li, Kwok-Yan Lam and Lingyan Jia, "SAODV: A MANET Routing Protocol that can Withstand Black Hole Attack", International Conference on Computational Intelligence and Security, Vol. 2, Pp. 421-425, 2009.
2. Papadimitratos, P. and Haas, Z.J., "Secure On-Demand Distance Vector Routing in Ad Hoc Networks", IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication, Pp. 168-171, 2005.
3. Pirzada, A.A. and McDonald, C., "Secure Routing with the AODV Protocol", Asia-Pacific Conference on Communications, Pp. 57-61, 2005.
4. Sanzgiri, K., LaFlamme, D., Dahill, B., Levine, B.N., Shields, C. and Belding-Royer, E.M, "Authenticated Routing for Ad Hoc Networks", IEEE Journal on Selected Areas in Communications, Vol. 23, No. 3, Pp. 598-610, 2005.
5. Khan, R.H., Imtiaz-ud-Din, K.M., Faruq, A.A., Kamal, A.R.M. and Mottalib, A., "A Security Adaptive Protocol Suite: Ranked Neighbor Discovery (RND) and Security Adaptive AODV (SA-AODV)", International Conference on Electrical and Computer Engineering, Pp. 588-593, 2008.
6. Gurrero Zapata, M. and Asokan, N., "Securing Ad hoc Routing Protocols", Proceeding 1st ACM Workshop. Wireless Sec., Pp. 1-10, 2002.
7. Davide Cerri, Alessandro Ghioni, "Securing AODV: The A-SAODV Secure Routing Prototype", IEEE Communication Magazine, Pp. 120-125, 2008.
8. Perlman, R., "Fault-tolerant broadcast of routing information", In Computer Networks, Pp. 395–405, 1983.
9. Kent, S., Lynn, C., Mikkelson, J. and Seo, K., "Secure Border Gateway Protocol (S-BGP)", Real World Performance and Deployment Issues, 2000.
10. Zhou, L. and Haas, Z.J., "Securing Ad Hoc Networks", IEEE Network Magazine, Pp. 24–30, 1999.
11. Dahill, B., Levine, B.N., Royer, E. and Shields, C., "A Secure Routing Protocol for Ad Hoc Networks", Technical Report UM-CS-2001-037, University of Massachusetts, Departament of Computer Science, 2001.
12. Papadimitratos, P. and Haas, Z.J., "Secure Routing for Mobile Ad Hoc Networks", SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), 2002.
13. Hauser, R., Przygienda, A. and Tsudik, G., "Reducing the Cost of Security in Link State Routing", Symposium on Network and Distributed Systems Security, Pp. 93–99, 1997.
14. Cheung, S., "An Efficient Message Authentication Scheme for Link State Routing", 13th Annual Computer Security Applications Conference, Pp. 90–98, 1997.
15. Perrig, A., Szewczyk, R., Wen, V., Culler, D.E, and Tygar, J.D., "SPINS: Security Protocols for Sensor Networks", Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, Pp. 189–199, 2001.
16. Zhang, K., "Efficient Protocols for Signing Routing Messages", Proceedings of the Symposium on Network and Distributed Systems Security, 2001.
17. Hu, Y.C., Johnson, D. and Perrig, A., "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks", 4th IEEE Workshop on Mobile Computing Systems and Applications, Pp. 3–13, 2002.
18. Broch, J., Maltz, D.A, Johnson, D.B., Hu, Y.C. and Jetcheva, J., "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", Proceedings of the 4th Annual International Conference on Mobile Computing and Networking, Pp. 85–97, 1998.
19. Brijesh Kumar, "Integration of Security in Network Routing Protocols", SIGSAC Review, Pp. 18–25, 1993.

This page is intentionally left blank

Global Journal of Computer Science & Technology
Volume 11 Issue 6 Version 1.0 April 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Accurate Corner Detection Methods using Two Step Approach

By Nitin Bhatia , Megha Chhabra
Thapar University

Abstract- : Many image features are proved to be good candidates for recognition. Among them are edges, lines, corners, junctions or interest points in general. Importance of corner detection in digital images is increasing with increasing work in computer vision in imagery. One of the most promising techniques is the one based on Harris corner detection method. This work describes different approaches to detect corner in efficient way. Based on the works carried out by Harris method, the authors have worked upon increasing efficiency using edge detection methods on image, along with applying the Harris on this pre-processed image. Most of the time, such a step is performed as one of the first steps upon which more complicated algorithm rely. Hence, good outcome of such an operation influences the whole vision channel. This paper contains a quantitative comparison of three such modified techniques using Sobel–Harris, Canny-Harris and Laplace-Harris with Harris operator on the basis of distances computed by these methods from user detected corners.

Keywords: Corner Detection, Harris, Laplace, Canny, Sobel.

Classification: GJCST Classification: I.4.6

ACCURATE CORNER DETECTION METHODS USING TWO STEP APPROACH

Strictly as per the compliance and regulations of:

Accurate Corner Detection Methods using Two Step Approach

Nitin Bhatia[α], Megha Chhabra[Ω]

Abstract- Many image features are proved to be good candidates for recognition. Among them are edges, lines, corners, junctions or interest points in general. Importance of corner detection in digital images is increasing with increasing work in computer vision in imagery. One of the most promising techniques is the one based on Harris corner detection method. This work describes different approaches to detect corner in efficient way. Based on the works carried out by Harris method, the authors have worked upon increasing efficiency using edge detection methods on image, along with applying the Harris on this pre-processed image. Most of the time, such a step is performed as one of the first steps upon which more complicated algorithm rely. Hence, good outcome of such an operation influences the whole vision channel. This paper contains a quantitative comparison of three such modified techniques using Sobel–Harris, Canny-Harris and Laplace-Harris with Harris operator on the basis of distances computed by these methods from user detected corners.

Keywords- Corner Detection, Harris, Laplace, Canny, Sobel.

I. Introduction

The concept of interest points connects to the idea of corner detection, where corner features are detected with the primary goal of obtaining robust, stable and well-defined image features for object tracking and recognition of three-dimensional CAD-like objects from two-dimensional images etc. The use of interest points also connects to the notion of regions of interest, which have been used to signal the presence of objects like edges or circles etc. Corner in an image is significantly spotted feature. Conventional corner detection, is one of the common non-destructive testing which employs manual image interpretation. This paper describes a system to detect corners that are present in the given image as a point for which there are two dominant and different edge directions in a local neighbourhood of the point. The system utilizes together two different algorithms, which are distinctively filtered by edge detection method to identify the best object candidate and then Harris as corner detection method to detect the presence of corner in a more efficient way.

The result of the experiment shows a promising output in recognition. In comparison to other recognition features, the edge and corner detection have a great advantage that there is huge variability of the pattern, meaning that large databases can be searched without going into details of stored data about image if appropriate search detection methods are utilised.

Corner detection comes within reach of computer visualization systems to extract certain kinds of characteristics and understand the contents of an image.

Corner detection is frequently used in motion detection, image matching, tracking, image mosaicing, panorama stitching, 3D modelling and object recognition.

Corners in images represent a lot of important information. Extracting corners accurately is significant to image processing, which can reduce much of the calculations. Paper aims at comprehensively using Harris after reading the study made on two widely used corner detection algorithms, SUSAN and Harris corner detection algorithms. Literature proves that both methods based on intensity, when compared in stability, and the runtime of each algorithm, concluded that Harris corner detection algorithm is superior to Susan corner detection algorithm on the whole. The technology is exploited to further sharpening the skills of detecting the corners with the help of edge detection operators. Methods used for edge detection as pre-processing are:

- Sobel Operator
- Canny Operator
- Laplace Operator

Research Literature serves the purpose of improvement desires in context of corner detection .as found in many of research works, a great deal of efforts has been done by computer vision community in solving the problem of efficiently detecting corners and edges. To start with, R. Deriche, R. and Giraudon, G. (1993) present the idea of corner detection by designing a new scale-space based approach by combining properties from the Laplacian and Beaudet's measure for correcting and detecting exactly the corner position. An extension of this approach is then developed to solve the problem of trihedral vertex characterization and detection. Another study is done by Lee, K. & Bie, Z. (1996) in which the gray-level corner detection problem is formulated as a pattern

About [α]- Assistant Professor, Department of Computer Sciecne, DAV College, Jalandhar-144002, Punjab, INDIA.
(Telephone: +91 9872 799 797 email: n_bhatia78@yahoo.com)
About [Ω] - M.Tech. (Student), Computer Science and Applications, SMCA, Thapar University, Patiala, INDIA.
(email: megha.chhbr@gmail.com)

classification problem to determine whether a pixel belongs to the class of corners or not using the concepts of Bayesian classifier and fuzzy logic. Wang, M. J. et al. (1995) presented the idea of corner detection as that the directions of the forward and backward vectors of a non-corner point will cancel each other to detect corners. The bending value is used to assess the degree of possibility of a point being a corner. A paper by Seeger, U. & Seeger, R. (1994) states that only few locally parallel integer operations on 3 × 3 pixel matrices and on six-membered strings of edge elements are required. Within a given direction quantization, local curvature is approximated by finite differences. The extrema of curvature are classified and subsets are selected as corners. Trajkovi, M. & Hedley, M. (1998) present a corner response function (CRF) is computed as a minimum change of intensity over all possible directions. To compute the intensity change in an arbitrary direction an interpixel approximation is used. A multigrid approach is employed to reduce the computational complexity and to improve the quality of the detected corners. Another approach is presented by Freeman, H. & Davis, L. S. (2006), in which a method for detecting sharp corners in a chain-coded plane curve is described. A measure for the prominence of a corner is introduced. Ryu, H. et al. (2007) give a method that proposes that The Hessian matrix has information of ellipse with intensity variance, and corner can be detected by using the eigen-value and eigen-vector analysis and decided weight value. Li, X. et al. (2007) present a method that proposes a hierarchical corner detection framework based on spectral clustering (SC). The framework consists of three stages: contour smoothing, corner cell extraction and corner localization. Alvarez, L. et al. (2001) propose a method using Affine Morphological Scale Space (AMSS) to corner detection with sub-pixel precision.

Many authors have worked on Susan and Harris method of corner detection as well. As presented by Smith, S. M. & Brady, J. M. (1997), concept of each image point having associated with it a local area of similar brightness is the basis for the SUSAN principle. From the size, centroid and second moments of the USAN two dimensional features and edges can be detected. Harris detector is another approach towards the aim.

II. Existing Methods

Corner detection is used as the first step of many vision tasks. Hence, a large number of corner detectors exist in the literature. With so many already available it may appear unnecessary to present yet another detector to the community. However, we have a strong interest in producing a suite of high-speed detectors with the help of combination of edge detection operators. Our work present edge detection operators like Sobel, Canny, Laplace used each with Harris corner detection.

Many diverse interest point detectors have been projected with a wide range of definitions for what points in an image are interesting. Corner points are interesting as they are formed from two or more edges and are generally more abundant in real images than straight edges and are considered to be the most important features. They are striking due to their high information content and hence they are ideal features for tasks such as camera calibration, object tracking or fast interpretation of a robot's environment. Therefore, we present a comparative study of Harris applied with different edge detection methods as pre-processing on input gray-scale images.

The work of this paper can provide a direction to the improvement and the utilization of these corner detection algorithms.

a) *Harris Corner Detection*

Moravec's algorithm is one of the earliest corner detection algorithms. The algorithm tests each pixel in the image to see if a corner is present. It considers a local window in the image, and shifts the window in various directions. This results in changes in image intensity when intensity changes three cases are considered to be important:

- Windowed image pattern is flat as a result no change in intensity takes place in window shifts.
- In case of an edge in image shift along the edge will result in small change whereas perpendicularly shifting the window results in large change.
- If considering a corner or an isolated point ,all shifts will result in large changes

The corner is detected when the smallest SSD (sum of the squared differences) of intensities between the pattern and its neighbours is detected. If this number is locally maximal, then a feature of interest is present.

b) *Sobel Edge Detection Operator*

The Sobel operator performs a 2-D spatial gradient measurement on an image. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image.

Steps:

- The Sobel edge detector uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows).
- A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time.

- If we define A as the source image, and G_x and G_y are two images which at each point contain the horizontal and vertical derivative approximations, then the masks are as follows:

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

- The magnitude of the gradient is then calculated using the formula:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

- An approximate magnitude can be calculated using:

$$|G| = |G_x| + |G_y|$$

c) *Canny Edge Detection Operator*

Canny edge detector discovers the optimal edges. In this situation, an "optimal" edge detector means it should mark all possible edges .marked edges are visited only once and possibly are the only edges not any false data.

Steps:

- The Canny edge detector uses a filter based on the first derivative of a Gaussian.
- It is prone to noise present on raw unprocessed image data, so to begin with, the raw image is convolved with a Gaussian filter.
- Result is a slightly blurred version of the original which is not affected by a single noisy pixel to any significant degree. Smaller filters cause less blurring, and allow detection of small, sharp lines.
- An edge in an image may point in a variety of directions, so the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image.
- The edge detection operator (Sobel for example) returns a value for the first derivative in the horizontal direction (Gy) and the vertical direction (Gx). From this the edge gradient and direction can be determined:

d) *Laplacian of Gaussian*

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image spots the regions of rapid intensity change and is therefore often used for edge detection.

III. Proposed Work and Experimental Analysis

Corner detection puts effort on time and accuracy calculation. Proposed work accepts the fact and uses it as the basis of enhancement in corner detection by Harris method. In this method, edge detection is performed over the image sequence before corner detection can be preformed. Instead of using data of whole of the image, edges are detected and that data is only required out of the whole image for detecting corners.

Processing steps for achieving results:

- EDGE DETECTION:

Sobel edge detection, canny edge detector and laplacian operator are applied to detect edges in an image.

- CORNER DETECTION:

Harris is then utilised on various images to detect corners from this reduced amount of data. To make the process more clear, images are of human faces with various expressions.

Experimental analysis: experiments have been performed on image data set of some common human face expressions [11].

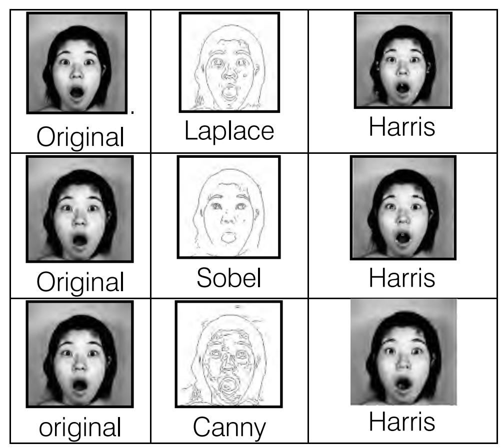


Fig.1. Comparison between outputs of images applied with various edge detectors and then Harris.

In Experiments, user selects a corner point on the image. After this, distances of corners detected by Sobel-Harris, Canny-Harris, and Laplace-Harris is compared with distance calculated by Harris from the point detected by user. This affirms the accuracy of these proposed three ways of fast detection of corners in terms of fact that these methods give better results than Harris. The results are shown graphically as well as tables demonstrating all the fields used to determine the fast methodology of proposed methods than original.

Table 1 show various feature points in a facial image along with the respective co-ordinate values of corners as selected manually. The results of Harris corner detector, various proposed detectors are also given in the same table. Each of the corner detectors have the Euclidean distance computed from the actual corner point.

Global Journal of Computer Science and Technology Volume XI Issue VI Version I April 2011

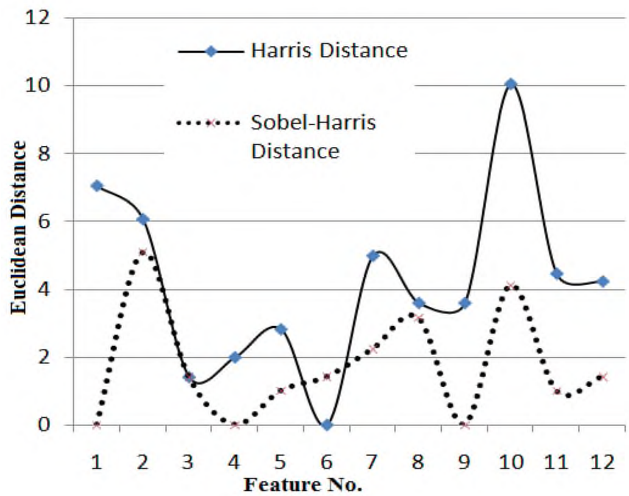


Fig. 2. *Harris vs. Sobel-Harris Corner Detector*

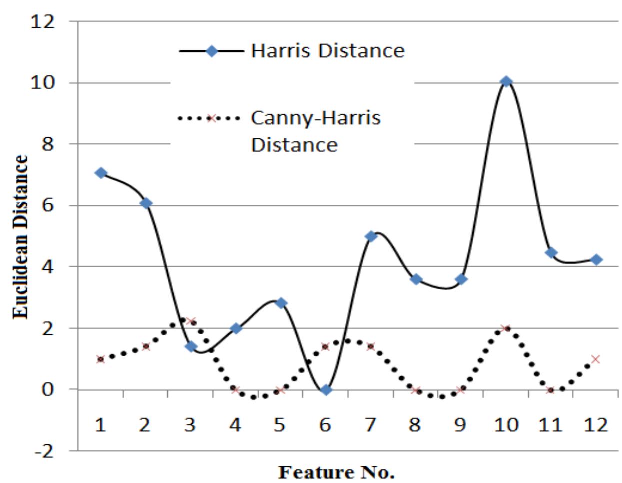


Fig. 3. *Harris vs. Canny-Harris Corner Detector*

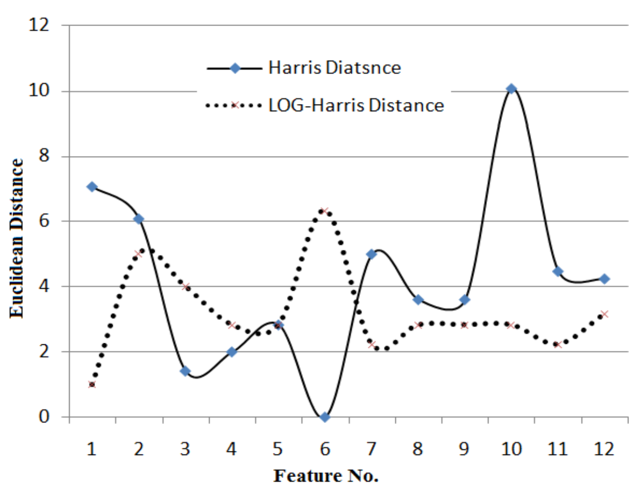


Fig 4. *Harris vs. LoG-Harris Corner Detector*

Figure 2 shows a graphical comparison of Harris method and Sobel-Harris method. In this case proposed Sobel-Harris detector outperforms Harris detector in case of all features except feature number 6, i.e. extreme right corner of left eyebrow. Similarly, figure 3 presents a comparison of Harris method and Canny-Harris method. In this case proposed Canny-Harris detector outperforms Harris detector in case of all features except feature numbers 3 and 6, i.e. extreme left of right eyeball and extreme right corner of left eyebrow. In fig 4, we present comparison of Harris method and LoG-Harris method. In this case proposed Log-Harris detector outperforms Harris detector in case of all features except feature numbers 3, 4 and 6, i.e. both extremes of right eyeball and extreme right corner of left eyebrow. But it is interesting to note that wherever the proposed methods are outperforming the well established Harris method, the accuracy is achieved with a great margin, for example, feature numbers 1 and 10, i.e. extreme left of left eyeball and right nostril.

IV. Conclusion

Experiments show that enhancement in Harris detection of corners in terms of pre-processing and hence reducing data has increased accuracy in terms of less distance detection from user detected corner points. This is proved that Laplace-Harris operator is best suitable enhancement in such a case. This determines even those corners and with good amount of precision, which Harris or other enhancements are not able to discover. Paper presents a stable and accurate corner detection algorithm, which is simple and an efficient means of producing input points of interest for feature-based approaches.

References Références Referencias

1. Alvarez, L., Cuenca C. & Mazorra, L. (2001). Morphological Corner Detection. Application to Camera Calibration. IASTED International Conference on Signal Processing, Pattern Recognition & Applications.
2. Li, X., Hu, W. & Zhang, Z. (2007). Corner Detection of Contour Images using Spectral Clustering. IEEE International Conference on Image Processing, 2007. ICIP 2007.
3. Lee, K. & Bien, Z. (1996).A gray-level corner detector using fuzzy logic. Pattern Recognition Letters, Volume 17, Issue 9, 1 August 1996, Pages 939-950.
4. Seeger, U. & Seeger, R. (1994). Fast corner detection in grey-level images. Pattern Recognition Letters, Volume 15, Issue 7, July 1994, Pages 669-675.
5. Smith, S. M. & Brady, J. M. (2001). SUSAN—A New Approach to Low Level Image Processing. International Journal of Computer Vision, Volume 23, Number 1, 45-78, DOI: 10.1023/A:1007963824710.
6. Wang, M. J., Wu, W. Y., Huang, L. K. & Wang, D. M. (1995). Corner detection using bending

value. Pattern Recognition Letters, Volume 16, Issue 6, June 1995, Pages 575-583.
7. Deriche, R. & Giraudon, G. (1993). A computational approach for corner and vertex detection. International Journal of Computer Vision archive, Volume 10 Issue 2, April 1993.
8. Trajkovi, M. & Hedley, M. (1998). Fast corner detection. Image and Vision Computing, Volume 16, Issue 2, 20 February 1998, Pages 75-87.
9. Freeman, H. & Davis, L.S. (2006). A Corner-Finding Algorithm for Chain-Coded Curves. IEEE Transactions on Computers, DOI: 10.1109/TC.1977.1674825.
10. Ryu, H., Lee, J., Hwang, E., Jing, L., Lee, H. & Choi, W. (2007). A new corner detection method of gray-level image using Hessian matrix. International Forum on Strategic Technology, 2007. IFOST 2007. DOI: 10.1109/IFOST.2007.4798654.
11. Lyons, M.J., Akamatsu, S., Kamachi, M., Gyoba, J., (1998). Coding Facial Expressions with Gabor Wavelets. Third IEEE International Conference on Automatic Face and Gesture Recognition, April 14-16 1998, Nara Japan, IEEE Computer Society, pp. 200-205.

Table 1: *Comparison of Proposed Methods with Existing Harris Corner Detection Method*

Feature Points	Manually Selected Corner Points		Existing Method Harris			Proposed Methods Sobel-Harris			Canny-Harris			L.O.G-Harris		
	X	Y	X	Y	Dist	X	Y	Dist	X	Y	dist	X	Y	Dist
Left eyeball extreme left	86	130	87	123	7.0711	86	130	0	86	131	1	86	131	1
Left eyeball extreme right	102	128	108	129	6.0828	107	127	5.09902	101	129	1.41421	107	128	5
Right eyeball extreme left	148	127	147	128	1.4142	147	128	1.41421	147	129	2.23607	148	131	4
Right eyeball extreme right	170	128	170	130	2	170	128	0	170	128	0	172	130	2.82843
Left eyebrow extreme left	70	114	72	112	2.8284	71	114	1	70	114	0	72	116	2.82843
Left eyebrow extreme right	111	109	111	109	0	112	108	1.41421	110	108	1.41421	117	107	6.32456
Right eyebrow extreme left	140	109	145	109	5	142	108	2.23607	141	110	1.41421	142	110	2.23607
Right eyebrow extreme right	180	112	182	109	3.6056	181	109	3.16228	180	112	0	182	114	2.82843
Left nostrils	111	170	113	167	3.6056	111	170	0	111	170	0	113	168	2.82843
Right nostrils	140	170	130	171	10.05	136	169	4.12311	138	170	2	142	168	2.82843
Left mouth corner	102	191	100	195	4.4721	103	191	1	102	191	0	104	190	2.23607
Right mouth corner	150	191	147	194	4.2426	151	190	1.41421	151	191	1	149	188	3.16228

Global Journal of Computer Science & Technology
Volume 11 Issue 6 Version 1.0 April 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Redundancy Effect on Fault Tolerance in Wireless Sensor Networks

By A. Mojoodi , M. Mehrani , F. Forootan , R.Farshidi
Islamic Azad University

Abstract- : There are several usages for wireless sensor networks in military, supervisory, region considering, physiology, etc. Sensor nodes are usually equipped with some non chargeable batteries having limitations in lifetime while they cannot be replaced with other sensors when they fail. Each sensor has a failure probability which is affected from some factors like electrical energy, hardware failures, communication error, undesired environment situations, etc. Thus, fault tolerant is a very important and critical factor in such networks. Hardware redundancies like having redundant nodes and paths are suitable echniques used for increasing fault tolerant factor. On the other hand, using hardware redundancies leads increasing in overall network consumption because of using more number of sensors. In this paper we estimate the affect of redundancy on the number of correct responses that wireless sensor networks have on the received queries and also show the level of redundancy needed in different network conditions like having different fault probabilities and needed clusters to response the received queries. Keywords: wireless sensor network, fault tolerance, redundancy, reliability

Keywords: Knowledge Penetration Process, Knowledge Discovery in Databases, Splitted KDD, Parameter Dependent and Parameter Independent Processes, Genetic Algorithms.

Classification: GJCST Classification: H.3.3, B.8.1

REDUNDANCY EFFECT ON FAULT TOLERANCE IN WIRELESS SENSOR NETWORKS

Strictly as per the compliance and regulations of:

Redundancy Effect on Fault Tolerance in Wireless Sensor Networks

A. Mojoodi[α], M. Mehrani[Ω], F. Forootan[β], R.Farshidi[Ψ]

Abstract- there are several usages for wireless sensor networks in military, supervisory, region considering, physiology, etc. Sensor nodes are usually equipped with some non chargeable batteries having limitations in lifetime while they cannot be replaced with other sensors when they fail. Each sensor has a failure probability which is affected from some factors like electrical energy, hardware failures, communication error, undesired environment situations, etc. Thus, fault tolerant is a very important and critical factor in such networks. Hardware redundancies like having redundant nodes and paths are suitable echniques used for increasing fault tolerant factor. On the other hand, using hardware redundancies leads increasing in overall network consumption because of using more number of sensors. In this paper we estimate the affect of redundancy on the number of correct responses that wireless sensor networks have on the received queries and also show the level of redundancy needed in different network conditions like having different fault probabilities and needed clusters to response the received queries. Keywords: wireless sensor network, fault tolerance, redundancy, reliability

I. Introduction

Recent advancements in integrated circuits have fostered the emergence of a new generation of tiny, inexpensive and low power sensors. Due to their economic and computational feasibility, a network of hundreds and thousands of sensors has the potential for numerous applications in both military and civil applications such as combat field surveillance, security and disaster management. These sensing devices are capable to monitor a wide variety of ambient conditions such as: temperature, pressure, motion etc. The sheer number of these devices and their ad-hoc deployment in the area of interest brings numerous challenges in networking and management of these systems.

Sensors are typically disposable and expected to last until their energy drains. Therefore, energy is a very scarce resource for such sensor systems and has to be managed wisely in order to extend the life of the sensors for the duration of a particular mission. Sensors send the received data to a base station. This can be done periodically or according to an event. The base station is located faraway from the area where the sensors are usually deployed. In order to conserve energy consumed in communication with the command node various multi-hop and energy aware routing techniques have been suggested in the literature [5][6]. Considering the application of sensor networks, fault tolerance is very important for them. This importance can be observed in some applications such as military environments and nuclear experiment obviously.

Fault tolerant is a characteristic of the network which leads the network to continue its functionality when an error occurs by covering the affect of occurred error, such that we can make sure about the service presented by the network. Since in some sensor network applications receiving correct data by base station is a very important factor and decision making based on non correct data leads huge damages, fault tolerant in sensor networks has an extreme importance. Fault tolerant is the ability of maintaining the proper usefulness of the sensor network without having any lacks made by network problems. [4]

Using hardware redundancies like applying more sensor nods and more data transmitting paths for recognizing and covering occurred faults are some techniques used to achieve at a fault tolerant wireless sensor network. [5-8]

Using hardware redundancy increases overall network consumption because of applying more number of sensor nodes to send and receive data packets, but in other hand, it increases network fault tolerant and reliability.

By paying attention to mentioned notes, implementing hardware redundancy can be useful against sensors faults and transmitting errors but it should be used just in critical situations to avoid network overall energy consumption.

II. Faults and Hardware Redundancies

The occurred faults in wireless sensor networks have two reasons. First, since the sensor networks are dispersed in a region such that replacing the sensors is very hard or even impossible, losing energy by sensor nodes is one of the main reasons of faults. The network consumes its energy to do its duty.

About [α, β, Ψ] - Islamic Azad University, Dezful Branch, Dezful Iran
E-mail- mojoodi@iaud.ac.ir, forootan@iaud.ac.ir, farshidi@iaud.ac.ir
About [Ω]-Islamic Azad University, Shoushtar Branch, Shoushtar, Iran
E-mail- m.mehrani@iau-shoushtar.ac.ir

The second main reason is the sensor problems like hardware lacks, software lacks or data transmitting lacks made by noises or having interferes in the network. Also, having changes in binary data packets during processing leads data inconsistency.

There are some different hardware redundancies to avoid the mentioned possible faults. Using the following strategies increases fault tolerant factor for WSN.

The first strategy is path redundancy. This technique instead of using a single path to connect a source cluster to the processing center, utilizes m disjoint paths between them. Node redundancy is the second strategy. In this strategy, instead of having one sensor node in charge of returning requested data, ms sensor nodes within a source cluster are used. This is done to cope with sensor or transmission faults.

In the next part we estimate using of the mentioned hardware redundancies in a given sensor network and try to calculate the fault tolerance degree, network energy consumption and the number of correct reactions to the number of queries.

III. HOW TO APPLY REDUNDANCY IN WSN

Both of the average energy consumption of each request (Eq) and its average safety factor (Rq) are dependent on the redundancy level presented by the network to reply a request. The magnitude of Eq and Rq largely depends on the redundancy level used by the system to reply the incoming request.

Without any redundancy, both E_q and R_q are decreased. But using excess redundancy leads the same increment in both E_q and R_q. So, a trade off is required between these two parameters.

Like [1] we consider a wireless sensor network with a certain number of nodes dispersed in a non accessible and dense region. All the sensor nodes have same initial energy. Also a clustering algorithm like LEACH [10] or HEED [9] is used for partitioning the nodes in different groups. Obviously the clustering method denotes a cluster head for each cluster to manage the cluster activities during each round of network functioning.

Usually, the network nodes are equipped by low power batteries without chance if recharging, so they use multi hop data transmission to save energy.

Let q_t to be the failure rate of data transmission caused by noise or interfere while q_s represents the sensor fault.

Assume that the user has a query. Therefore, the processing center is called to send this request to cluster head. Each user query may involve one cluster, some of them or even all the network clusters in responding the issued query. When a query calls for a cluster, one sensor is sufficient to answer the query. It can transmit its collected information from environment to processing center through a path.

To achieve at fault tolerance, the chosen sensor nodes for receiving region information can send information to data processing center via m separate paths, instead of sending them just to the cluster head. Based on this data transmission technique, considering the probability of data transmission faults and also by paying attention to probability of sensor faults we can make sure about calculating the reliability and energy consumption of the network.

IV. RELIABILITY AND ENERGY CONSUMPTION

In this section we calculate the network energy consumption and network reliability factors for the network which is equipped by some redundant ensors to achieve at hardware redundancy.

Assume d as a random variable which determines distance between a sensor and processing center. Thus, the number of hops between the processing center and the source sensor, denoted by h, is given by: h=d/r-1 [1]. Like [1] let the source cluster head be randomly located at (Xi, Yi) in the square shape sensor area when -A/2 ≤ Xi ≤ A/2 and- A/2 ≤ Yi ≤ A/2 and the processing center is located at the center of the network area and its coordinates is (0, 0). Then, E[h] which is the expected value of *h* is given by:

$$E[h]=\int_{-A/2}^{A/2}\int_{-A/2}^{A/2}\left(\frac{\sqrt{(X_i^2+Y_i^2)}}{r}-1\right)\left(\frac{1}{A}\right)\left(\frac{1}{A}\right)dX_i\,dY_i=\frac{0.3825A}{r}-1$$

(1)

For notational convenience, let .Here N_h represents the average number of hops (or sensors) to forward sensor data from a source sensor to the processing center.

A sensor will fail to return its reading to the cluster head when any hop fails, so the failure probability of that source cluster in delivering data to the processing center due to sensors fault is given by:

$$P_s=1-(1-q_s)^{N_h+1} \qquad (2)$$

Also, the failure probability of that source cluster in delivering data to the processing center due to transmissions fault is given by:

$$P_t=1-(1-q_t)^{N_h} \qquad (3)$$

Now, suppose there is just one simple path from the source sensor to the processing center and because of sensor and transmission failures the process of sending data from the source cluster to the processing center may encounter with a failure. Therefore, the probability of the mentioned possible failure is calculated as following:

$$P_f = 1-(1-p_t)\,(1-p_s) \qquad (4)$$

If a sensor uses m separated paths to return its reading hen the failure probability is given by:

$$P_{fm} = (p_f)^m \qquad (5)$$

Suppose that the application demands k source clusters to return sensor data to reply a query, then the failure probability is given by:

$$P_{fkm} = 1-(1-p_{fm})^k \qquad (6)$$

Therefore, the reliability of a query that requires k clusters to respond is given by:

$$R_{qm}(k) = 1 - P_{fkm} \qquad (7)$$

Now energy consumption in this process is to be estimated and calculated as following. The energy used for communication is denoted as Eelec per bit. Thus, the energy spent by a sensor node to sense (or to receive) and transmit a data packet of length nb bits is given by:

$$E_{packet} = 2 * n_b E_{elec} \qquad (8)$$

m redundant paths transmit received data from a sensor to the data processing center, separately. In this situation, if a request needs k clusters and each one uses m redundant paths to reply the request then the overall needed energy for the sensors placed in these *K* clusters to send data to the processing center can be calculated as $E_{packet} * N_h * k * m$. Thus, the amount of energy spent by the system, Eq(k), to answer a query that demands k clusters to respond is given by:

$$Eq(k) = E_{packet} * N_h * k * m \qquad (9)$$

The average number of queries that the system is able to sustain before running out its energy is given by:

$$N_q = \frac{E_{initial} - E_{threshold}}{Eq} \qquad (10)$$

Let the average reliability of a query be Rq , given as:

$$R_q = \sum_{k=1}^{np} R_q(k) P_q(k) \qquad (11)$$

If the system is able to reply Nq requests with reliability factor Rq then the number of the requests that can be replied correctly before system loses its energy can be used as a scale to measure the network life time.

V. Numeric Results

Assume a wireless sensor network characteristics: n = 1000 nodes, r = 1, J = 10 nodes/square unit, A = 10 units, nb = 50 bytes, Eelec = E0 = 2J, Ethreshold = 0. We want to count the number of requests that network can reply correctly in different conditions.

In the first phase with assumption that network needs just one cluster to reply each request and has different fault probabilities while there isn't any redundancy in the network we calculate the number of correct replies. Then we use two, three and four redundant paths to send data and calculate the number of correct replies in each mentioned situations. The results are shown in figure 1.

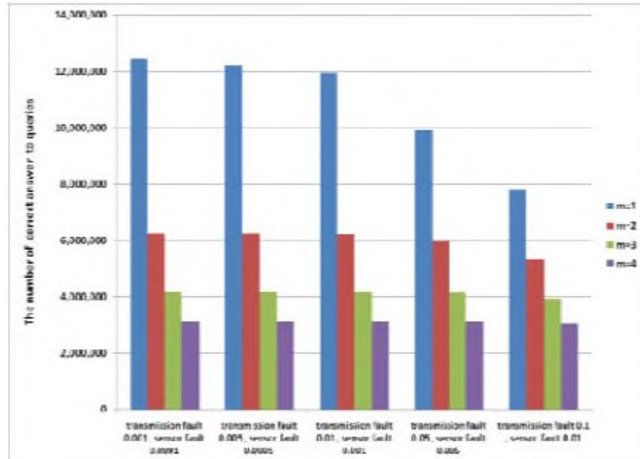


Figure 1. The number of correct answers when network requires one cluster to answer

In the next phases we assume the data transmission fault rate between 0.001 and 0.1, also we suppose the sensor faults rate between 0.0001 and 0.001 (less than data transmission fault rate [1]), then we calculate the number of correct replies of the system.

The number of needed clusters to reply the requests is chosen from k=1 to k=50 and then the calculations for the situations that the number of redundant paths are zero, two, three or four will be repeated. The results are shown in figure 2 to figure 6.

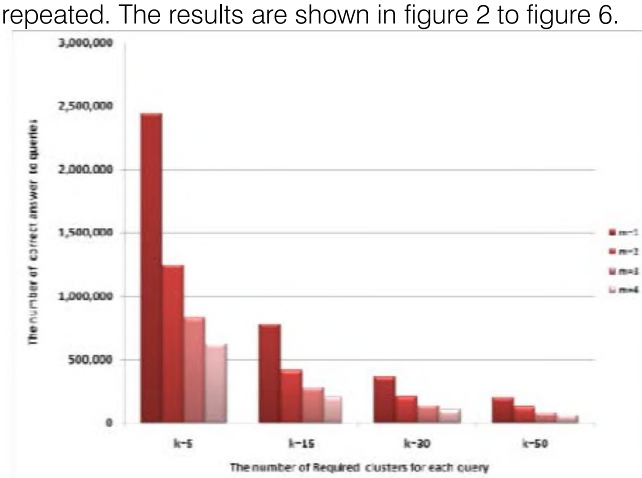


Figure 2. The number of correct answers when transmission fault is 0.001 and sensor fault is 0.0001

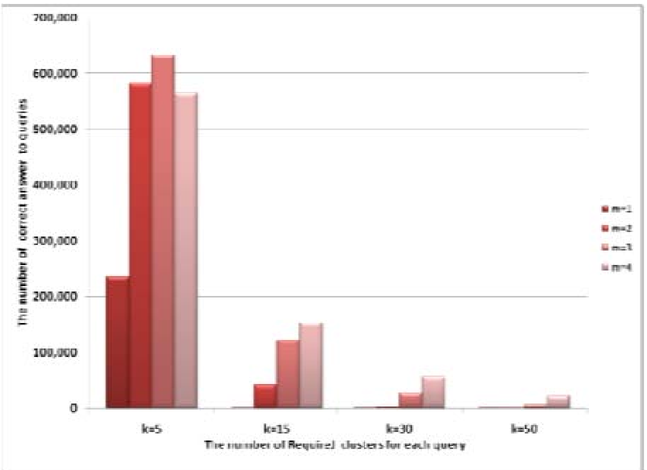


Figure 3. The number of correct answers when transmission fault is 0.005 and sensor fault is 0.0005

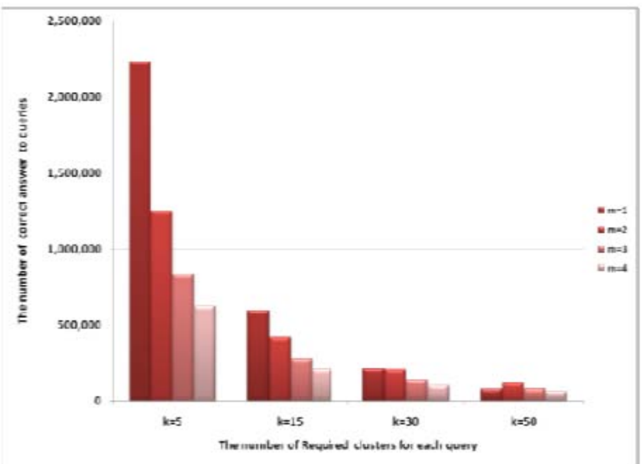


Figure 4. The number of correct answers when transmission fault is 0.01 and sensor fault is 0.001

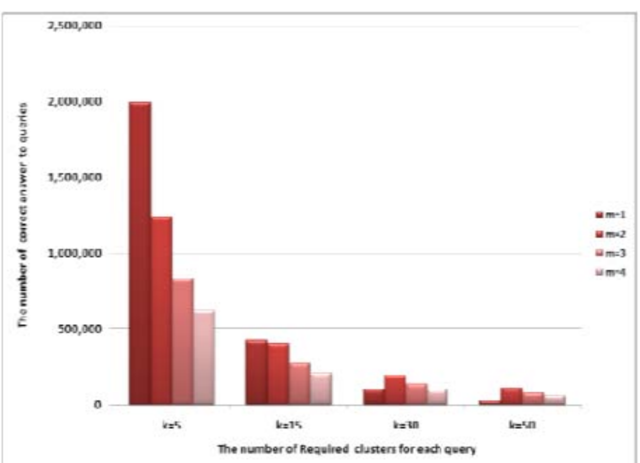


Figure 5. The number of correct answers when transmission fault is 0.05 and sensor fault is 0.005

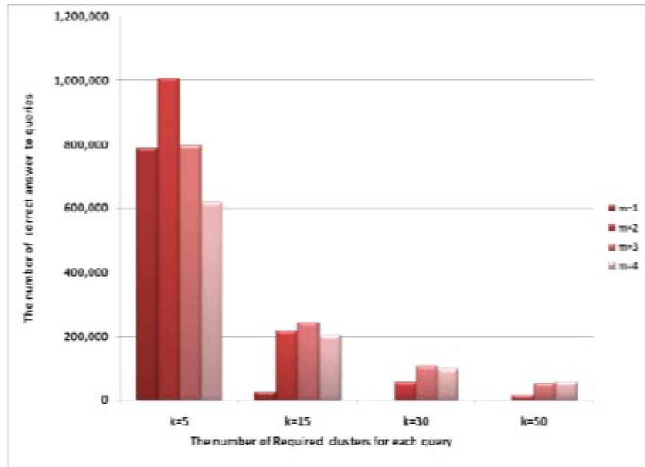


Figure 6. The number of correct answers when transmission fault is 0.1 and sensor fault is 0.01

VI. Results

According to the charts presented in the previous section the following results have been seen.

1. If the average number of clusters needed to response a request is equal to one then the use of redundancy at any level despiting the high probability of transmission and sensor errors, reduces the number of correct responses to the request in the network lifetime. The reason is clear. Due to the use of only one cluster, the reliability of each request will not greatly reduce, however if the redundant path is used the energy of sensors in the redundant paths are wasted and based on the charts if the redundant path is not to be used then the average network lifetime will be longer.

2. When the probability of transmission errors and sensor errors is negligible, the use of redundancy reduces the number of correct responses during network lifetime. Because the error probability is low, so there is high reliability. Therefore, the use of redundancy increases energy consumption and reduces the number of correct responses in network lifetime.

3. Whatever the number of used clusters for responding the request increases and also whatever the possibility of errors raises, then the affect of redundancy is going to be more visible. In this case, using the different levels of redundancy will have different affects such that whatever the number of used paths increases the number of correct responses during network lifetime will increases until using more redundancy will have reverse affect. This means that the use of excessive number of redundant paths reduces the number of correct responses because it causes network energy wasting in excessive paths.

So in this paper, by estimating the affect of redundancy on the number of correct responses that an on- demand network can have during its lifetime and before losing its total energy, we found out that the effect of redundancy on different scenarios of the network is different.

When the number of clusters needed to response to the request is low or little chance of error exists in the network, using the redundancy decreases the number of correct responses. Against, when the number of clusters is high or there is a big chance of error then more paths are needed to maximize the reactions. In other hand, these redundant paths have an optimum number in evernetwork situation such that using more paths reduces the correct reactions.

References Références Referencias

1. Anh Phan Speer and Ing-Ray Chen , "Effect of Redundancy on the Mean Time to Failure of Wireless Sensor Networks" , Concurrency Computat. 2007. Pract. Exper. 19:1119- 1128.
2. S. Singh, M. Woo and C. S. Raghavendra, "PowerAwareRouting in Mobile Ad Hoc Networks", Proc. of ACM MOBICOM', Dallas, Texas, October 2004
3. D. Estrin, R. Govindan, J. Heidemann, and S. umar. "Scalable coordination in sensor networks". Proc. of ACM/IEEE MobiCom , Seattle, Washington, August 2005.
4. Koushanfar F., Potkonjak M., and Sangiovanni A., "Fault tolerance Techniques for sensor networks", in: Proceedings of the IEEE Sensors Mag, Vol. 2, pp. 1491–1496, 2002.
5. Ding M., Chen D., Thaeler A. and Cheng X., "Fault-Tolerant Target Detection in Sensor Networks", in: Proceedings of the IEEE WCNC 2005, Vol. 4, pp. 2362 2362-2368, 13-17 March 2005.
6. Ding M., Chen D., Xing K. and Cheng X., "Localized Fault- Tolerant Event Boundary Detection in Sensor Networks", in: Proceedings of the IEEE INFOCOM 2005, Miami, March 2005.
7. Luo X., Dong M. and Huang Y., "On Distributed Fault Y., "On Distributed Fault-Tolerant Detection in Wireless Sensor Networks", in: Proceedings of the IEEE Transactions on Computers, Vol. 55, no. 1, pp. 58 ters, Vol. 55, no. 1, pp. 58-70, January 2006.
8. Huang X. M., Deng J. and Ma J., "Fault Tolerant Routing For Wireless Sensor Grid Networks", in: Proceedings of the IEEE s", in: Proceedings of the IEEE Sensors Applications Symposium, Houston, TX, February 2006 m, Houston, TX, February 2006
9. O. Younis and S. Fahmy, "HEED: A Hybrid Energy Efficient, Distributed Clustering Approach for Ad Hoc Sensor Network", IEEE Transaction on Mobile Computing, Vol. 3, No. 3, October- December 2004,pp. 366-379.
10. W. Heinzelman, C. Chandrakasan and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Transactions on Wireless Communication, Vol.1, No. 4, 2002, pp. 660 02, pp. 660-670.

Global Journal of Computer Science and Technology Volume XI Issue VI Version I April 2011

This page is intentionally left blank

Global Journal of Computer Science & Technology
Volume 11 Issue 6 Version 1.0 April 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Fuzzy Cluster Means Expert System for the Diagnosis of Tuberculosis

By Imianvan A.A. , Obi J.C.
University of Benin

Abstract- : -Tuberculosis (TB) is a global public health problem of enormous dimension. Tuberculosis is usually associated with mycobacterium tuberculosis (the bacterium causing tuberculosis). TB is an infectious disease, transmitted and spread via aerosols (droplets from the mouth and respiratory tract) that are coughed, sneezed, or forcibly expelled from the body to the surrounding air. These droplets, when inhaled by a susceptible host, can infect a new person and, within weeks to months, the disease begins to develop within the infected person. The lungs are the primary site of infection. The disease can spread to almost any other organ such as: kidneys, bladder, bones, spine, liver, spleen and brain. TB symptoms are characterized by low grade fever, coughing, fatigue, and a loss of appetite. Later, hemoptysis (coughing up blood), may occur. The application of Fuzzy Cluster Means (FCM or Fuzzy C-Mean) analysis to the identification of different types of tuberculosis is the focal point of this paper. Application of cluster analysis involves a sequence of methodological and analytical decision steps that enhances the quality and meaning of the clusters produced. The uncertainties often associated with analysis of tuberculosis test data are eliminated by the proposed system.

Keywords: Fuzzy logic, Fuzzy Cluster Means, Tuberculosis (TB).

Classification: GJCST Classification: FOR Code: 080299,080204,080108

FUZZY CLUSTER MEANS EXPERT SYSTEM FOR THE DIAGNOSIS OF TUBERCULOSIS

Strictly as per the compliance and regulations of:

GLOBAL ASSOCIATION OF RESEARCH

Fuzzy Cluster Means Expert System for the Diagnosis of Tuberculosis

Imianvan A.A. , Obi J.C.

Abstract-Tuberculosis (TB) is a global public health problem of enormous dimension. Tuberculosis is usually associated with mycobacterium tuberculosis (the bacterium causing tuberculosis). TB is an infectious disease, transmitted and spread via aerosols (droplets from the mouth and respiratory tract) that are coughed, sneezed, or forcibly expelled from the body to the surrounding air. These droplets, when inhaled by a susceptible host, can infect a new person and, within weeks to months, the disease begins to develop within the infected person. The lungs are the primary site of infection. The disease can spread to almost any other organ such as: kidneys, bladder, bones, spine, liver, spleen and brain. TB symptoms are characterized by low grade fever, coughing, fatigue, and a loss of appetite. Later, hemoptysis (coughing up blood), may occur. The application of Fuzzy Cluster Means (FCM or Fuzzy C-Mean) analysis to the identification of different types of tuberculosis is the focal point of this paper. Application of cluster analysis involves a sequence of methodological and analytical decision steps that enhances the quality and meaning of the clusters produced. The uncertainties often associated with analysis of tuberculosis test data are eliminated by the proposed system.

Keywords: Fuzzy logic, Fuzzy Cluster Means, Tuberculosis (TB).

I. Introduction

Tuberculosis (TB) is a global health problem. It is estimated that about one billion individuals are infected world wide with tuberculosis, with 10 million new cases and over 3 million deaths per year (Taura et al., 2008). Tuberculosis is amongst the world's leading cause of death from a single infectious disease.

TB most commonly affects the lungs but also can involve any organ of the body. Tuberculosis has been known under a variety of names during the course of history. It has been a difficult disease to diagnose and has been confused with many other diseases. The actual name "Tuberculosis" was introduced during the first half of the nineteenth century. It refers to the diseased condition caused by infectious agents known as mycobacterium tuberculosis or tubercle bacilli (Neil and Janet, 2005). The disease has also been known under other names, such as phthisis, Scrofula, tabes, bronchitis, and inflammation of the lungs, hectic fever, gastric fever, and lupus (Neil and Janet, 2005). It was also known as the great white plague or "consumption" (MedicineNet, 2005).

About - Department of Computer Science, University of Benin, Benin City. Nigeria.
E-mail: tonyvanni@yahoo.com

It's a rod-shaped bacterium 2 to 4 micrometers in length (CWS, 2000). Although tuberculosis can attack the whole body it is frequently known more for its damage to the lungs (Neil and Janet, 2005). TB was first isolated in 1882 by a German physician named Robert Koch who received the Nobel Prize for this discovery. TB is usually transmitted and spread via aerosols (droplets from the mouth and respiratory tract) that are coughed, sneezed, or forcibly expelled from the body to the surrounding air. These droplets, when inhaled by a susceptible host, can infect a new person and, within weeks to months, the disease begins to develop in that infected person. Apart from the lungs, the disease can spread to almost any other organ such as: kidneys, bladder, bones, spine, liver, spleen and brain. TB infections are characterized by low grade fever, coughing, fatigue, and a loss of appetite. Later, coughing with hemoptysis (blood in the sputum), may occur. If the infection in the lung worsens, then further symptoms can include chest pain, and shortness of breath. If the infection spreads beyond the lungs, the symptoms will depend upon the organs involved (MedicineNet, 2011).

Fuzzy logic provides a means for representing and manipulating data that are not precise, but rather fuzzy. Fuzzy logic presents an inference morphology that enables appropriate human reasoning capabilities to be applied to knowledge-based systems. The theory of fuzzy logic encompasses a mathematical strength to capture the uncertainties associated with human cognitive processes. This paper presents a Fuzzy Cluster Mean (Fuzzy C-Mean or FCM) knowledge-based model for the diagnosis of TB. Fuzzy C-means clustering algorithm is being used as the problem solving and reasoning algorithm in the inference engine of the knowledge base system for the evaluation, classification and matching of patterns to more than one class of tuberculosis. Statistics, Neural network and Physiology are also incorporated. The main objective is to classify and match any individual independently with more than one cluster depending on the degree of membership.

II. Literature Review

There are many form of TB, but for the purpose of this paper we shall consider seven of them (Taura et al., 2008). These are

i. Tuberculosis Gastrointestinal

ii. Tuberculosis Meningitis
iii. Tuberculosis Lymphadenitis (Scrofula)
iv. Cutaneous Tuberculosis (Lupus vulgaris; Tuberculosis verrucosa; Milary
v. tuberculosis)
vi. Osteo-articular Tuberculosis
vii. Genitourinary Tuberculosis
viii. Drug-resistant Tuberculosis (Multi-drug resistant tuberculosis (MDR-TB) and
ix. Extensively drug-resistant tuberculosis (XDR-TB))

III. Tuberculosis Meningitis

Tuberculosis meningitis is a TB infection of the brain and the spinal cord. The initial symptoms can be irritability and restlessness. Later the patient may develop other symptoms such as a stiff neck, headaches, vomiting, and variations in mental behavior, seizures, or coma.

IV. Gastrointestinal Tuberculosis

Gastrointestinal tuberculosis is TB of the gastrointestinal tract: mouth, oesophagus, stomach, small and large intestine, and the anus. The symptoms are abdominal pain, fever, weight loss, nausea, vomiting, and change in bowel habits.

V. Tuberculosis Lymphadenitis (Scrofula)

Tuberculosis lymphadenitis is TB of the lymph nodes, usually along the neck. The symptoms are the formation of masses along the neck, and if the disease is advanced the mass may burst and form a draining sinus.

VI. Cutaneous Tuberculosis

Cutaneous tuberculosis is TB of the skin or mucous membrane from an external source of mycobacteria. There are several types of cutaneous tuberculosis: Lupus vulgaris, tuberculosis verrucosa cutis and milary tuberculosis.

- Lupus vulgaris is a persistent type of cutaneous TB. The symptoms are small reddish brown lesions that are found on the face, eyelids, around the nose, cheeks, and ears.
- Tuberculosis verrucosa cutis is only contracted through direct skin inoculation when an individual had been previously exposed to mycobacteria. This type of cutaneous TB can last for years. The symptoms are reddish brown wart-like growths on the body, skin lesions on hands, feet, buttocks, elbows and knees. Sometimes pus will seep through the fissures present in the lesions
- Milary tuberculosis is a cutaneous TB that starts off as a pulmonary TB infection which then travels through the bloodstream. The symptoms are small red spots on the skin (which are sometimes concentrated to the trunk of the body), necrosis of infected areas, and the development of ulcers or abscesses on the skin.

VII. Osteo-Articular Tuberculosis

Osteo-articular tuberculosis is TB of the joints: knees, hips, ankles, wrists, shoulders, and elbows. It usually affects one joint. The symptoms are similar to those experienced by individuals with arthritis and pain or stiffness is only felt in the infected area.

VIII. Genitourinary Tuberculosis

Genitourinary tuberculosis is TB that initially begins as a pulmonary (lungs) TB infection which then travels through the bloodstream to the genitourinary tract. The genitourinary tract includes the urinary tract and the reproductive system. The symptoms are blood present in urine, painful or uncomfortable urination, and experiencing pain on one side of the body between the upper abdomen and back.

IX. Drug-Resistant Tuberculosis

Drug-resistant tuberculosis is a TB infection that does not respond to drugs used for treatment of TB infection. This type of TB occurs due to the poor management of TB care or the individual was infected by bacteria that were already drug-resistant. There are two types of drug-resistant tuberculosis: multiple drug-resistant tuberculosis and extensive drug-resistant tuberculosis.

- Multi-drug resistant tuberculosis (MDR-TB) is resistant to no less than two of the first-line of drugs used to fight TB infection.
- Extensively drug-resistant tuberculosis (XDR-TB) is resistant to three or more of the second-line of drugs used to fight TB infection. This makes it the worst kind of TB infection as treatment for the TB infection is drastically reduced.

The symptoms for MDR-TB and XDR-TB are similar to the symptoms experienced by individuals suffering with pulmonary TB. These symptoms are weight loss, fever, night sweats, coughing and/or coughing up blood, fatigue, and chest pain.

Fuzzy Logic (FL) is a branch of machine intelligence that helps computers paint pictures of uncertain world. Fuzzy sets were introduced by Zadeh (1965) as a means of representing and manipulating data that are not precise, but rather fuzzy. Fuzzy logic provides an inference morphology that enables appropriate human reasoning capabilities to be applied to knowledge-based systems. The theory of fuzzy logic provides a mathematical strength to capture the uncertainties associated with human cognitive processes, such as thinking and reasoning. A fuzzy set

A is called trapezoidal fuzzy number (Figure 1) with tolerance interval [a, b], left width **α** and right width **β** if its membership function has the following form

$$A(t) = \begin{cases} 1-(a-t)/\alpha & if\ a-\alpha \le t \le a \\ 1 & if\ a \le t \le b \\ 1-(t-b)/\beta & if\ a \le t \le b+\beta \\ 0 & otherwise \end{cases}$$

and we use the notation A = (a, b,**α**,**β**). It can easily be shown that

$[A]^{\gamma} = [a - (1 - \gamma)\alpha, b + (1 - \gamma)\beta], \forall \gamma \in [0, 1]$.

The support of A is (a − **α**, b + **β**).

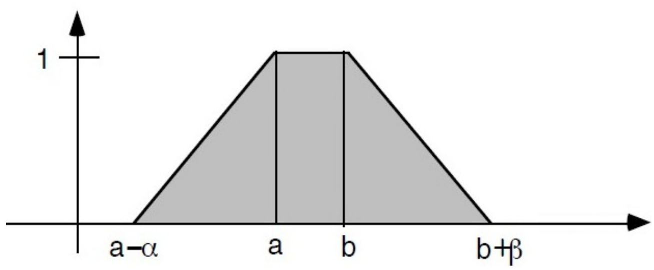


Figure 1: Trapezoidal fuzzy number.

X. Fuzzy Clustering

Clustering involves the task of dividing data points into homogeneous classes or clusters so that items in the same class are as similar as possible and items in different classes are as dissimilar as possible (Yang and Wang 2001). Clustering can also be thought of as a form of data compression, where a large number of samples are converted into a small number of representative prototypes or clusters (Giles and Draeseke 2001). Depending on the data and the application, different types of similarity measures may be used to identify classes, where the similarity measure controls how the clusters are formed (Inyang 2005). Some examples of values that can be used as similarity measures include distance, connectivity, and intensity (Berks et al 2000).

In non-fuzzy or hard clustering, data is divided into crisp clusters, where each data point belongs to exactly one cluster (Albayrak and Amasyali 2003). In fuzzy clustering, the data points can belong to more than one cluster, and associated with each of the points are membership grades which indicate the degree to which the data points belong to the different clusters (Nascimento, 1991).

XI. Overview of Fuzzy Clustering Means (FCM) Algorithm

The FCM algorithm is one of the most widely used fuzzy clustering algorithms. The FCM algorithm attempts to partition a finite collection of elements $X=\{X_1, X_2,...,X_n\}$ into a collection of c fuzzy clusters with respect to some given criterion. Given a finite set of data, the algorithm returns a list of c cluster centers V, such that

$V = V_i$, i=1, 2,..., c

and a partition matrix U such that

$U = U_{ij}$, i =1,..., c , j =1,..., n

where U_{ij} is a numerical value in [0, 1] that tells the degree to which the element X_j belongs to the i-th cluster.

The following is a linguistic description of the FCM algorithm, which is implemented by fuzzy Logic. The algorithm is as follows;

Step 1: Select the number of clusters c ($2 \le c \le n$), exponential weight **μ** ($1 < \mu < \infty$), initial partition matrix U^0, and the termination criterion **ε**. Also, set the iteration index 1 to 0.

Step 2: Calculate the fuzzy cluster centers $\{V_i^1 |\ i=1, 2,..., c\}$ by using U^1.

Step 3: Calculate the new partition matrix U^{1+1} by using $\{V_i^1 |\ i=1, 2,..., c\}$.

Step 4: Calculate the new partition matrix = $|| U^{1+1} - U^1 || = | U_{ij}^{1+1} - U_{ij}^1 |$. If >**ε**, then set l = l + 1 and go to **step 2**. If ≤**ε**, then stop.

The initial cluster centers are computed in two ways; Arithmetic Means of all the data points or running FCM several times each starting with different initial cluster centers. In this work the first method is adopted.

XII. Methodology

This expert system which employ fuzzy C-Means for the diagnosis of TB is developed in an environment characterized by Microsoft Window XP professional Operating System, Microsoft Access Database Management system, Visual Basic Application Language and Microsoft Excel.

Neuro Solutions and Crystal reports were used for neural network analysis and graphical representation. An approach for analyzing clusters to identify meaningful pattern for determining whether a patient suffers from TB or not is presented. The system provides a guide for diagnosis of TB within the decision-making framework.

The process for the medical diagnosis of TB starts when an individual consults a physician (doctor) and presents a set of complaints (symptoms). The physician then requests further information from the patient or from others close to him who knows about the patient's symptoms in severe cases. Data collected include patient's previous state of health, living condition and other medical conditions. A physical examination of the patient condition is conducted and in most cases, a medical observation along with medical test(s) is carried out on the patient prior to medical treatment.

From the symptoms presented by the patient, the physician narrows down the possibilities of the illness that corresponds to the apparent symptoms and make a list of the conditions that could account for what is wrong with the patient. These are usually ranked in possibility order (Low, Moderate and high). The physician then conducts a physical examination of the patient, studies his or her medical records and ask further questions, as he goes in an effort to rule out as many of the potential conditions as possible. When the list has been narrowed down to a single condition, it is called differential diagnosis and provides the basis for a hypothesis of what is ailing the patient. Until the physician is certain of the condition present; further medical test are performed or schedule such as medical imaging, scan, X-rays in part to conform or disprove the diagnosis or to update the patient medical history. Other Physicians, specialist and expert in the field may be consulted (sought) for further advices.

Despite all these complexities, most patient consultations are relatively brief because many diseases are obvious or the physician's experience may enable him to recognize the condition quickly. Upon the completion of the diagnosis by the physician, a treatment plan is proposed, which includes therapy and follow-up (further meeting and test to monitor the ailment and progress of the treatment if needed). Review of diagnosis may be conducted again if there is failure of the patient to respond to treatment that would normally work. The procedure of diagnosing a patient suffering from TB is synonymous to the general approach to medical diagnosis. The physician may carry out a precise diagnosis, which requires a complete physical evaluation to determine whether the patient have TB. The examining physician accounts for possibilities of having TB through an interview, physical examination and laboratory test. Many primary health care physicians use screening tools for TB evaluation.

A thorough diagnostic evaluation may include a complete history of the following:

a. When did the symptoms start?

b. How long have the symptoms lasted?

c. How severe are the symptoms?

d. Have the symptoms occurred before, and if so, were they treated and what treatment was received?

XIII. Results and Discussion

To design the FCM Knowledge Base System for diagnosis of TB, we design a system which consists of a set of parameters needed for diagnosis (here, we are using 20 basic and major parameters) presented in Table 1.

1.	Coma (Seizure)
2.	Stiff Neck
3.	Headache
4.	Abdominal Pain
5.	Weight Pain
6.	Fever
7.	Masses along the neck
8.	Draining Sinus
9.	Small Reddish brown lesions(face, eyelid, nose, cheek and ear
10.	Reddish brown wart-like growth on the body
11.	Skin lesions on hand, feet, elbow and knees.
12.	Ulcer or abscesses on the Skin
13.	Necrosis of infected Skin
14.	Stiffness of affected area
15.	Blood present in Urine
16.	Painful or uncomfortable Urination
17.	Hemopysis (coughing up blood)
18.	Fatigue
19.	Chest pain
20.	Night Sweat

Table: 1 Symptoms of Tuberculosis (TB)

Figure 1 presents the model of the FCM system for the diagnosis of TB. It comprises of knowledge base system, fuzzy c-means inference engine and decision support system. The knowledge base consists of the database Engine. The knowledge base system holds the symptoms for TB.

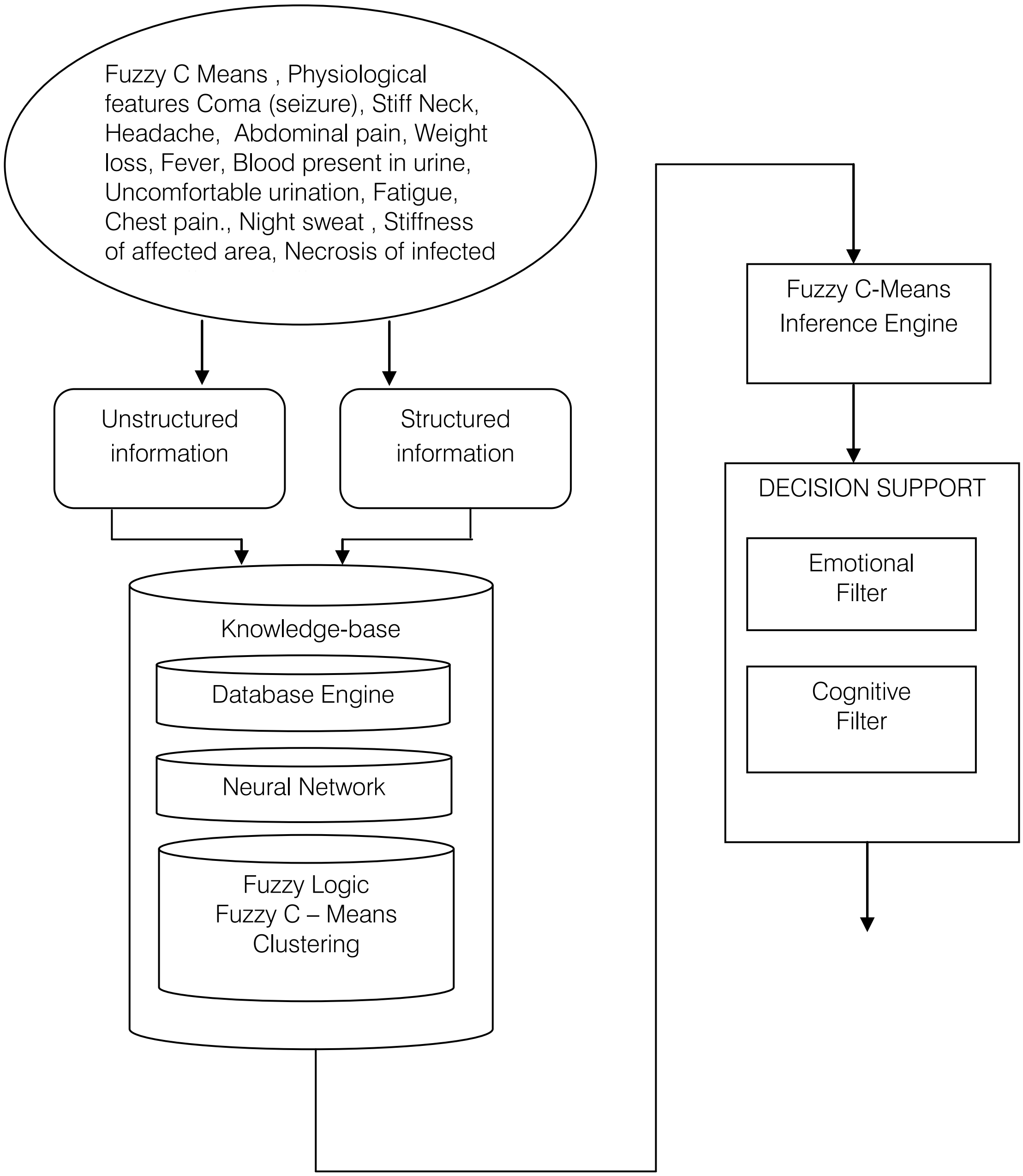


Figure 1: Architecture of FCM Knowledge Base System for the diagnosis of TB

The values for the diagnosis of TB symptoms are not precise hence the adoption of fuzzy logic as a means of analyzing these information. The different forms of TB symptoms, therefore, constitute the fuzzy parameters of the knowledge base system. The fuzzy set of parameters is represented by 'P', which is defined as $P = \{P_1, P_2 \ldots, P_n\}$

where P_i represents the j^{th} parameter and n is the number of parameters (in this case n=20). Neural network provides the structure intelligent learning for all forms of TB symptoms, which serves as a platform for the inference engine. The inference engine consists of reasoning algorithms, driven by production rules. These production rules are evaluated by using the forward chaining approach of reasoning. The fuzzy logic and fuzzy C-means algorithm provides the rules for the partitioning of patients into a number of homogenous clusters with respect to a suitable similarity measure.

In this paper, the patients were classified into seven form of TB according to physician. TB meningitis refer to patients experiencing four or less of the enlisted symptoms along the brain and the spinal cord, Gastrointestinal tuberculosis refers to patients who have 5 or more of the enlisted symptoms along the

Global Journal of Computer Science and Technology Volume XI Issue VI Version I April 2011

gastrointestinal tract (mouth, oesophagus, stomach, small and large intestine, and the anus), **TB lymphadenitis** refers to patients experiencing four or less of the under listed symptoms along the lymph nodes, (usually along the neck), **Cutaneous TB** refers to patients experiencing four or less of the under listed symptoms along the skin or mucous membrane, **Osteo-articular TB** refers to patients experiencing four or less of the under listed symptoms along the joints: knees, hips, ankles, wrists, shoulders, and elbows (It usually affects one joint). **TB genitourinary** refers to patients experiencing four or less of the under listed symptoms along the genitourinary tract includes the urinary tract and the reproductive system and **Drug-resistant TB** refers to patients experiencing four or less of the under listed symptoms after drugs has been prescribed to fight the disease.

Each of the symptoms highlighted in Table 1 is represented with P (starting from 1 – 20 i.e. P_1 – P_{20}). In addition, we form seven clusters namely, TB meningitis, gastrointestinal TB, TB lymphadenitis Cutaneous TB, Osteo-articular TB, TB genitourinary and Drug-resistant TB. These will make up the degree of Membership or intensity.

Table 2 below represents the degree of membership, for instance, P10 in cluster 1, we notice it has 0.51. In term of percentage it can be represented as 51%, in cluster 2, 9%, in cluster 3, 10%, in cluster 4, 5%, in cluster 5, 10%, in cluster 6, 5% and in cluster 7, 10%. This means that the degree of symptoms of P10 matches **51% of TB meningitis, 9% of Gastrointestinal TB**, **10% of TB lymphadenitis, 5% 0f Cutaneous TB, 10 % of Osteo-articular TB, 5% of TB genitourinary and 10 of Drug-resistant TB**.

The FCM clustering distribution shown Figure 2 depicts a total of two symptoms with high degree of membership of **TB meningitis,** eight symptoms with high degree of membership of **Gastrointestinal TB**, two symptoms with high degree of membership of **TB lymphadenitis,** two symptoms with high degree of membership of **Cutaneous TB,** two symptoms with high degree of membership of **Osteo-articular TB**, two symptoms with high degree of membership of **TB genitourinary** and two symptoms with high degree of membership of **Drug-resistant TB.**

Table 2: FCM membership grade of all patients in all clusters

CODES	DEGREE OF MEMBERSHIP						
	CLUSTER 1	CLUSTER 2	CLUSTER 3	CLUSTER 4	CLUSTER 5	CLUSTER 6	CLUSTER 7
P01	0.05	0.50	0.10	0.10	0.10	0.05	0.10
P02	0.50	0.05	0.10	0.05	0.05	0.15	0.10
P03	0.10	0.55	0.05	0.15	0.05	0.05	0.05
P04	0.15	0.10	0.10	0.05	0.05	0.05	0.50
P05	0.10	0.10	0.10	0.05	0.05	0.58	0.02
P06	0.05	0.05	0.05	0.05	0.53	0.17	0.10
P07	0.08	0.06	0.04	0.62	0.05	0.09	0.06
P08	0.08	0.10	0.56	0.10	0.02	0.10	0.04
P09	0.05	0.66	0.11	0.06	0.05	0.04	0.03
P10	0.51	0.09	0.10	0.05	0.10	0.05	0.10
P11	0.05	0.61	0.10	0.04	0.05	0.10	0.05
P12	0.10	0.20	0.50	0.05	0.0 7	0.04	0.04
P13	0.04	0.15	0.05	0.56	0.05	0.05	0.10
P14	0.08	0.05	0.05	0.03	0.72	0.03	0.04
P15	0.05	0.11	0.06	0.08	0.06	0.55	0.09
P16	0.06	0.04	0.16	0.12	0.04	0.04	0.54
P17	0.05	0.56	0.14	0.05	0.05	0.08	0.07
P18	0.05	0.65	0.10	0.05	0.05	0.05	0.05
P19	0.10	0.59	0.01	0.05	0.05	0.10	0.10
P20	0.09	0.51	0.10	0.05	0.10	0.05	0.10

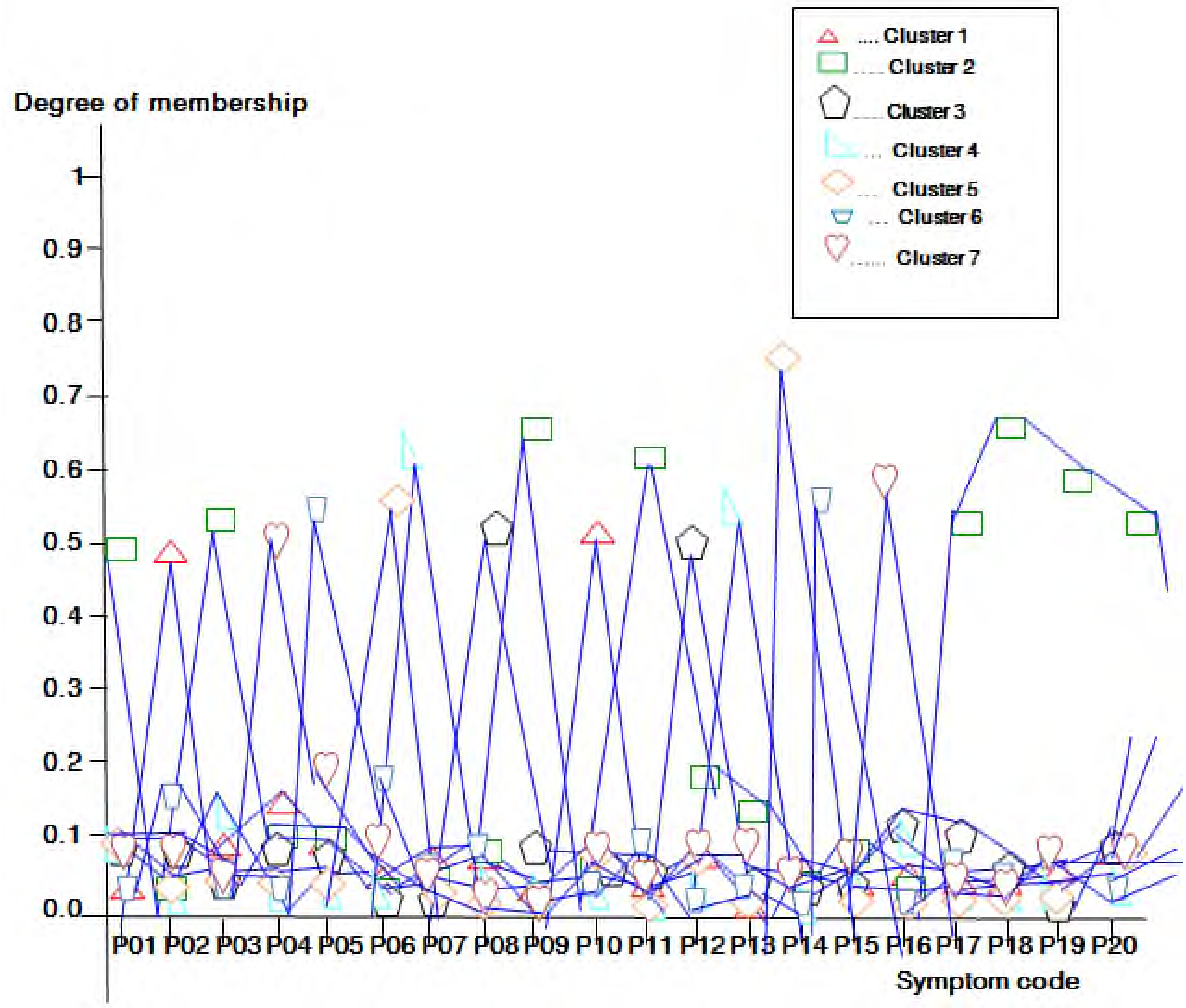


Figure 2: Graphical representation of Membership Grades of TB symptoms.

XIV. Conclusion

The need to design a system that would assist doctors in medical diagnosis has become imperative and hence cannot be over emphasized. This paper present a diagnostic fuzzy cluster means system to help in diagnosis of Tuberculosis using a set of symptoms. This advanced system which uses a set of clustered data set is more precise than the traditional system. The classification, verification and matching of symptoms to the seven groups of clusters was necessary especially in some complex scenarios. This paper demonstrates the practical application of IT (Information Technology) in the domain of diagnostic pattern appraisal by determining the extent of membership of individual symptoms. The model proposed allows for the classification of and matching of cluster groups to TB symptoms. The fuzzy- cluster means model proposed in this paper appears to be a more natural and intelligent way of classification and matching of symptoms to Tuberculosis groups.

References Références Referencias

1. Albayark S. and Amasyali F. (2003), "Fuzzy C-Means Clustering on Medical Diagnostic
2. Systems", International XII Turkish Symposium on Artificial Intelligence and Neural Networks.
3. Berks G., Keyserlingk D.G.V., Jantzeen J., Dotoli M. and Axer H. (2000): Fuzzy Clustering– A Versatile Mean to Explore Medical Database ESIT 2000, Aachen, Germany.
4. CWS. (2000), "Tuberculosis Research Paper", retrieved from http://writing4students.blogspot.com/2009/12/tuberculosis-research-paper.html
5. Giles D. E. A. and Draeseke R. (2001), "Economic Modeling Based on Pattern recognition via fuzzy C-Means clustering

algorithm", Department of Economics Working Paper EWP0101, University of Victoria, Canada.

6. Inyang U. G. (2004, " Neuro Fuzzy Expert System for Recognition of the Behavioural Patterns of some Selected Ethnic Groups in Nigeria", M.Tech. Thesis, Department of Computer Science, Federal University of Technology Akure, Nigeria.
7. MedicineNet (2011), "Tuberculosis", retrieved from http:// www.medicinenet.com main/art.asp?articlekey=505&page=4.
8. Nascimento L. (1991), "Comparative Genomus of two Leptospira interrogans Serovirs" retrieved from http//:jb.asm.org/cgi/content/full/186/7/2164.
9. Neil and Janet C. (2005), "History of Tuberculosis", retrieved from http:// www.micklebring.com/Oakwood/ch18.htm.
10. Nicole E. (2010), "Article Source: Different form of Tuberculosis", retrieved from EzineArticle.com??expore=Nicole Esbach.
11. Taura DW, Sale IT, Mohammed Y (2008), "The prevalence of tuberculosis in patients attending the infectious diseases hospital", Kano, Nigeria. Int. J. P. Appl. Sci., 2: 63-69.
12. Yang X. and Wand W. (2001), "GIS Based Fuzzy C-Means Clustering Analysis of Urban Transit Network Service", The Nanjing City Case Study. Road and Transport Research China.

Global Journal of Computer Science & Technology
Volume 11 Issue 6 Version 1.0 April 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

a New Three Dimensional Clustering Method for Wireless Sensor Networks

By Nima Attarzadeh, Mohamad Mehrani
Islamic Azad University

Abstract- : Today, the wireless sensor networks are popular therefore they have high features such as small tools, high computational ability and usability in all environments. But restriction power is the weakness that clustering is trying to improve. Clustering usually pays attention to two imensional WSNs. In this paper, three-dimensional geometric forms have been focused for some places in which reviewing different environments with different surfaces is necessary. Based on a mathematical model for the presented 3D clustering method we calculate the network life time according to the number of used sensors.

Keywords: Wireless sensor networks; WSNs; 3DWSNs; 3DClustering;

Classification: GJCST Classification: H.3.3, B.8.1

A NEW THREE DIMENSIONAL CLUSTERING METHOD FOR WIRELESS SENSOR NETWORKS

Strictly as per the compliance and regulations of:

a New Three Dimensional Clustering Method for Wireless Sensor Networks

Nima Attarzadeh[α], Mohamad Mehrani[Ω]

Abstract- Today, the wireless sensor networks are popular therefore they have high features such as small tools, high computational ability and usability in all environments. But restriction power is the weakness that clustering is trying to improve. Clustering usually pays attention to two imensional WSNs. In this paper, three-dimensional geometric forms have been focused for some places in which reviewing different environments with different surfaces is necessary. Based on a mathematical model for the presented 3D clustering method we calculate the network life time according to the number of used sensors.

Keywords- wireless sensor networks; WSNs; 3DWSNs; 3DClustering;

I. Introduction

Recently, the use of Wireless Sensor Networks (WSN) has been greatly expanded. Military, health and medicine, surveillance and even industrial applications are just a few of the fields where a WSN is already used extensively [1]. Sensors are generally equipped with data processing and communication capabilities [2], but the sensors have power limitations for sensing the environment. Using some efficient designs can be assumed as ways to optimize power consumption.

Clustering is a way that only one sensor (Cluster Head (CH)) is in charge of sending information to the base station and other sensors try to connect to CH [3],[6]. Clustering will be causing that all sensors are not sending information to base station but are sending information to CH which is nearer than base station, in other word, by doing such, less power will be consumed to send information. So the chosen CH has an important role on the network performance. CH should be chosen so that: 1) it should be placed at the center of the cluster with almost same distance from all the regular sensors in that cluster. 2) gathering sensors around it is appropriate 3) it has to have enough power for sending information to the base station.

Many attempts have been done to achieve at good clustering methods and selecting best nodes as cluster heads [2, 3, 4, 5]. Many ideas have been introduced for clustering that use geometric shapes to obtain appropriate clusters [5], but most of them have been raised in two-dimensional (2D) space [5, 6, 7], but always reviewing the space is not just at the surface with same heights. For example during reviewing and controlling the forests, sensors may be sit on tree with different heights or some factors and events that occur among foliage of trees should be considered. Therefore three-dimensional (3D) clustering is introduced. Recently, studying 3D WSNs have been considered, such as [7, 8, 9] but other clustering methods mostly tried to develop methods in 2D. This paper attempts to consider the characteristics of 3D space WSNs performance.

About[α] Islamic Azad University Mahshahr Branch Mahshahr, Iran.
E-mail- n.attarzadeh@mahshahriau.ac.ir
About[Ω] Islamic Azad University Shoushtar Branch, Shoushtar, Iran.
E-mail- m.mehrani@iau-shoushtar.ac.ir

II. The Proposed Method

As mentioned, two dimensional diagram shapes have less using in three dimensional areas. So, three dimensional diagram shapes like cube, cylinder or pyramid are to be used so much more in the proposed method. These shapes have different surfaces that are all monoliths and are dependent to each other based on a common law. For example in pyramid the above head has a same distance to all the bottom heads. In other word, the above head can be a cluster head because as mentioned it has one of the most important factors to be a cluster head. In the proposed method we try to use pyramid for partitioning sensor nodes.

In this idea there are three different levels for all the sensor nodes. The main node (sink) is placed at the first and highest level which is known as root. The cluster heads are at the middle level which name is CH level. This level is lower than root while it is higher in comparison with lowest level. The lowest level contains the most number of nodes and has the bigger size of area. This level includes active nodes that are in charge of receiving data from the region. At the rest of this paper we estimate each level and its functionality.

a) The root level

The root level contains some nodes that receive data from lower nodes and make best decisions according to received information. These nodes don't work so much in comparison with active nodes and consequently don't consume so much energy during functioning. The chance of each given node to be selected as a root node is so low.

b) The CH level

This level includes network cluster heads that are in charge of receiving data from lowest level nodes and manage the process of transmitting data from the active nodes to the root. Cluster heads should be selected in a way that the distance between each cluster head and every one of its active nodes is the same. By doing such, almost all the active nodes consume a same amount of energy to relate with their cluster head. It is clear that cluster heads have a very important role in designing the three dimensional network because they are between root and active nodes and also they have a same distance with their active members.

c) Active nodes level

This level has the most number of nodes that are all active and sense data from the region which is the main task of a sensor network. They are grouped into some clusters according to the position of above cluster heads. These nodes consume so much energy because of their very much activity. There are some energy limitations in WSNs, thus some substitute nodes for each active node can be considered. Substitute nodes should have the smallest distance to their active nodes and they are in waiting state while they receive an activation message from the active node.

III. Implementing The Proposed Method

We have a brief introduction for the different levels of our new designed network architecture for wireless mobile sensor networks, now we try to explain the detail of implementation. At first we disperse the sensor nodes in the region, randomly and without any denotation about sensor positions. Now, we can divide all the senor nodes into three different groups and let them know about their duty according to their position. Some pyramids with square bottom are used for grouping the sensors before network configuration.

We begin from the root (higher level). This level has the lowest amount of nodes, so a node with most score is selected as the root. To calculate the score, each node calculates its distance from the center of the region considering symmetry. Based on this factor, the sensor score can be calculated easily, and then the node sends its score to its same level neighbors and mutually receives some scores from them. By comparing all the scores the node decides whether to be active or awaiting. If a node decides to be active then it finds the position of its substitute node and sends it an activation message when necessary. Otherwise, the node sleeps in awaiting mode and listens to receive activation message from its active node. The root sensor divides its lower level sensors into four groups based on the node positions and sends them corresponding messages to inform them about this decision. The mentioned grouping is on the basis of a supposition square where the root position is at a certain height of the crossing of the square diagonals. Selecting the root node consumes two amounts of energy:

1. the energy that each node consumes to send its score to its same level neighbors, this message is called public message.
2. Consuming energy to inform lower level nodes for partitioning sensor nodes.

Each node placed at the next (third) level informs its neighbors that are positioned on the same square bottom about its score. This score is found on the distance of the node from the head of the square. In the all four mentioned groups a node with most score is selected as the cluster head for all the sensor nodes placed in that area while others can be waiting. Cluster heads assumes itself at the head of a supposition pyramid and divides its nodes into four groups by considering the square bottom of the pyramid. Doing such, has two kind of energy consumption for cluster head:

1. the energy that each node consumes to send its score to its same group neighbors.
2. Energy consumed for partitioning lower level sensor nodes.

At the lowest level sensor nodes evaluate their functionality sate (active or waiting) in the same manner. Now, the network starts to work.

During network functioning, the active nodes existing at the lowest level evaluate the regional events and send received information to their cluster heads in certain tome slices. When the remaining energy of the active node is going to fall below the threshold energy (energy needed for sending two activation messages) it would be replaced by its waiting node. We select the threshold energy equal to energy needed for sending two activation messages to make sure about receiving activation message which leads increasing the safety factor. Sensor replacement operation is same in all the levels.

IV. Network Simulation

We used Matlab software to simulate the proposed network architecture. The pseudo code of network initial configuration is shown in Figure 1 while network shape after configuration can be seen Figure 2. Network performance begins after configuration. The active nodes of the lowest level sense region date and forward them to their cluster heads. In parallel with this operation both of the active node and cluster head lose energy equal to sending energy and receiving energy, respectively.

An important note should be considered by active nodes is to have energy level more than twice needed energy for sending an activation message. If necessary, active node sends two activation messages to the substitute node which is now in waiting mode. Then the substitute node changes its state from waiting mode to active mode by sending an acceptance activation message and starts its duty as an active node. The last active node turns off or it dies, in other word.

Transmit sensor in space considered for wireless sensor network coverage Regarding a possible z component of each sensor values, the presence of 0,1,2 . Probability that value Z "2" is least likely, "1" is more likely and "0" is highest. Components x,y is determined randomly. Primary power as the same are given to all sensors.
Determine the root In this case the value of Eb (energy spent to send public messages, including points) and Amount of Er (energy spent to send a message to determine the lower level clusters) decrease of primary power active sensors.
Determine the CHs At this stage value Eb (energy spent to send public messages, including points) and value Ech (energy spent to send a message to determine the lower level clusters) decrease of primary power active sensors.
Characterize clusters At this stage, only one value Eb (power spent to send public messages, including points), decrease of the primary sensor.

Figure 1. Pseudo code of the network initial configuration

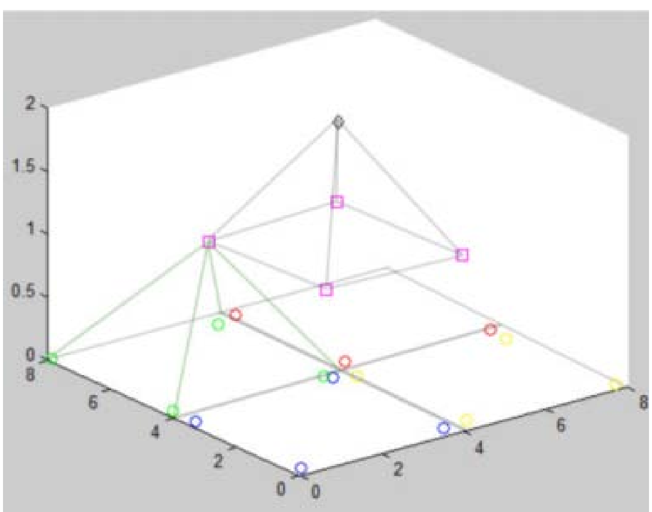


Figure 2. Network form after configuration

The mentioned operations are done until one of the following conditions appear:

1. Root sensors die
2. Cluster head dies or just cluster head is still alive while its members have died.
3. At least two clusters are active. (clusters die when they have just one active node)

The pseudo codes of these operations are written in Figure3.

While (true)
{
The active sensors of lower surface are reviwing environment and send self information to their CH. CH also sends information to root.
IF (only one cluster is active) **OR** (all CH sensors are death) **OR** (Root sensor is not available)
{
BREAK;
}
}

Figure 3. Network operation pseudo code

Now, we estimate network performance. Fig 4 shows the column chart of network lifetime for different number of sensors.

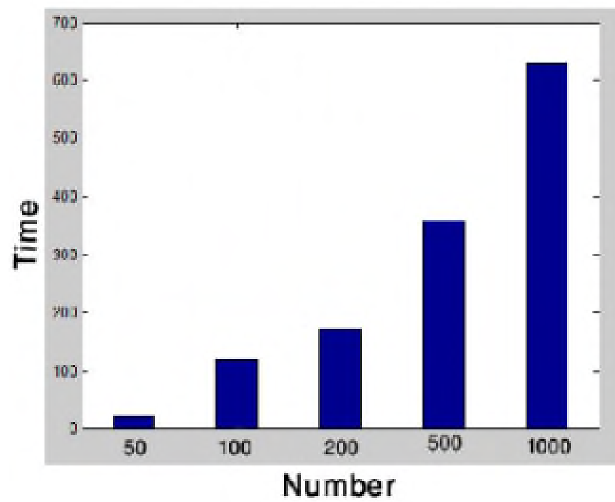


Figure 4. Network lifetime for different number of sensors

As can be seen in Figure 4, in parallel with increasing number of network nodes, network lifetime increases. It is natural because when the number of network nodes increases then the number of substitute nodes increases, consequently. In this situation when a node dies a substitute node will replace it quickly. The most important factor of the proposed architecture which is shown in Figure 4 is that similar to raising the number of sensors network lifetime raises. The rate of this similarity is very high. This can be considered in Figure 5 with more accurate rate. The slant of increasing network lifetime in respect of the number of sensors is almost constant. This leads network lifetime to be less dependent to the number of nodes.

April 2011
47
Global Journal of Computer Science and Technology Volume XI Issue VI Version I

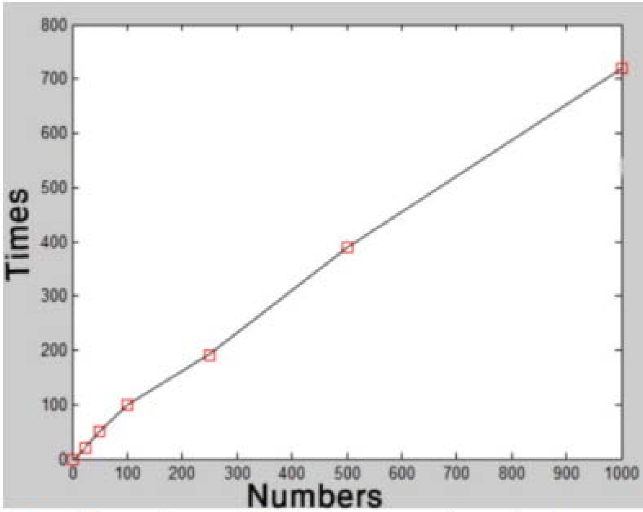


Figure 5. Increasing network lifetime in respect of the number of sensors

V. Mathematical Model

A very good point of the chart shown in Fig 5 is its almost constant slant which means network lifetime has lowest dependently amount to the number of nodes in the proposed architecture. To be more accurate, we propose a mathematical model using Newton-Raphson method. In this model, some calculated results in different states are needed. For this purpose, we calculate the network lifetime to the number of sensors in different moments. The problem is that the network lifetime is not always the same for the alike number of sensors because the network sensors are dispersed randomly. To solve this problem and to count the most exact amount for network lifetime, the algorithm is to be executed several times for the constant number of sensors and then we assume the average counted amounts as the network lifetime for each state. The formula (1) presents the mathematical mode of the algorithm.

$$T = 0.734*N + 13.32 \qquad (1)$$

T : Network lifetime

N: the number of network sensors

The TABLE I shows the average amount of the network lifetime for different number of network nodes. In this table, the average amount of the network lifetime is resulted from simulation. Difference between simulation and mathematical model mentions the difference of the lifetime to result of (1).

Table I. Comparison Between Simulation Value And Mathematical Model Value

Difference between simulation and mathematical model	Mathematical model value	Average life time of network	Sensor number
17.7	50.2	32.5	50
3.28	86.72	90	100
15.68	196.82	212.5	250
4.68	380.32	385	500
4.82	747.32	742.5	1000

As can be seen in the table there is not so much difference between the results of simulation and mathematical model. This shows the appropriate approximate of the mathematical model. The formula (1) has a constant slant which shows the lowest dependent of this slant to the number of nodes.

VI. Conclusion

As mentioned, to achieve at a good clustering or designing a three dimensional network areas, three dimensional diagram shapes can be used when two dimensional diagram shapes are not satisfying where these estimated areas don't have a same level. Also the diagram shapes can be used to simplify the network configuration. In the proposed design we achieve at good results by using pyramid diagram shapes with square bottom. First network lifetime has been calculated and then the dependent rate of network lifetime to the number of sensors achieved at its lowest amount which is the most important score of the proposed design.

References Références Referencias

1. K. BenkiK, M. Malajner, A. PeuliK, and Ž. MuKej, "Academic education Wireless Sensor Network: AeWSN," 50th International Symposium ELMAR-2008 Zadar, Croatia, 10-12 September 2008.
2. A. A. Abbasi, and M. Younis, " A survey on clustering algorithms for wireless sensor networks ," Computer Communications 30 (2007) 2826–2841, 21 June 2007.
3. S. Bandyopadhyay and E. J. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, 1713 - 1723 vol.3, 2003.
4. C. Y. Wen, and W. A. Sethares, "Automatic Decentralized Clustering forWireless Sensor Networks," EURASIP Journal onWireless

Communications and Networking, 28 March 2005.
5. D. Bein, W. W. Bein, and S. Malladi, "Fault Tolerant Coverage Model for Sensor Networks," ICCS 2005, LNCS 3515, pp. 535–542, 2005.
6. S. Banerjee, and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks," 2001.
7. M.K. Watfa and S. Commuri, "Optimality Measures for Coverage in 3D Wireless Sensor Networks", Wireless Pervasive Computing, 2006 1st International Symposium on, 2006.
8. S. Toriumi, Y. Sei and S. Honiden, "Energy-EfQcient Event Detection in 3D Wireless Sensor Networks", Wireless Days, 2008. WD '08. 1st IFIP, pages 1-5, Nov 2008.
9. H.M. Ammari and S.K. Das, "Critical Density for Coverage and Connectivity in Three-Dimensional Wireless Sensor Networks Using Continuum Percolation", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 6, JUNE 2009.
10. N. Attarzadeh, A. Barati, and A. Movaghar, "A New Method for Coverage in Wireless Sensor Networks" 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, Dec 2009.
11. M. K. Watfa, and S. Commuri, "Optimality measures for coverage in 3D wireless sensor networks",Wireless Pervasive Computing, 2006 st International Symposium on 16-18 Jan 2006.
12. D.R. Raymond, R.C. Marchany, M. I. Brownfield, and S.F.Midkiff, "Effects of Denial-of-Sleep Attacks on Wireless Sensor Network MAC Protocols", Vehicular Technology, IEEE Transactions on , Jan. 2009.
13. M.K.watfa, and S.Commuri, "A Coverage Algorithm in 3D Wireless Sensor Networks", Wireless Pervasive Computing, 2006 1st International Symposium on , 16-18 Jan. 2006.

This page is intentionally left blank

Global Journal of Computer Science & Technology
Volume 11 Issue 6 Version 1.0 April 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Fingerprint Identification

By Kuntal Barua, Samayita Bhattacharya, Dr. Kalyani Mali
University of Kalyani

Abstract- : Fingerprint identification is one of the most popular and reliable personal biometric identification methods. This paper describes an on-line fingerprint identification system consisting of image acquisition, edge detection, thinning, feature extractor and classifier. The preprocessing part includes steps to acquire binarized and skeletonized ridges, which are needed for feature point extraction. Feature points (minutia) such as endpoints, bifurcations, and core point are then extracted, followed by false minutia elimination. Human fingerprints are rich in details called minutiae, which can be used as identification marks for fingerprint verification. The goal of this project is to develop a complete system for fingerprint identification.

Keywords: Fingerprint, Minutia, Thinning, Edge Detection, Ridge, Bifurcation.

Classification: GJCST Classification: I.5.4, I.4.6

FINGERPRINT IDENTIFICATION

Strictly as per the compliance and regulations of:

Fingerprint Identification

Kuntal Barua[α], Samayita Bhattacharya[Ω], Dr. Kalyani Mali[β]

Abstract- Fingerprint identification is one of the most popular and reliable personal biometric identification methods. This paper describes an on-line fingerprint identification system consisting of image acquisition, edge detection, thinning, feature extractor and classifier. The preprocessing part includes steps to acquire binarized and skeletonized ridges, which are needed for feature point extraction. Feature points (minutia) such as endpoints, bifurcations, and core point are then extracted, followed by false minutia elimination. Human fingerprints are rich in details called minutiae, which can be used as identification marks for fingerprint verification. The goal of this project is to develop a complete system for fingerprint identification.

Keywords- Fingerprint, Minutia, Thinning, Edge Detection, Ridge, Bifurcation.

I. Introduction

Fingerprints are imprints formed by friction ridges of the skin and thumbs. They have long been used for identification because of their immutability and individuality. Immutability refers to the permanent and unchanging character of the pattern on each finger. Individuality refers to the uniqueness of ridge details across individuals; the probability that two fingerprints are alike is about 1 in 1.9x1015. However, manual fingerprint verification is so tedious, time consuming and expensive that is incapable of meeting today"s increasing performance requirements. An automatic fingerprint identification system is widely adopted in many applications such as building or area security and ATM machines. Our approach will be described in this project for fingerprint recognition:

Our approach is based on minutiae located in a fingerprint.

II. Approach

Most automatic systems for fingerprint comparison are based on minutiae matching [2] Minutiae are local discontinuities in the fingerprint pattern. A total of 150 different minutiae types have been identified. In practice only ridge ending and ridge bifurcation [5] minutiae types are used in fingerprint recognition [4]. Examples of minutiae are shown in figure 1.

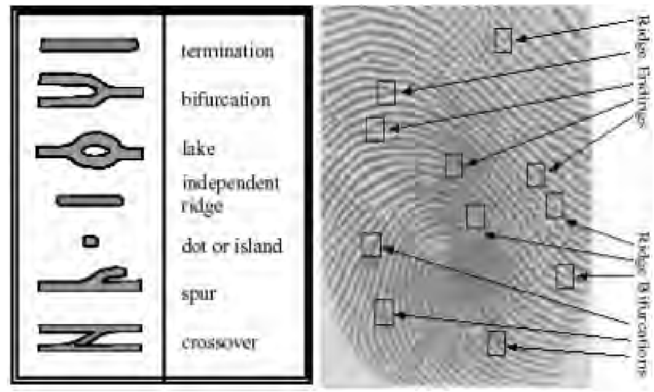


Figure 1. (a) Different minutiae types, (b) Ridge ending & Bifurcation

Many known algorithms have been developed for minutiae extraction based on orientation and gradients of the orientation fields of the ridges. In this project we will adopt the method used by Leung where minutiae are extracted using feed-forward artificial neural networks.

The building blocks of a fingerprint recognition system are: Image acquisition, Edge detection, Thinning, Feature extractor, Classifier. Figure 2 shows the building blocks.

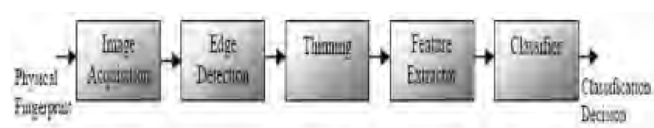


Figure 2. Fingerprint recognition system

a) *Image Acquisition*

The first stage of any vision system is the image acquisition stage. Image acquisition is hardware dependent. A number of methods are used to acquire fingerprints. Among them, the inked impression method remains the most popular one. Inkless fingerprint scanners are also present eliminating the intermediate digitization process [6].

i. *2D Image Input*

The basic two-dimensional image is a monochrome (grayscale) image which has been digitized. Describe image as a two-dimensional light intensity function f(x,y) where x and y are spatial coordinates and the value of f at any point (x, y) is proportional to the brightness or grey value of the image at that point.

About- Department of Computer Science & Engineering, University of Kalyani, Kalyani, West Bengal, India
E-mail[α]- kuntal.barua@gmail.com
E-mail[Ω]- samayita.bhattacharya@gmail.com
E-mail[β] - kalyanimali1992@gmail.com

April 2011

Global Journal of Computer Science and Technology Volume XI Issue VI Version I

A digitized image is one where

- spatial and grayscale values have been made discrete
- intensity measured across a regularly spaced grid in x and y directions
- intensities sampled to 8 bits (256 values)

Figure 3 shows a grayscale image.

Figure 3: Grayscale fingerprint image.

For computational purposes, we may think of a digital image as a two-dimensional array where x and y index an image point. Each element in the array is called a pixel (pictureelement).

90	71	61	51	49	40	35	53	56	99
93	74	53	56	48	45	48	72	55	102
101	60	57	53	54	52	64	62	66	101
107	82	64	63	59	60	81	90	93	100
114	93	76	59	72	85	84	99	85	99
117	108	94	92	97	101	100	106	105	99
116	114	109	106	105	108	108	102	107	110
115	113	109	114	111	111	113	105	111	115
110	113	111	109	106	108	110	115	120	122
103	107	105	108	109	114	120	124	124	132

Figure: Pixel values in highlighted region

ii. *3D Image Input*

A 3D image containing has many advantages over its 2D counterpart:

- 2D images give only limited information the physical shape and size of an object in a scene.
- 3D images express the geometry in terms of three-dimensional coordinates.

e.g Size (and shape) of an object in a scene can be straightforwardly computed from its three-dimensional coordinates.

b) *Edge Detection*

An edge is the boundary between two regions with relatively distinct gray level properties. The idea underlying most edge-detection techniques is on the computation of a local derivative operator such as „Sobel" operators [7]. In practice, the set of pixels obtained from the edge detection algorithm seldom characterizes a boundary completely because of noise,breaks in the boundary and other effects that introduce spurious intensity discontinuities. Thus, edge detection algorithms typically are followed by linking and other boundary detection procedures designed to assemble edge pixels into meaningful boundaries.

c) *Thinning*

Thinning [8] is a morphological operation that successively erodes away the foreground pixels until they are one pixel wide. A standard thinning algorithm is employed, which performs the thinning operation using two sub-iterations. This algorithm is accessible in MATLAB via the `thin' operation under the bwmorph function. Each sub-iteration begins by examining the neighbourhood of each pixel in the binary image, and based on a particular set of pixel-deletion criteria, it checks whether the pixel can be deleted or not. These sub-iterations continue until no more pixels can be deleted. The application of the thinning algorithm to a fingerprint image preserves the connectivity of the ridge structures while forming a skeletonised version of the binary image. This skeleton image is then used in the subsequent extraction of minutiae. An important approach to representing the structural shape of a plane region is to reduce it to a graph. This reduction may be accomplished by obtaining the skeleton of the region via thinning (also called skeletonizing) algorithm. The thinning algorithm while deleting unwanted edge points should not:

- Remove end points.
- Break connectedness
- Cause excessive erosion of the region

d) *Feature Extraction*

Extraction of appropriate features is one of the most important tasks for a recognition system. The feature extraction method used in will be explained below. A multilayer perceptron [9] (MLP) [10] of three layers is trained to detect the minutiae in the thinned fingerprint image of size 300x300. The first layer of the network has nine neurons associated with the components of the input vector. The hidden layer has five neurons and the output layer has one neuron. The network is trained to output a "1" when the input window in centered on a minutiae and a "0" when it is not. Figure 3 shows the initial training patterns which are composed of 16 samples of bifurcations in eight different orientations and 36 samples of non-bifurcations.

State the number of epochs needed for convergence as well as the training time for the two methods. Once the network is trained, the next step is to input the prototype fingerprint images to extract the minutiae. The fingerprint image is scanned using a 3x3 window given

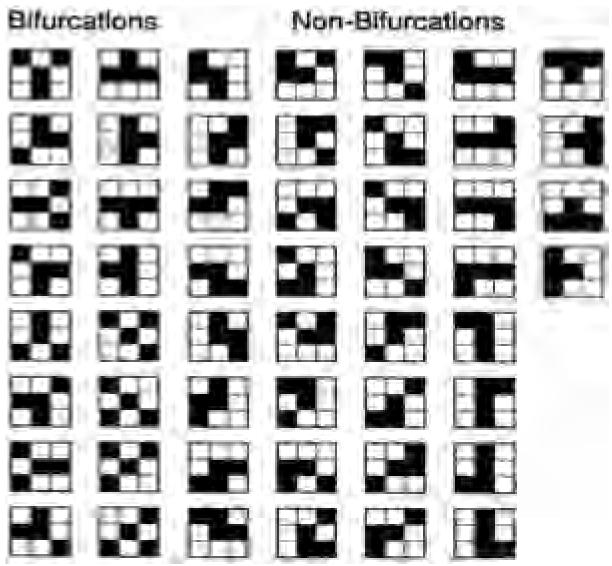


Figure 4: *Training set*

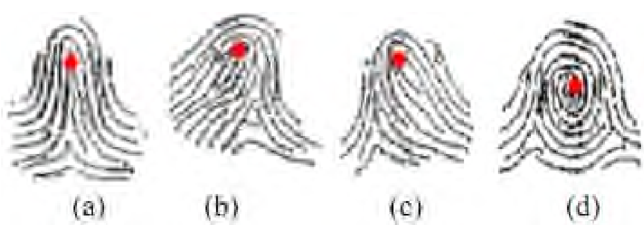


Figure 5. Core points on different fingerprint patterns. (a) tented arch, (b) right loop, (c)left loop, (d) whorl.

e) *Classifier*

After scanning the entire fingerprint image, the resulting output is a binary image revealing the location of minutiae. In order to prevent any falsely reported output and select "significant" minutiae, two more rules are added to enhance the robustness of the algorithm:

1) At those potential minutiae detected points, we re-examine them by increasing the window size by 5x5 and scanning the output image.

2) If two or more minutiae are to close together (few pixels away) we ignore all of them.

To insure translation, rotation and scale-invariance, the following operations will be performed:

• The Euclidean distance d(i) from each minutiae detected point to the center is calculated. The referencing of the distance data to the center point guarantees the property of positional invariance.

• The data will be sorted in ascending order from d(0) to d(N), where N is the number of detected minutiae points, assuring rotational invariance.

• The data is then normalized to unity by shortest distance d (0), i.e: dnorm(i) = d(0)/d(i); This will assure scale invariance property.

In the algorithm described above, the center of the fingerprint image was used to calculate the Euclidean distance between the center and the feature point. Usually, the center or reference point of the fingerprint image is what is called the "core" point. A core point, is located at the approximate center, is defined as the topmost point on the innermost upwardly curving ridgeline.

The human fingerprint is comprised of various types of ridge patterns, traditionally classified according to the decades-oldHenry system: left loop, right loop, arch, whorl, and tented arch. Loops make up nearly 2/3 of all fingerprints, whorls are nearly 1/3, and perhaps 5-10% are arches. Figure 5 shows some fingerprint patterns with the core point marked. Many singularity point detection algorithms were investigated to locate core points. For simplicity we will assume that the core point is located at the center of the fingerprint image.

After extracting the location of the minutiae for the prototype fingerprint images, the calculated distances will be stored in the database along with the ID or name of the person to whom each fingerprint belongs.

The last phase is the verification phase where testing fingerprint image:

1. is inputted to the system
2. minutiae are extracted
3. Minutiae matching: comparing the distances extracted minutiae to the one stored in the database
4. Identify the person: State the results obtained (i.e: recognition rate).

III. Conclusion

Our approach is based on minutiae located in a fingerprint. We further want to implement the fingerprint identification system based on a different approach, namely frequency content and ridge orientation of a fingerprint. The reliability of any automatic fingerprint recognition system strongly relies on the precision obtained in the minutiae extraction process. The minutiae based matching is highly sensible, as, if the finger is moved even a little bit that gives us a different set of minutiae.

References Références Referencias

1. Fingerprint Minutiae Extraction, Department of Computer Science National Tsing Hua University Hsinchu, Taiwan 30043
2. Fingerprint Recognition using Minutiae Score Matching by Ravi J, K. B. Raja and Venugopal K. R.
3. Jain, L. Hong and R. Boler, "Online Fingerprint Verification", IEEE trans, 1997
4. An Algorithm for Fingerprint Image Postprocessing by Marius Tico, Pauli Kuosmanen
5. Ridges and Furrows – history and science of fingerprint Identification, technology and legal

Global Journal of Computer Science and Technology Volume XI Issue VI Version I April 2011

issues: http://ridgesandfurrows.homestead.com/fingerprint.html
6. Image Digitization Process by Dorothy VanDeCarr
7. http://futura.disca.upv.es/~imd/cursosAnteriors/2k7-2k8/tmp/sobel3.pdf
8. An Algorithm for Fingerprint Image Postprocessing by Marius Tico, Pauli Kuosmanen.
9. Multilayer Perceptrons by Jasmin Steinwender (jsteinwe@uos.de), Sebastian Bitzer (sbitzer@uos.de).
10. Multilayer perceptrons and backpropagation Learning by Sebastian Seung.
11. Fingerprint Image Enhancement and Minutiae Extraction By Raymond Thai.
12. Fingerprint Matching using Gabor Filters by Muhammad Umer Munir and Dr. Muhammad Younas Javed.
13. Polemi, D. Biometric Techniques: Review and Evaluation of Biometric Techniques For Identification and Authentication, Including An appraisal of the Areas Where They are most Applicable: http://en.infosoc.gr/content/downloads/biomet.pdf.

Global Journal of Computer Science & Technology
Volume 11 Issue 6 Version 1.0 April 2011
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

The Analysis And Implementation Of OMNeT++ Framework

By K. Sivakumar , G. Dalin

Abstract- : The OMNeT++ is basically a collection of software tools and libraries which you can use to build your own simulation models. OMNeT++ is an object oriented modular discrete event network simulation framework. It has a generic rchitecture, so it can be used in various problem domains, such as, validating of hardware architecture, evauating performence aspects of the complex software systems, protocol modelling etc.. OMNeT++ simulations can be run under various use interfaces. When building and running the OMNeT++ simulation we must consider the topology, messages and the simulation system provides the simulation kernel and user interfaces. This paper presents the implementation and analysis of OMNeT++ framework, visualizing results with plove and scalars.

Keywords: *Simulation, NED language, modules, parameters, messages, analysis, IDE, simulation kernel, discrete, eventlog, own IDE modules & packet.*

Classification: *GJCST Classification: C.2.4*

THE ANALYSIS AND IMPLEMENTATION OF OMNET++ FRAMEWORK

Strictly as per the compliance and regulations of:

The Analysis and Implementation of OMNeT++ Framework

K. Sivakumar[α], G. Dalin[Ω]

Abstract- The OMNeT++ is basically a collection of software tools and libraries which you can use to build your own simulation models. OMNeT++ is an object oriented modular discrete event network simulation framework. It has a generic rchitecture, so it can be used in various problem domains, such as, validating of hardware architecture, evauating performence aspects of the complex software systems, protocol modelling etc.. OMNeT++ simulations can be run under various use interfaces. When building and running the OMNeT++ simulation we must consider the topology, messages and the simulation system provides the simulation kernel and user interfaces. This paper presents the implementation and analysis of OMNeT++ framework, visualizing results with plove and scalars.

Keywords: Simulation, NED language, modules, parameters, messages, analysis, IDE, simulation kernel, discrete, eventlog, own IDE modules & packet.

I. Introduction

The OMNeT++ IDE is based on the Eclipse platform which is an extensible, Java based framework. While it started as an IDE framework only, its main goal is to be a generic integration platform. OMNeT++ adds functionality for creating and configuring models (NED and ini files), performing batch executions, and analyzing simulation results, while Eclipse provides C++ editing, SVN/GIT integration, and other optional features via various open-source and commercial plug-ins. The OMNeT++ IDE is in fact an Eclipse istallation with some additional - simulaton related - tools pre-installed:

• The OMNeT++ feature which contains all OMNeT++ specific tools you use: the NED, MSG and INI file editor, simulation launcher, result analysis tools, sequence char view, documentation generator etc.

• CDT (C/C++ Development Tooling - eclipse.org/cdt) - for C++ development and debugging. This feature integrates with the standard gcc toolchain and the gdb debugger.

If you would like to develop your own plugins for the IDE you will need to install some additional components manuall

* JDT (Java Development Tools) for java development. JDT contains a java compiler and all the editors and debuggers and tools used during java development.

* PDE (Plug-in Development Environment) - this component contains additional tools. API definitions and documentation for developing plugins. PDE requires the presence of JDT.

a) The NED Editor

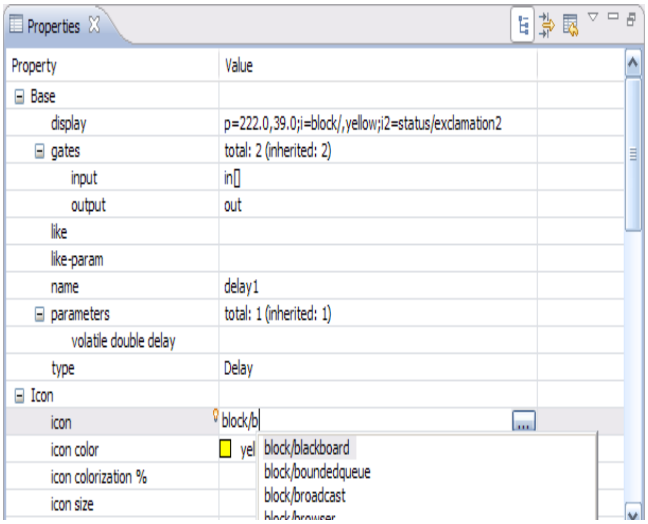
The NED Editor can edit NED files both graphically or in text mode, and the user can switch between two modes at any time, using the tabs at the bottom of the editor window In graphical mode, one can create compound modules, channels and other component types. Palettes are used to submodules from available module types.

Properties view is used to modify the visual and non-visual properties of the context menu. The NED Editor provide the following graphical features,

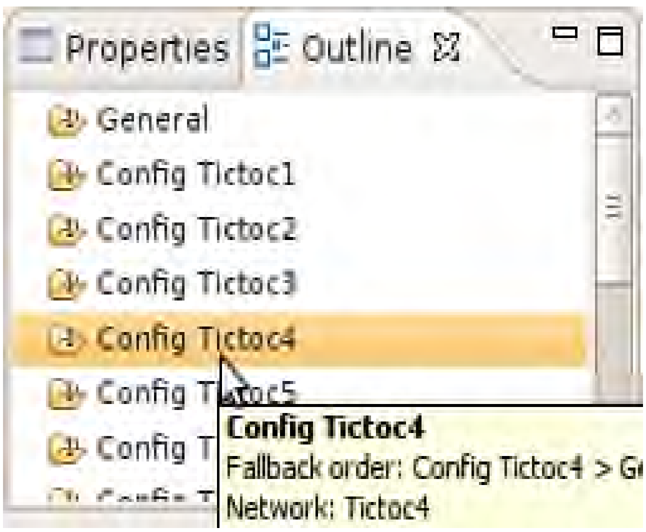
* Editing background image.
* Editing background grid.
* Default icons.
* Icon sizing and coloring.
* Transmission range and many others.

The properties view lets the user edit graphical and non- graphical properties of objects. Special cell editors facilitate selecting colors, icons etc., Undo and Redo is supported for property changes too. The properties view is also used with other editors like the Result analysis editor, where it allows the user to customize charts and other objects. The NED source is continually parsed and validated as the user is, typing and errorsare displayed in real time on the left margin. Syntax highlighting automatic indentation, and automatic conversion from the OMNeT++ NED syntax are also provided. NED parameters view are also used in NED editor.

About[α] - M.Sc., MCA., M.Phil, Assistant Professor SNMV CAS
E-mail- ksivadreams@gmail.com.
About[Ω] - M.SC(SS)., Assistant Professor SNMV CAS
E-mail- gdalinmsc@gmail.com



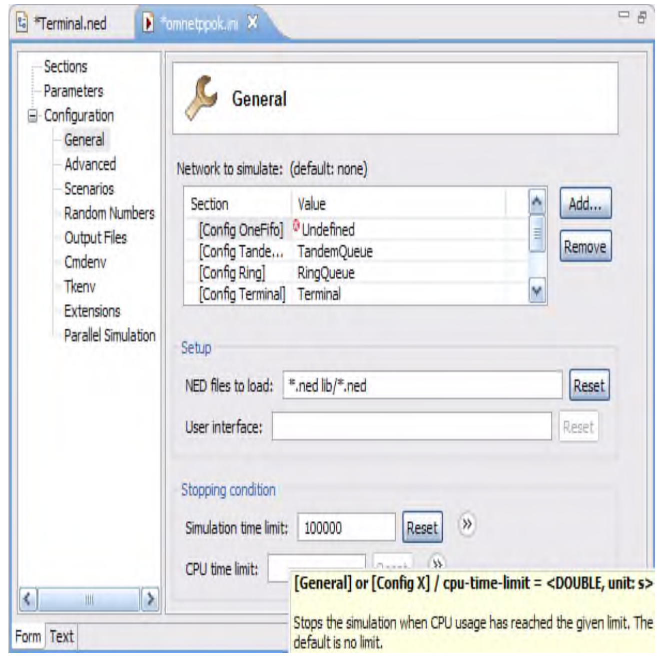
Properties View



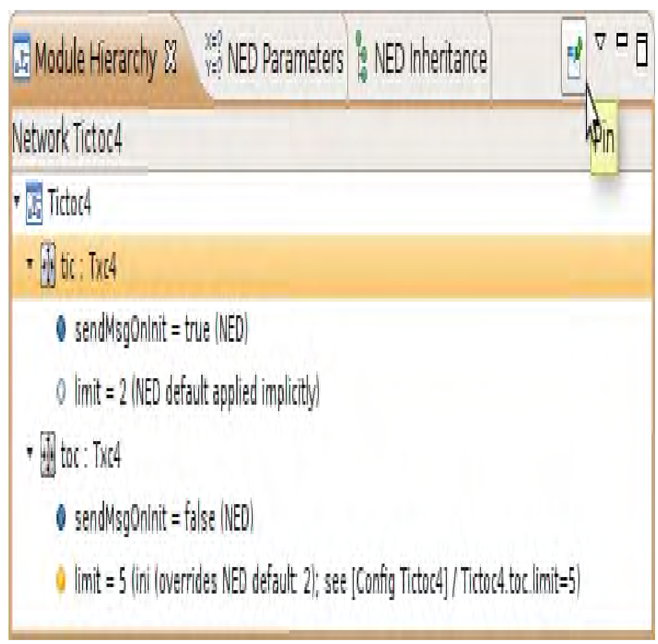
Outline View

b) *The INI File Editor*

The Ini File editor is used to support for user to configure simulation models for execution. The Ini Editor is used for both form based and source editing. The structure of the ni file is visualized and editable using Drag and drop and dialogs. The user can easily work under the text editor. The content of the INI file is divided into sections.



Form based INI File Editing



Module Hierarchy View

On the first page of the form editor, we can edit the sections. The sections are displayed as a tree, the nodes inherit the settings from parent. The INI file editor considers all supported configuration options and offers them in several forms, organized by topics.

II. Building OMNeT++ Application

Various views are available in INI Editor. These views are displayed by choosing the view from window.

1. *Outline View*

It provide the overview of sections in the current INI file.

2. *Problem View*

The problem view is used to view the errors and warning messages. The parser is generate these kinds of messages.

3. *Module Hierarchy view*

It shows the submodules, module parameters and where its values comes from.

4. Parameters view

The selected section parameters are displayed in this view. It also displays the inherited parameters and unassigned in the configuration.

5. NED Inheritance view

It display the inheritance tree of the network configured in the selected section.

These views can be displayed by choosing the view from window, then select view submenu.

a) The NED Language

OMNeT++ is an object oriented modular discrete event simulator. The abbreviation of OMNeT++ is "Objective Modular Network Testbed in C++". Each entity in a simulation needs to communicate via messages with itself and other entities. messages can be used for many purposes. One is to represent sticks that are available. Another one is to convey information of the entity. Simple modules are defined in NED file by their i) Parameters ii) Gates.

* Parameters:

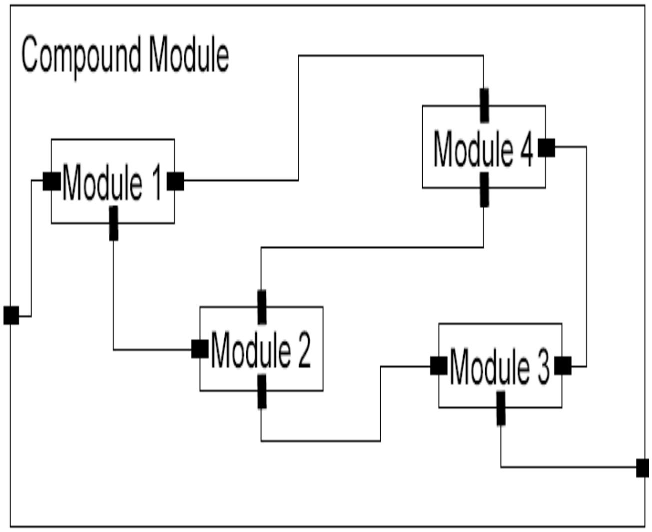
Values that can be set from outside the simulation program. Parameters can be easily accessed from the C++ code using cModule's method

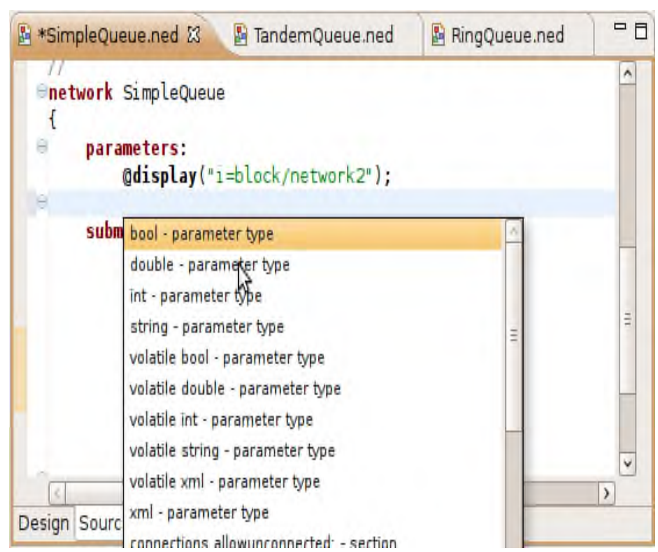
* Gates:

In NED, the gates of simple modules are defined as well as either in or out gates.

Gates can also be defined as arrays. Gates encapsulate the knowledge where to within the module.

The NED compound modules consist of one or more sub modules. It has two things such as Inside & Outside. Gates of the compound modules are connected to gates of the composing modules. Parameters of compound modules are similar to simple modules. Gates of the compound modules are identical to simple module gates.





NED Functions in Editor

Submodules section of a compound module defines which modules constitute the compound module. Submodules can be written as vectors of modules. Module type of sub modules need not be specified explicitly, can be left as a parameter, but the interface must be declared. The NED language describes links among modules and it also describes composition model. It has two divisions, it may be text edited or GUI.

b) NED Functions

The following functions are used in NED expressions and INI files. The NED functions are classified into various categories. The categories are,

c) Conversion

Conversion functions are used to convert on type to other. Such as double, int and string.

Global Journal of Computer Science and Technology Volume XI Issue VI Version I April 2011

April 2011

Global Journal of Computer Science and Technology Volume XI Issue VI Version I

d) Math

For mathematical calculations, we using the following functions.

1. fabs: Returns the absolute value of the quantity.
2. fmod: Returns floating-point remainder.
3. max: Returns the greater one of two quantities.
4. min: Returns the smaller one of two quantities. Trigonometric functions also used in math NED.

e) NED

More number of functions used in NED, and it is based on the module or channel.

1. ancestorIndex: Returns the index of the ancestor module.
2. FullName: Returns the full name of the module.
3. Fullpath: Returns the full path the module.
4. ParentIndex:Returns the index of the parent module.

The other NED function categories are as follows,

f) random/continuous

g) random/disrete

The NED language features are as folloows,

1. Hierachical :- A complex single entity is broken into smaller modules, and used as a compound module.
2. Compoenet Based Module:- These modules are inherently reusable and it allows component libraries.
3. Interfaces:- Interfaces module can be used as a placeholder where normally a module or channel type would be used.
4. Packages:- The NED language features a java-like package structure, to reduce the risk of name clashes between different models.

h) Modules

A complex problem or large peoblem is divided into smaller units is called Modules. Basically modules are used in two major reasons, such as, 1. To reduce the length of the code. 2. To increase the execution speed.

A discrete event system is a system where state changes happen at discrete instances in time, and events take zero time to happen. It is assumed that nothing happens between two consecutive events, that is, no state take place in the system between events. The time when event occur is often called event timestamp. With OMNeT++ we use the term arrival time. Time within the model is often called simulation time, model time or virtual time.

1. Events in OMNeT++

OMNeT++ uses messages to represent events. Each event is represented by an instance of the cMessage class or one of its subclasses; there is no separate event class. Messages sent from one module to another-this means that the place where the "event will occur" is the message's destination module, and the model time when the event occurs is the arrival time of the message.

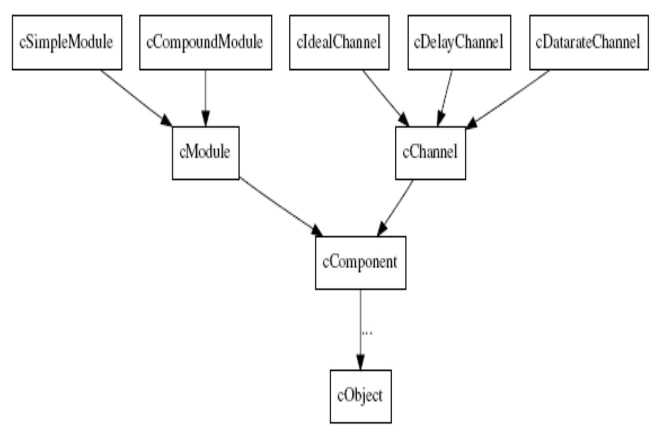
Events like "timeout expired" are implemented by the module sending a message to itself. Events are consumed from the FES in arrival time order, to maintain causality. More precisely, given two messages, the following rules apply:

* The one with the smaller scheduling priority value is executed first. If priorities are the same,

* The one scheduled or sent earlier is executed first. Add Scheduling priority is a user-assigned integer attribute of messages.

2. Components, Simple Modules and Channels

OMNeT++ simulation models are composed of modules and connections; Modules may be simple modules or compound modules. Simple modules are the active components. connections may have assoociated channel objects. Channel object encapsulate channel behavior, propagation and transmission time modeling, error modeling. Modules and Channels are called components.Components are represented with the c++ class cComponent. Simple Modules has two subclasses called cSimpleModule and cCompoundModule.



III. Implementation

The discrete event simulation can be divided into three main components, such as

1. *Core Platform.*
2. *Protocol Library.*
3. *Documentation.*

The main platform include lot of functionalities, based on the service. The major functionalities are,

1. *User Interface.*
2. *Simulation Engine.*
3. *Result Alalysis Tool.*

The network simulation is done using the following steps

1. *Set up the topology.*
2. *Configure protocol parameters.*
3. *Set traffic pattern.*
4. *Run simulation.*
5. *Analyze the results.*

The core platforms two major functionalities are, User Interface may be graphical user interface (GUI), textual and command line or hybrid. Simulation engine is the unit that executes the discrete event simulation. It stores the event list and manipulates it.

a) Creating Projects

The OMNeT++ IDE contains CDT to help you develop the C++ part of your simulation. The OMNeT++ IDE several extensions to the standard CDT features for ease of use:

* Customized project creation dialog.

* Extended C++ toolchain for OMNeT++

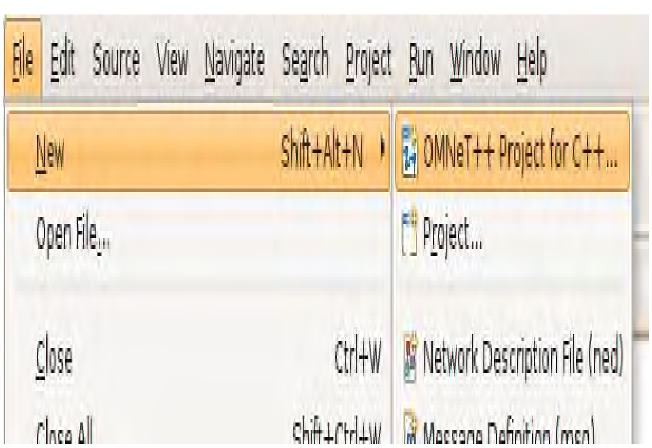
* Automatic makefile generation.

* OMNeT++ specific build configuration.

* Special launch configuration for launch & debug

1. Creating a C++ project

To create an OMNeT++ project that supports C++ development. Select New option from file menu. The following dialog box show the new OMNeT++ project wizard.



This will show the new OMNeT++ project wizaed that lets create a project that supports NED, MSG and INI file editing as well as C++ development for simple modules.

2. Configuring the Project

The OMNeT++ IDE adds several extensions to the standard CDT to make the building of simulations much easier. The project properties dialog is extended with a special page - OMNeT++ | Makemake - where you can configure the options used for automatically generating the makefile(s). A separate makefile will be created according to the settings you specify in the dialog. On the C/C++ Build page you can specify which makefile will be called by the IDE build process when you initiate a Build action. To configure the makefile the following options will be used.

* The target type can be either executable or a shared/static library.

* You may set the target name.

* The output directory can specify where the object files & the final target will be created.

3. Dependent Project

If our project uses code from other projects we should make the project dependent on that code. This step ensures that our code is linked with the dependent project's output and that the NED path can be correctly set. Include files are also automatically used from dependent projects.

b) Editing C++ code

The OMNeT++ IDE comes with a C++ editor provided by the CDT component. In addition to the standard editor features provided by the Eclipse environment, the C++ editor provides syntax highlighting and content assistance on the source code.

Use CTRL+SPACE to activate the content assist window anywhere in our source code. An other useful key is CTRL+TAB, which switches between your C++ and header file. Press CTRL+SHIFT+L to get a list of currently active key bindings.

c) Building the Project

When we create the source file, it is configured and it is converted into executable format. Once the makefile is configured correctly, it is transferred into project using build project option from the project menu. After the makefile is converted into project, we should switch to the console view to see the actual progress of the build. The project is debugged and we see the progress using the following views,

* Outline view
* Type Hierarchy view
* Problems view
* Console View

IV. Analyzing the Results

OMNeT++ is the statistical analysis tool is integrated into Eclipse Environment. When we creating an analysis, the user first selects the input of the analysis by specifying file names or file name patterns.

Data of interest can be selected into datasets using extra pattern rules. The user can define datasets by adding various processing, filtering and charting steps.

V. Conclusions

We presented a simulator focused application development and simulation point of view. OMNeT++ is the discrete event simulator mainly focused on the communication networks. OMNeT++ is designed to support parallel execution(MPI based), and it runs on both windows and Linux. The key elements of OMNeT++ programminh model is topology and behavior. Topology of module connections is specified using an OMNeT++ specific language called NED. OMNeT++ is a tool for discrete event simulation, managment of events is a primary task. Events are generated by modules sending messages to other modules orthemselves. It supports both finite state/event based simulation as well as process based simulation. OMNeT++ is the best tool for simulating the communication network and various research groups building the different models.

References Références Referencias

1. Gene Bellinger. (2004). Modeling & Simulation – An Introduction URL: http://www.systems-thinking.org/ simulation/model.htm
2. OMNeT++ Home Page. http://www.omnetpp.org
3. MiXiM home page.http://sourceforge.net/projects/mixim
4. H. Karl. Praxis der Simulation. course slides, Technische Universität Berlin.
5. Varga. OMNeT++: Object-Oriented Discrete. Event Simulator http://www.omnetpp.org/
6. David A. Cook. (2001). Computers and M&S - . Modeling and Simulation. The Journal of Defense Software Engineering
7. Patrick J. Delaney. "What is a Simulation? https://www.amso.army.mil/library/primers/what-is/
8. I.Dietrich, C. Sommer, F. Dressler. Simulating DYMO in OMNeT++.
9. S. Valentin. 2006. ChSim - A wireless channel simulator for OMNeT++,
10. Max Caceres, “Syscall Proxying - Simulating remote execution”,
11. Mattias Björlin. (2005) A study of Modeling and Simulation for computer and network security using OPNET
12. JiST home page.http://jist.ece.cornell.edu

Global Journals Inc. (US) Guidelines Handbook 2011

www.GlobalJournals.org

FELLOW OF INTERNATIONAL CONGRESS OF COMPUTER SCIENCE AND TECHNOLOGY (FICCT)

- FICCT' title will be awarded to the person after approval of Editor-in-Chief and Editorial Board. The title 'FICCT" can be added to name in the following manner e.g. **Dr. Andrew Knoll, Ph.D., FICCT, Er. Pettor Jone, M.E., FICCT**
- FICCT can submit two papers every year for publication without any charges. The paper will be sent to two peer reviewers. The paper will be published after the acceptance of peer reviewers and Editorial Board.
- Free unlimited Web-space will be allotted to 'FICCT 'along with subDomain to contribute and partake in our activities.
- A professional email address will be allotted free with unlimited email space.
- FICCT will be authorized to receive e-Journals - GJCST for the Lifetime.
- FICCT will be exempted from the registration fees of Seminar/Symposium/Conference/Workshop conducted internationally of GJCST (FREE of Charge).
- FICCT will be an Honorable Guest of any gathering hold.

ASSOCIATE OF INTERNATIONAL CONGRESS OF COMPUTER SCIENCE AND TECHNOLOGY (AICCT)

- AICCT title will be awarded to the person/institution after approval of Editor-in-Chef and Editorial Board. The title 'AICCTcan be added to name in the following manner:
  eg. **Dr. Thomas Herry, Ph.D., AICCT**
- AICCT can submit one paper every year for publication without any charges. The paper will be sent to two peer reviewers. The paper will be published after the acceptance of peer reviewers and Editorial Board.
- Free 2GB Web-space will be allotted to 'FICCT' along with subDomain to contribute and participate in our activities.
- A professional email address will be allotted with free 1GB email space.
- AICCT will be authorized to receive e-Journal GJCST for lifetime.
- A professional email address will be allotted with free 1GB email space.
- AICHSS will be authorized to receive e-Journal GJHSS for lifetime.

ANNUAL MEMBER

- Annual Member will be authorized to receive e-Journal GJMBR for one year (subscription for one year).
- The member will be allotted free 1 GB Web-space along with subDomain to contribute and participate in our activities.
- A professional email address will be allotted free 500 MB email space.

PAPER PUBLICATION

- The members can publish paper once. The paper will be sent to two-peer reviewer. The paper will be published after the acceptance of peer reviewers and Editorial Board.

Process of submission of Research Paper

The Area or field of specialization may or may not be of any category as mentioned in 'Scope of Journal' menu of the GlobalJournals.org website. There are 37 Research Journal categorized with Six parental Journals GJCST, GJMR, GJRE, GJMBR, GJSFR, GJHSS. For Authors should prefer the mentioned categories. There are three widely used systems UDC, DDC and LCC. The details are available as 'Knowledge Abstract' at Home page. The major advantage of this coding is that, the research work will be exposed to and shared with all over the world as we are being abstracted and indexed worldwide.

The paper should be in proper format. The format can be downloaded from first page of 'Author Guideline' Menu. The Author is expected to follow the general rules as mentioned in this menu. The paper should be written in MS-Word Format (*.DOC,*.DOCX).

The Author can submit the paper either online or offline. The authors should prefer online submission.Online Submission: There are three ways to submit your paper:

(A) (I) First, register yourself using top right corner of Home page then Login. If you are already registered, then login using your username and password.

(II) Choose corresponding Journal.

(III) Click 'Submit Manuscript'. Fill required information and Upload the paper.

(B) If you are using Internet Explorer, then Direct Submission through Homepage is also available.

(C) If these two are not convenient, and then email the paper directly to dean@globaljournals.org.

Offline Submission: Author can send the typed form of paper by Post. However, online submission should be preferred.

MANUSCRIPT STYLE INSTRUCTION **(Must be strictly followed)**

Page Size: 8.27" X 11'"

- Left Margin: 0.65
- Right Margin: 0.65
- Top Margin: 0.75
- Bottom Margin: 0.75
- Font type of all text should be Times New Roman.
- Paper Title should be of Font Size 24 with one Column section.
- Author Name in Font Size of 11 with one column as of Title.
- Abstract Font size of 9 Bold, “Abstract” word in Italic Bold.
- Main Text: Font size 10 with justified two columns section
- Two Column with Equal Column with of 3.38 and Gaping of .2
- First Character must be two lines Drop capped.
- Paragraph before Spacing of 1 pt and After of 0 pt.
- Line Spacing of 1 pt
- Large Images must be in One Column
- Numbering of First Main Headings (Heading 1) must be in Roman Letters, Capital Letter, and Font Size of 10.
- Numbering of Second Main Headings (Heading 2) must be in Alphabets, Italic, and Font Size of 10.

You can use your own standard format also.
Author Guidelines:

1. General,

2. Ethical Guidelines,

3. Submission of Manuscripts,

4. Manuscript’s Category,

5. Structure and Format of Manuscript,

6. After Acceptance.

1. GENERAL

Before submitting your research paper, one is advised to go through the details as mentioned in following heads. It will be beneficial, while peer reviewer justify your paper for publication.

Scope

The Global Journals Inc. (US) welcome the submission of original paper, review paper, survey article relevant to the all the streams of Philosophy and knowledge. The Global Journals Inc. (US) is parental platform for Global Journal of Computer Science and Technology, Researches in Engineering, Medical Research, Science Frontier Research, Human Social Science, Management, and Business organization. The choice of specific field can be done otherwise as following in Abstracting and Indexing Page on this Website. As the all Global

Journals Inc. (US) are being abstracted and indexed (in process) by most of the reputed organizations. Topics of only narrow interest will not be accepted unless they have wider potential or consequences.

2. ETHICAL GUIDELINES

Authors should follow the ethical guidelines as mentioned below for publication of research paper and research activities.

Papers are accepted on strict understanding that the material in whole or in part has not been, nor is being, considered for publication elsewhere. If the paper once accepted by Global Journals Inc. (US) and Editorial Board, will become the copyright of the Global Journals Inc. (US).

Authorship: The authors and coauthors should have active contribution to conception design, analysis and interpretation of findings. They should critically review the contents and drafting of the paper. All should approve the final version of the paper before submission

The Global Journals Inc. (US) follows the definition of authorship set up by the Global Academy of Research and Development. According to the Global Academy of R&D authorship, criteria must be based on:

1) Substantial contributions to conception and acquisition of data, analysis and interpretation of the findings.

2) Drafting the paper and revising it critically regarding important academic content.

3) Final approval of the version of the paper to be published.

All authors should have been credited according to their appropriate contribution in research activity and preparing paper. Contributors who do not match the criteria as authors may be mentioned under Acknowledgement.

Acknowledgements: Contributors to the research other than authors credited should be mentioned under acknowledgement. The specifications of the source of funding for the research if appropriate can be included. Suppliers of resources may be mentioned along with address.

Appeal of Decision: The Editorial Board's decision on publication of the paper is final and cannot be appealed elsewhere.

Permissions: It is the author's responsibility to have prior permission if all or parts of earlier published illustrations are used in this paper.

Please mention proper reference and appropriate acknowledgements wherever expected.

If all or parts of previously published illustrations are used, permission must be taken from the copyright holder concerned. It is the author's responsibility to take these in writing.

Approval for reproduction/modification of any information (including figures and tables) published elsewhere must be obtained by the authors/copyright holders before submission of the manuscript. Contributors (Authors) are responsible for any copyright fee involved.

3. SUBMISSION OF MANUSCRIPTS

Manuscripts should be uploaded via this online submission page. The online submission is most efficient method for submission of papers, as it enables rapid distribution of manuscripts and consequently speeds up the review procedure. It also enables authors to know the status of their own manuscripts by emailing us. Complete instructions for submitting a paper is available below.

Manuscript submission is a systematic procedure and little preparation is required beyond having all parts of your manuscript in a given format and a computer with an Internet connection and a Web browser. Full help and instructions are provided on-screen. As an author, you will be prompted for login and manuscript details as Field of Paper and then to upload your manuscript file(s) according to the instructions.

To avoid postal delays, all transaction is preferred by e-mail. A finished manuscript submission is confirmed by e-mail immediately and your paper enters the editorial process with no postal delays. When a conclusion is made about the publication of your paper by our Editorial Board, revisions can be submitted online with the same procedure, with an occasion to view and respond to all comments.

Complete support for both authors and co-author is provided.

4. MANUSCRIPT'S CATEGORY

Based on potential and nature, the manuscript can be categorized under the following heads:

Original research paper: Such papers are reports of high-level significant original research work.

Review papers: These are concise, significant but helpful and decisive topics for young researchers.

Research articles: These are handled with small investigation and applications

Research letters: The letters are small and concise comments on previously published matters.

5.STRUCTURE AND FORMAT OF MANUSCRIPT

The recommended size of original research paper is less than seven thousand words, review papers fewer than seven thousands words also.Preparation of research paper or how to write research paper, are major hurdle, while writing manuscript. The research articles and research letters should be fewer than three thousand words, the structure original research paper; sometime review paper should be as follows:

Papers: These are reports of significant research (typically less than 7000 words equivalent, including tables, figures, references), and comprise:

(a)Title should be relevant and commensurate with the theme of the paper.

(b) A brief Summary, "Abstract" (less than 150 words) containing the major results and conclusions.

(c) Up to ten keywords, that precisely identifies the paper's subject, purpose, and focus.

(d) An Introduction, giving necessary background excluding subheadings; objectives must be clearly declared.

(e) Resources and techniques with sufficient complete experimental details (wherever possible by reference) to permit repetition; sources of information must be given and numerical methods must be specified by reference, unless non-standard.

(f) Results should be presented concisely, by well-designed tables and/or figures; the same data may not be used in both; suitable statistical data should be given. All data must be obtained with attention to numerical detail in the planning stage. As reproduced design has been recognized to be important to experiments for a considerable time, the Editor has decided that any paper that appears not to have adequate numerical treatments of the data will be returned un-refereed;

(g) Discussion should cover the implications and consequences, not just recapitulating the results; conclusions should be summarizing.

(h) Brief Acknowledgements.

(i) References in the proper form.

Authors should very cautiously consider the preparation of papers to ensure that they communicate efficiently. Papers are much more likely to be accepted, if they are cautiously designed and laid out, contain few or no errors, are summarizing, and be conventional to the approach and instructions. They will in addition, be published with much less delays than those that require much technical and editorial correction.

The Editorial Board reserves the right to make literary corrections and to make suggestions to improve briefness.

It is vital, that authors take care in submitting a manuscript that is written in simple language and adheres to published guidelines.

Format

Language: The language of publication is UK English. Authors, for whom English is a second language, must have their manuscript efficiently edited by an English-speaking person before submission to make sure that, the English is of high excellence. It is preferable, that manuscripts should be professionally edited.

Standard Usage, Abbreviations, and Units: Spelling and hyphenation should be conventional to The Concise Oxford English Dictionary. Statistics and measurements should at all times be given in figures, e.g. 16 min, except for when the number begins a sentence. When the number does not refer to a unit of measurement it should be spelt in full unless, it is 160 or greater.

Abbreviations supposed to be used carefully. The abbreviated name or expression is supposed to be cited in full at first usage, followed by the conventional abbreviation in parentheses.

Metric SI units are supposed to generally be used excluding where they conflict with current practice or are confusing. For illustration, 1.4 l rather than 1.4 × 10-3 m3, or 4 mm somewhat than 4 × 10-3 m. Chemical formula and solutions must identify the form used, e.g. anhydrous or hydrated, and the concentration must be in clearly defined units. Common species names should be followed by underlines at the first mention. For following use the generic name should be constricted to a single letter, if it is clear.

Structure

All manuscripts submitted to Global Journals Inc. (US), ought to include:

Title: The title page must carry an instructive title that reflects the content, a running title (less than 45 characters together with spaces), names of the authors and co-authors, and the place(s) wherever the work was carried out. The full postal address in addition with the e-mail address of related author must be given. Up to eleven keywords or very brief phrases have to be given to help data retrieval, mining and indexing.

Abstract, used in Original Papers and Reviews:

Optimizing Abstract for Search Engines

Many researchers searching for information online will use search engines such as Google, Yahoo or similar. By optimizing your paper for search engines, you will amplify the chance of someone finding it. This in turn will make it more likely to be viewed and/or cited in a further work. Global Journals Inc. (US) have compiled these guidelines to facilitate you to maximize the web-friendliness of the most public part of your paper.

Key Words

A major linchpin in research work for the writing research paper is the keyword search, which one will employ to find both library and Internet resources.

One must be persistent and creative in using keywords. An effective keyword search requires a strategy and planning a list of possible keywords and phrases to try.

Search engines for most searches, use Boolean searching, which is somewhat different from Internet searches. The Boolean search uses "operators," words (and, or, not, and near) that enable you to expand or narrow your affords. Tips for research paper while preparing research paper are very helpful guideline of research paper.

Choice of key words is first tool of tips to write research paper. Research paper writing is an art.A few tips for deciding as strategically as possible about keyword search:

- One should start brainstorming lists of possible keywords before even begin searching. Think about the most important concepts related to research work. Ask, "What words would a source have to include to be truly valuable in research paper?" Then consider synonyms for the important words.
- It may take the discovery of only one relevant paper to let steer in the right keyword direction because in most databases, the keywords under which a research paper is abstracted are listed with the paper.
- One should avoid outdated words.

Keywords are the key that opens a door to research work sources. Keyword searching is an art in which researcher's skills are bound to improve with experience and time.

Numerical Methods: Numerical methods used should be clear and, where appropriate, supported by references.

Acknowledgements: Please make these as concise as possible.

References

References follow the Harvard scheme of referencing. References in the text should cite the authors' names followed by the time of their publication, unless there are three or more authors when simply the first author's name is quoted followed by et al. unpublished work has to only be cited where necessary, and only in the text. Copies of references in press in other journals have to be supplied with submitted typescripts. It is necessary that all citations and references be carefully checked before submission, as mistakes or omissions will cause delays.

References to information on the World Wide Web can be given, but only if the information is available without charge to readers on an official site. Wikipedia and Similar websites are not allowed where anyone can change the information. Authors will be asked to make available electronic copies of the cited information for inclusion on the Global Journals Inc. (US) homepage at the judgment of the Editorial Board.

The Editorial Board and Global Journals Inc. (US) recommend that, citation of online-published papers and other material should be done via a DOI (digital object identifier). If an author cites anything, which does not have a DOI, they run the risk of the cited material not being noticeable.

The Editorial Board and Global Journals Inc. (US) recommend the use of a tool such as Reference Manager for reference management and formatting.

Tables, Figures and Figure Legends

Tables: Tables should be few in number, cautiously designed, uncrowned, and include only essential data. Each must have an Arabic number, e.g. Table 4, a self-explanatory caption and be on a separate sheet. Vertical lines should not be used.

Figures: Figures are supposed to be submitted as separate files. Always take in a citation in the text for each figure using Arabic numbers, e.g. Fig. 4. Artwork must be submitted online in electronic form by e-mailing them.

Preparation of Electronic Figures for Publication

Even though low quality images are sufficient for review purposes, print publication requires high quality images to prevent the final product being blurred or fuzzy. Submit (or e-mail) EPS (line art) or TIFF (halftone/photographs) files only. MS PowerPoint and Word Graphics are unsuitable for printed pictures. Do not use pixel-oriented software. Scans (TIFF only) should have a resolution of at least 350 dpi (halftone) or 700 to 1100 dpi (line drawings) in relation to the imitation size. Please give the data for figures in black and white or submit a Color Work Agreement Form. EPS files must be saved with fonts embedded (and with a TIFF preview, if possible).

For scanned images, the scanning resolution (at final image size) ought to be as follows to ensure good reproduction: line art: >650 dpi; halftones (including gel photographs) : >350 dpi; figures containing both halftone and line images: >650 dpi.

Color Charges: It is the rule of the Global Journals Inc. (US) for authors to pay the full cost for the reproduction of their color artwork. Hence, please note that, if there is color artwork in your manuscript when it is accepted for publication, we would require you to complete and return a color work agreement form before your paper can be published.

Figure Legends: Self-explanatory legends of all figures should be incorporated separately under the heading 'Legends to Figures'. In the full-text online edition of the journal, figure legends may possibly be truncated in abbreviated links to the full screen version. Therefore, the first 100 characters of any legend should notify the reader, about the key aspects of the figure.

6. AFTER ACCEPTANCE

Upon approval of a paper for publication, the manuscript will be forwarded to the dean, who is responsible for the publication of the Global Journals Inc. (US).

6.1 Proof Corrections

The corresponding author will receive an e-mail alert containing a link to a website or will be attached. A working e-mail address must therefore be provided for the related author.

Acrobat Reader will be required in order to read this file. This software can be downloaded

(Free of charge) from the following website:

www.adobe.com/products/acrobat/readstep2.html. This will facilitate the file to be opened, read on screen, and printed out in order for any corrections to be added. Further instructions will be sent with the proof.

Proofs must be returned to the dean at dean@globaljournals.org within three days of receipt.

As changes to proofs are costly, we inquire that you only correct typesetting errors. All illustrations are retained by the publisher. Please note that the authors are responsible for all statements made in their work, including changes made by the copy editor.

6.2 Early View of Global Journals Inc. (US) (Publication Prior to Print)

The Global Journals Inc. (US) are enclosed by our publishing's Early View service. Early View articles are complete full-text articles sent in advance of their publication. Early View articles are absolute and final. They have been completely reviewed, revised and edited for publication, and the authors' final corrections have been incorporated. Because they are in final form, no changes can be made after sending them. The nature of Early View articles means that they do not yet have volume, issue or page numbers, so Early View articles cannot be cited in the conventional way.

6.3 Author Services

Online production tracking is available for your article through Author Services. Author Services enables authors to track their article - once it has been accepted - through the production process to publication online and in print. Authors can check the status of their articles online and choose to receive automated e-mails at key stages of production. The authors will receive an e-mail with a unique link that enables them to register and have their article automatically added to the system. Please ensure that a complete e-mail address is provided when submitting the manuscript.

6.4 Author Material Archive Policy

Please note that if not specifically requested, publisher will dispose off hardcopy & electronic information submitted, after the two months of publication. If you require the return of any information submitted, please inform the Editorial Board or dean as soon as possible.

6.5 Offprint and Extra Copies

A PDF offprint of the online-published article will be provided free of charge to the related author, and may be distributed according to the Publisher's terms and conditions. Additional paper offprint may be ordered by emailing us at: editor@globaljournals.org .

the search? Will I be able to find all information in this field area? If the answer of these types of questions will be "Yes" then you can choose that topic. In most of the cases, you may have to conduct the surveys and have to visit several places because this field is related to Computer Science and Information Technology. Also, you may have to do a lot of work to find all rise and falls regarding the various data of that subject. Sometimes, detailed information plays a vital role, instead of short information.

2. Evaluators are human: First thing to remember that evaluators are also human being. They are not only meant for rejecting a paper. They are here to evaluate your paper. So, present your Best.

3. Think Like Evaluators: If you are in a confusion or getting demotivated that your paper will be accepted by evaluators or not, then think and try to evaluate your paper like an Evaluator. Try to understand that what an evaluator wants in your research paper and automatically you will have your answer.

4. Make blueprints of paper: The outline is the plan or framework that will help you to arrange your thoughts. It will make your paper logical. But remember that all points of your outline must be related to the topic you have chosen.

5. Ask your Guides: If you are having any difficulty in your research, then do not hesitate to share your difficulty to your guide (if you have any). They will surely help you out and resolve your doubts. If you can't clarify what exactly you require for your work then ask the supervisor to help you with the alternative. He might also provide you the list of essential readings.

6. Use of computer is recommended: As you are doing research in the field of Computer Science, then this point is quite obvious.

7. Use right software: Always use good quality software packages. If you are not capable to judge good software then you can lose quality of your paper unknowingly. There are various software programs available to help you, which you can get through Internet.

8. Use the Internet for help: An excellent start for your paper can be by using the Google. It is an excellent search engine, where you can have your doubts resolved. You may also read some answers for the frequent question how to write my research paper or find model research paper. From the internet library you can download books. If you have all required books make important reading selecting and analyzing the specified information. Then put together research paper sketch out.

9. Use and get big pictures: Always use encyclopedias, Wikipedia to get pictures so that you can go into the depth.

10. Bookmarks are useful: When you read any book or magazine, you generally use bookmarks, right! It is a good habit, which helps to not to lose your continuity. You should always use bookmarks while searching on Internet also, which will make your search easier.

11. Revise what you wrote: When you write anything, always read it, summarize it and then finalize it.

12. Make all efforts: Make all efforts to mention what you are going to write in your paper. That means always have a good start. Try to mention everything in introduction, that what is the need of a particular research paper. Polish your work by good skill of writing and always give an evaluator, what he wants.

13. Have backups: When you are going to do any important thing like making research paper, you should always have backup copies of it either in your computer or in paper. This will help you to not to lose any of your important.

14. Produce good diagrams of your own: Always try to include good charts or diagrams in your paper to improve quality. Using several and unnecessary diagrams will degrade the quality of your paper by creating "hotchpotch." So always, try to make and include those diagrams, which are made by your own to improve readability and understandability of your paper.

15. Use of direct quotes: When you do research relevant to literature, history or current affairs then use of quotes become essential but if study is relevant to science then use of quotes is not preferable.

16. Use proper verb tense: Use proper verb tenses in your paper. Use past tense, to present those events that happened. Use present tense to indicate events that are going on. Use future tense to indicate future happening events. Use of improper and wrong tenses will confuse the evaluator. Avoid the sentences that are incomplete.

17. Never use online paper: If you are getting any paper on Internet, then never use it as your research paper because it might be possible that evaluator has already seen it or maybe it is outdated version.

18. **Pick a good study spot:** To do your research studies always try to pick a spot, which is quiet. Every spot is not for studies. Spot that suits you choose it and proceed further.

19. Know what you know: Always try to know, what you know by making objectives. Else, you will be confused and cannot achieve your target.

20. Use good quality grammar: Always use a good quality grammar and use words that will throw positive impact on evaluator. Use of good quality grammar does not mean to use tough words, that for each word the evaluator has to go through dictionary. Do not start sentence with a conjunction. Do not fragment sentences. Eliminate one-word sentences. Ignore passive voice. Do not ever use a big word when a diminutive one would suffice. Verbs have to be in agreement with their subjects. Prepositions are not expressions to finish sentences with. It is incorrect to ever divide an infinitive. Avoid clichés like the disease. Also, always shun irritating alliteration. Use language that is simple and straight forward. put together a neat summary.

21. Arrangement of information: Each section of the main body should start with an opening sentence and there should be a changeover at the end of the section. Give only valid and powerful arguments to your topic. You may also maintain your arguments with records.

22. Never start in last minute: Always start at right time and give enough time to research work. Leaving everything to the last minute will degrade your paper and spoil your work.

23. Multitasking in research is not good: Doing several things at the same time proves bad habit in case of research activity. Research is an area, where everything has a particular time slot. Divide your research work in parts and do particular part in particular time slot.

24. Never copy others' work: Never copy others' work and give it your name because if evaluator has seen it anywhere you will be in trouble.

25. Take proper rest and food: No matter how many hours you spend for your research activity, if you are not taking care of your health then all your efforts will be in vain. For a quality research, study is must, and this can be done by taking proper rest and food.

26. Go for seminars: Attend seminars if the topic is relevant to your research area. Utilize all your resources.

27. Refresh your mind after intervals: Try to give rest to your mind by listening to soft music or by sleeping in intervals. This will also improve your memory.

28. Make colleagues: Always try to make colleagues. No matter how sharper or intelligent you are, if you make colleagues you can have several ideas, which will be helpful for your research.

29. **Think technically:** Always think technically. If anything happens, then search its reasons, its benefits, and demerits.

30. Think and then print: When you will go to print your paper, notice that tables are not be split, headings are not detached from their descriptions, and page sequence is maintained.

31. Adding unnecessary information: Do not add unnecessary information, like, I have used MS Excel to draw graph. Do not add irrelevant and inappropriate material. These all will create superfluous. Foreign terminology and phrases are not apropos. One should NEVER take a broad view. Analogy in script is like feathers on a snake. Not at all use a large word when a very small one would be sufficient. Use words properly, regardless of how others use them. Remove quotations. Puns are for kids, not grunt readers. Amplification is a billion times of inferior quality than sarcasm.

32. Never oversimplify everything: To add material in your research paper, never go for oversimplification. This will definitely irritate the evaluator. Be more or less specific. Also too, by no means, ever use rhythmic redundancies. Contractions aren't essential and shouldn't be there used. Comparisons are as terrible as clichés. Give up ampersands and abbreviations, and so on. Remove commas, that are, not necessary. Parenthetical words however should be together with this in commas. Understatement is all the time the complete best way to put onward earth-shaking thoughts. Give a detailed literary review.

33. Report concluded results: Use concluded results. From raw data, filter the results and then conclude your studies based on measurements and observations taken. Significant figures and appropriate number of decimal places should be used. Parenthetical remarks are prohibitive. Proofread carefully at final stage. In the end give outline to your arguments. Spot out perspectives of further study of this subject. Justify your conclusion by at the bottom of them with sufficient justifications and examples.

34. After conclusion: Once you have concluded your research, the next most important step is to present your findings. Presentation is extremely important as it is the definite medium though which your research is going to be in print to the rest of the crowd. Care should be taken to categorize your thoughts well and present them in a logical and neat manner. A good quality research paper format is essential because it serves to highlight your research paper and bring to light all necessary aspects in your research.

Informal Guidelines of Research Paper Writing

Key points to remember:

- Submit all work in its final form.
- Write your paper in the form, which is presented in the guidelines using the template.
- Please note the criterion for grading the final paper by peer-reviewers.

Final Points:

A purpose of organizing a research paper is to let people to interpret your effort selectively. The journal requires the following sections, submitted in the order listed, each section to start on a new page.

The introduction will be compiled from reference matter and will reflect the design processes or outline of basis that direct you to make study. As you will carry out the process of study, the method and process section will be constructed as like that. The result segment will show related statistics in nearly sequential order and will direct the reviewers next to the similar intellectual paths throughout the data that you took to carry out your study. The discussion section will provide understanding of the data and projections as to the implication of the results. The use of good quality references all through the paper will give the effort trustworthiness by representing an alertness of prior workings.

Writing a research paper is not an easy job no matter how trouble-free the actual research or concept. Practice, excellent preparation, and controlled record keeping are the only means to make straightforward the progression.

General style:

Specific editorial column necessities for compliance of a manuscript will always take over from directions in these general guidelines.

To make a paper clear

· Adhere to recommended page limits

Mistakes to evade

- Insertion a title at the foot of a page with the subsequent text on the next page

- Separating a table/chart or figure - impound each figure/table to a single page
- Submitting a manuscript with pages out of sequence

In every sections of your document

· Use standard writing style including articles ("a", "the," etc.)

· Keep on paying attention on the research topic of the paper

· Use paragraphs to split each significant point (excluding for the abstract)

· Align the primary line of each section

· Present your points in sound order

· Use present tense to report well accepted

· Use past tense to describe specific results

· Shun familiar wording, don't address the reviewer directly, and don't use slang, slang language, or superlatives

· Shun use of extra pictures - include only those figures essential to presenting results

Title Page:

Choose a revealing title. It should be short. It should not have non-standard acronyms or abbreviations. It should not exceed two printed lines. It should include the name(s) and address (es) of all authors.

Abstract:

The summary should be two hundred words or less. It should briefly and clearly explain the key findings reported in the manuscript--must have precise statistics. It should not have abnormal acronyms or abbreviations. It should be logical in itself. Shun citing references at this point.

An abstract is a brief distinct paragraph summary of finished work or work in development. In a minute or less a reviewer can be taught the foundation behind the study, common approach to the problem, relevant results, and significant conclusions or new questions.

Write your summary when your paper is completed because how can you write the summary of anything which is not yet written? Wealth of terminology is very essential in abstract. Yet, use comprehensive sentences and do not let go readability for briefness. You can maintain it succinct by phrasing sentences so that they provide more than lone rationale. The author can at this moment go straight to shortening the outcome. Sum up the study, with the subsequent elements in any summary. Try to maintain the initial two items to no more than one ruling each.

- Reason of the study - theory, overall issue, purpose
- Fundamental goal
- To the point depiction of the research
- Consequences, including definite statistics - if the consequences are quantitative in nature, account quantitative data; results of any numerical analysis should be reported
- Significant conclusions or questions that track from the research(es)

Approach:

- Single section, and succinct
- As a outline of job done, it is always written in past tense
- A conceptual should situate on its own, and not submit to any other part of the paper such as a form or table
- Center on shortening results - bound background information to a verdict or two, if completely necessary
- What you account in an conceptual must be regular with what you reported in the manuscript
- Exact spelling, clearness of sentences and phrases, and appropriate reporting of quantities (proper units, important statistics) are just as significant in an abstract as they are anywhere else

Introduction:

The **Introduction** should "introduce" the manuscript. The reviewer should be presented with sufficient background information to be capable to comprehend and calculate the purpose of your study without having to submit to other works. The basis for the study should be offered. Give most important references but shun difficult to make a comprehensive appraisal of the topic. In the introduction, describe the problem visibly. If the problem is not acknowledged in a logical, reasonable way, the reviewer will have no attention in your result. Speak in common terms about techniques used to explain the problem, if needed, but do not present any particulars about the protocols here. Following approach can create a valuable beginning:

- Explain the value (significance) of the study
- Shield the model - why did you employ this particular system or method? What is its compensation? You strength remark on its appropriateness from a abstract point of vision as well as point out sensible reasons for using it.
- Present a justification. Status your particular theory (es) or aim(s), and describe the logic that led you to choose them.
- Very for a short time explain the tentative propose and how it skilled the declared objectives.

Approach:

- Use past tense except for when referring to recognized facts. After all, the manuscript will be submitted after the entire job is done.
- Sort out your thoughts; manufacture one key point with every section. If you make the four points listed above, you will need a least of four paragraphs.
- Present surroundings information only as desirable in order hold up a situation. The reviewer does not desire to read the whole thing you know about a topic.
- Shape the theory/purpose specifically - do not take a broad view.
- As always, give awareness to spelling, simplicity and correctness of sentences and phrases.

Procedures (Methods and Materials):

This part is supposed to be the easiest to carve if you have good skills. A sound written Procedures segment allows a capable scientist to replacement your results. Present precise information about your supplies. The suppliers and clarity of reagents can be helpful bits of information. Present methods in sequential order but linked methodologies can be grouped as a segment. Be concise when relating the protocols. Attempt for the least amount of information that would permit another capable scientist to spare your outcome but be cautious that vital information is integrated. The use of subheadings is suggested and ought to be synchronized with the results section. When a technique is used that has been well described in another object, mention the specific item describing a way but draw the basic principle while stating the situation. The purpose is to text all particular resources and broad procedures, so that another person may use some or all of the methods in one more study or referee the scientific value of your work. It is not to be a step by step report of the whole thing you did, nor is a methods section a set of orders.

Materials:

- Explain materials individually only if the study is so complex that it saves liberty this way.
- Embrace particular materials, and any tools or provisions that are not frequently found in laboratories.
- Do not take in frequently found.
- If use of a definite type of tools.
- Materials may be reported in a part section or else they may be recognized along with your measures.

Methods:

- Report the method (not particulars of each process that engaged the same methodology)
- Describe the method entirely
- To be succinct, present methods under headings dedicated to specific dealings or groups of measures
- Simplify - details how procedures were completed not how they were exclusively performed on a particular day.
- If well known procedures were used, account the procedure by name, possibly with reference, and that's all.

Approach:

- It is embarrassed or not possible to use vigorous voice when documenting methods with no using first person, which would focus the reviewer's interest on the researcher rather than the job. As a result when script up the methods most authors use third person passive voice.
- Use standard style in this and in every other part of the paper - avoid familiar lists, and use full sentences.

What to keep away from

- Resources and methods are not a set of information.
- Skip all descriptive information and surroundings - save it for the argument.
- Leave out information that is immaterial to a third party.

Results:

The principle of a results segment is to present and demonstrate your conclusion. Create this part a entirely objective details of the outcome, and save all understanding for the discussion.

The page length of this segment is set by the sum and types of data to be reported. Carry on to be to the point, by means of statistics and tables, if suitable, to present consequences most efficiently.You must obviously differentiate material that would usually be incorporated in a study editorial from any unprocessed data or additional appendix matter that would not be available. In fact, such matter should not be submitted at all except requested by the instructor.

Content

- Sum up your conclusion in text and demonstrate them, if suitable, with figures and tables.
- In manuscript, explain each of your consequences, point the reader to remarks that are most appropriate.
- Present a background, such as by describing the question that was addressed by creation an exacting study.
- Explain results of control experiments and comprise remarks that are not accessible in a prescribed figure or table, if appropriate.
- Examine your data, then prepare the analyzed (transformed) data in the form of a figure (graph), table, or in manuscript form.

What to stay away from

- Do not discuss or infer your outcome, report surroundings information, or try to explain anything.
- Not at all, take in raw data or intermediate calculations in a research manuscript.

- Do not present the similar data more than once.
- Manuscript should complement any figures or tables, not duplicate the identical information.
- Never confuse figures with tables - there is a difference.

Approach

- As forever, use past tense when you submit to your results, and put the whole thing in a reasonable order.
- Put figures and tables, appropriately numbered, in order at the end of the report
- If you desire, you may place your figures and tables properly within the text of your results part.

Figures and tables

- If you put figures and tables at the end of the details, make certain that they are visibly distinguished from any attach appendix materials, such as raw facts
- Despite of position, each figure must be numbered one after the other and complete with subtitle
- In spite of position, each table must be titled, numbered one after the other and complete with heading
- All figure and table must be adequately complete that it could situate on its own, divide from text

Discussion:

The Discussion is expected the trickiest segment to write and describe. A lot of papers submitted for journal are discarded based on problems with the Discussion. There is no head of state for how long a argument should be. Position your understanding of the outcome visibly to lead the reviewer through your conclusions, and then finish the paper with a summing up of the implication of the study. The purpose here is to offer an understanding of your results and hold up for all of your conclusions, using facts from your research and generally accepted information, if suitable. The implication of result should be visibly described. Infer your data in the conversation in suitable depth. This means that when you clarify an observable fact you must explain mechanisms that may account for the observation. If your results vary from your prospect, make clear why that may have happened. If your results agree, then explain the theory that the proof supported. It is never suitable to just state that the data approved with prospect, and let it drop at that.

- Make a decision if each premise is supported, discarded, or if you cannot make a conclusion with assurance. Do not just dismiss a study or part of a study as "uncertain."
- Research papers are not acknowledged if the work is imperfect. Draw what conclusions you can based upon the results that you have, and take care of the study as a finished work
- You may propose future guidelines, such as how the experiment might be personalized to accomplish a new idea.
- Give details all of your remarks as much as possible, focus on mechanisms.
- Make a decision if the tentative design sufficiently addressed the theory, and whether or not it was correctly restricted.
- Try to present substitute explanations if sensible alternatives be present.
- One research will not counter an overall question, so maintain the large picture in mind, where do you go next? The best studies unlock new avenues of study. What questions remain?
- Recommendations for detailed papers will offer supplementary suggestions.

Approach:

- When you refer to information, differentiate data generated by your own studies from available information
- Submit to work done by specific persons (including you) in past tense.
- Submit to generally acknowledged facts and main beliefs in present tense.

Administration Rules Listed Before Submitting Your Research Paper to Global Journals Inc. (US)

Please carefully note down following rules and regulation before submitting your Research Paper to Global Journals Inc. (US):

Segment Draft and Final Research Paper: You have to strictly follow the template of research paper. If it is not done your paper may get rejected.

- The **major constraint** is that you must independently make all content, tables, graphs, and facts that are offered in the paper. You must write each part of the paper wholly on your own. The Peer-reviewers need to identify your own perceptive of the concepts in your own terms. NEVER extract straight from any foundation, and never rephrase someone else's analysis.

- Do not give permission to anyone else to "PROOFREAD" your manuscript.

- Methods to avoid Plagiarism is applied by us on every paper, if found guilty, you will be blacklisted by all of our collaborated research groups, your institution will be informed for this and strict legal actions will be taken immediately.)
- To guard yourself and others from possible illegal use please do not permit anyone right to use to your paper and files.

CRITERION FOR GRADING A RESEARCH PAPER (COMPILATION) BY GLOBAL JOURNALS INC. (US)

Please note that following table is only a Grading of "Paper Compilation" and not on "Performed/Stated Research" whose grading solely depends on Individual Assigned Peer Reviewer and Editorial Board Member. These can be available only on request and after decision of Paper. This report will be the property of Global Journals Inc. (US).

Topics	Grades		
	A-B	**C-D**	**E-F**
Abstract	Clear and concise with appropriate content, Correct format. 200 words or below	Unclear summary and no specific data, Incorrect form Above 200 words	No specific data with ambiguous information Above 250 words
Introduction	Containing all background details with clear goal and appropriate details, flow specification, no grammar and spelling mistake, well organized sentence and paragraph, reference cited	Unclear and confusing data, appropriate format, grammar and spelling errors with unorganized matter	Out of place depth and content, hazy format
Methods and Procedures	Clear and to the point with well arranged paragraph, precision and accuracy of facts and figures, well organized subheads	Difficult to comprehend with embarrassed text, too much explanation but completed	Incorrect and unorganized structure with hazy meaning
Result	Well organized, Clear and specific, Correct units with precision, correct data, well structuring of paragraph, no grammar and spelling mistake	Complete and embarrassed text, difficult to comprehend	Irregular format with wrong facts and figures
Discussion	Well organized, meaningful specification, sound conclusion, logical and concise explanation, highly structured paragraph reference cited	Wordy, unclear conclusion, spurious	Conclusion is not cited, unorganized, difficult to comprehend
References	Complete and correct format, well organized	Beside the point, Incomplete	Wrong format and structuring

INDEX

save our planet

Global Journal of Computer Science and Technology

Visit us on the Web at www.GlobalJournals.org | www.ComputerResearch.org
or email us at helpdesk@globaljournals.org

ISSN 9754350