ZERONE

an annual technical journal published by the students of department of electronics and computer engineering, pulchowk campus

Volume 4•2062/2005

http://www.ioe.edu.np/zerone

Printed at

Malla Press

Chhauni, Tel: 4276544

few words...

The fourth volume of **ZERONE** is finally out. This issue is a combined product of all the students and faculty who supported us with their articles/ papers, the editorial and layout team whose dedication ensured the status quo quality of ZERONE was maintained and the marketing team without whose effort the publication would not have been possible.

This ZERONE team apologizes for the delay in publication of this issue of the journal. The ZERONE team has been through some difficult times during preparation of this issue. But the resilient nature of the team and support from all helped to make it through and finally the issue is out in your hands. The ZERONE team looks forward to your continued support.

This issue marks the introduction of several new categories in the ZERONE. For the first time, the ZERONE team conducted an interview with a department faculty and a practicing engineer Mr.Tika Upreti, and the excerpt of the interview is published in this issue. The ZERONE team believes such interviews allow young engineering students to get an idea of who an engineer is supposed to be. Similarly, following the norms of engineering journals, this issue includes and article on a recent senior level project, the Automated Engraver. As usual, issue features articles ranging from discussion on cutting edge technology like Quantum Computing, OLED to contemporary technologies like VSAT, RADAR, WiFi.

Technical journals are integral part of engineering practice. It provides a platform for presenting innovative ideas as well as analysis of existing technologies. ZERONE provides an opportunity for students to present their ideas research work on their field of interest. The articles from fresher to seniors, over a wide range of issues in ICT are a proof that the opportunity presented by ZERONE was accepted.

The ZERONE team would like to thank all the organizations and various bodies within the institute who supported us financially despite difficult times. Also the team acknowledges all the students who supported us with their articles and valuable suggestions without which the continuity of ZERONE would not have been possible.

The ZERONE team 2005 wishes good luck for the new team for the next issue.

ZERONE team wishes a Happy New Year 2063.

Table of Contents

Emerging Technologies

Contemporary Technologies

Network Technologies

Table of Contents

Mystery Demystified

Final Year Projects

Editorial Feature

Computer Operation & Programming

A 'Small' step for Technology, A Quantum Leap for Computers

Anjan Narsingh Rayamajhi
Ashay Thakur
Jeewan Shrestha
2060 Electronics

A hell of a computer with memory larger than the square of the square of the largest available memory in today's world and could manipulate, say googles of inputs simultaneously, is what we attempt to try and unveil today. Yes, a quantum computer could hold mountains of data and process 10^{100} and more inputs concurrently.

All of us who refuse to succumb to our urges to believe in quantum mechanics are in for a rude awakening. According to the well publicized Moore's law, the number of transistors in a computer doubles every 18 months. Itanium, do I hear? Well I-64 followed I-32 with double the number of transistors in 20 months; pretty close eh? Now that it is a norm that the doubling period is around that mark, the problem arises with the decreasing size of the microprocessors – as the microprocessors shrink we could see transistors the size of a hydrogen atom in the next eight to ten years. Even given the advent of new technologies, we only have up to 2020 before the transistors are so small that they will show the wave-particle duality. Hence the idea of Quantum computers is here to stay, to govern the way we shall communicate, manipulate not tomorrow, but the day after.

The primary reason that put general relativity on the map was its prediction of the bending of starlight by the sun, which in 1919 was confirmed by observation during a solar eclipse. That was the moment when general relativity emerged from the realm of theory and entered the empire of being a piece of reality as we know it. The time came for quantum computing to do the same in 2001 when in MIT a 7-qubit model was designed and Shor's algorithm implemented to factorize the number 15. Big deal! Well, the same algorithm could not have been performed using all supercomputers put together being assisted by each PC around the globe. I can say $5\times3=15$ though, right. But what are the prime factors of a number with $10^{1000000000000000000000000}$ digits? Only algorithms like the Shor's can do this.

Well, have we implanted too much technology already? We shall now look to explain the quantum lingo before we go ahead. The foremost in the process is Qubits.

Qubits

Quantum bits (Qubits) are essentially quaternary bits as opposed to the classical binary bits. A qubit can exist in states corresponding to a blend or superposition of the classical states or the binary system. In other words, a qubit can exist as a zero, a one, or simultaneously as both 0 and 1, with a numerical coefficient representing the probability for each state. Qubit exhibits quantum parallelism as, a qubit is at 0 in our universe, it has a value 1 in the parallel universe. A qubit is hence a set of two universes and the state it is in is determined by the reference taken. So, a computer with n qubits would simplify 2^n universes.

we only have up to 2020 before the transistors are so small that they will show the wave-particle duality

Superposition

The theory of quantum superposition states that any given particle which is unobserved and has more than one possible state, is simultaneously in all possible states until it is observed. In 1935, Erwin Schrödinger, the physicist who devised the central equation of quantum mechanics, described a thought experiment that is popular to as a Schrödinger's Cat experiment.

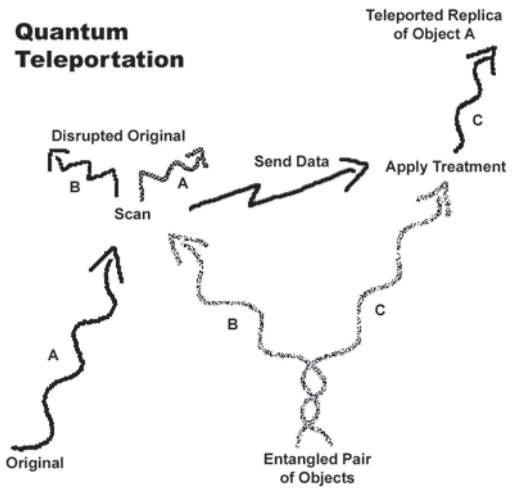
The probability that a cat, placed in sealed box with radioactive element having ½ chance of decaying in an hour, lives or dies cannot be deciphered until the box is opened. The cat, says quantum mechanics, is superposed between life and death and only when the box is opened is its fate decided.

Entanglement

Quantum Particles exhibit a remarkable property known as entanglement. A pair of quantum particles can exist in entangled 'superposition', a mixture of states that resolves only when some physical property such as spin or polarization is measured. Entanglement is the imbroglio of quantum particles, a change in property of one is reflected on the others even if they are separated by light years. A point in case has been cited below.

When two quantum systems are created while conserving some property, their state vectors are correlated, or entangled. For example, when two photons have spins of '1' and '-1'. When one photon's spin is measured to be '1', the other photon's spin of '-1' immediately becomes known too. There are no forces involved and no explanation of the mechanism.

Quantum entanglement is a fundamental requirement for Quantum Computing, but till now only entanglement between microscopic particles has been generated. Using new method of generating entanglement, an entangled state involving two macroscopic objects, each consisting of cesium gas sample containing about 1012 atoms, has now been created. The entangled spin state can survive for 0.5 milliseconds.



Teleportation

Teleportation is the feat of making an object or person disintegrate in one place while a perfect replica appears somewhere else. The general idea seems to be that the original object is scanned in such a way as to extract all the information from it, then this information is transmitted to the receiving location and used to construct the replica, not necessarily from the actual material of the original, but perhaps from atoms of the same kinds, arranged in exactly the same pattern as the original.

In 1993 an international group of six scientists, including IBM Fellow Charles H. Bennett showed that perfect teleportation is indeed possible in principle, but only if the original is destroyed. In subsequent years, other scientists have demonstrated teleportation experimentally in a variety of systems, including single photons, coherent light fields, nuclear spins, and trapped ions. Teleportation facilitates long range quantum communication (perhaps "quantum internet"), and making it much easier to build a working quantum computer.

Quantum Interference

Quantum interference is an expression of the wave behavior of particles (or indivisible energy quanta). Perhaps everybody knows that a strong "classical" coherent radiation interferes - i.e. its amplitudes are added in a destructive or constructive way. But even if the radiation is so dim that there is at most one photon with a high probability in the interferometer one can still observe an interference pattern. A contra-intuitive fact is that the single photon must go somehow through both the arms of the interferometer and interfere with itself. However, if one is able to determine which path the photon chose interference disappears. Interference is the essential component of many experiments carried out in our laboratory.

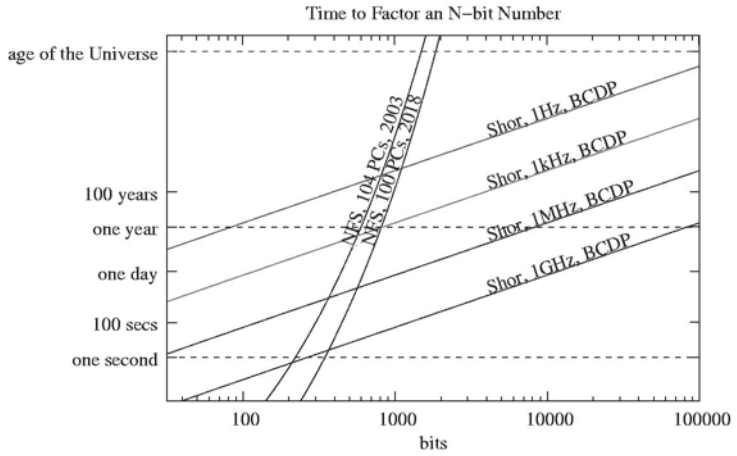
Error correction and decoherence are two of the best qualities that the quantum computers promise.

Quantum Algorithms

Quantum computation is aimed at utilizing the quantum nature of physical systems in order to efficiently solve computation problems which are deemed impossible in classical computers. Many algorithms have been formulated for the quantum computers which use the quantum superposition and entanglement principles, the operation of which can be generalized as shown:

- Runs are begun by creating a superposition of all possible input values.

- Executing a function gives a superposition of answers of all possible inputs! The hard part is extracting the answer we want.
- Every part of the superposition works independently on the algorithm.
- They all work by using interference. The phase of parts of the superposition are arranged to cancel out and leave only the interesting answer.



Among the various algorithms the benchmarks can be highlighted as:

Shor's Factorization Algorithms

P W Shor was able to utilize the quantum computers ability to obtain unprecedented parallelism by his algorithm for factorizing composite integers into prime factors in polynomial speed.

Using the Quantum Fourier Transform (QFT), factoring an L-bit number is $O(L^3)$ which provides super polynomial speed-up.

It is found that the time required for factoring N-bit number increases exponentially for the classical computers whereas the quantum computer using the Shor's algorithm has a linear rise as shown in the figure above.

Grover's search algorithm

L K Grover presented an iterative algorithm for searching unstructured list of N items with quadratic speed-up using quantum parallelism over algorithms run on classical computers. It is shown by Grover that by having the elements of the database in a coherent superposition of states, one can search an object in $O(\sqrt{N})$ quantum mechanical steps.

Grover's algorithm provides best efficiency for single match but the number of iteration increases as the matches increase which is undesired for a search algorithm since the problem is expected to be easier for multiple matches. This problem can be minimized by using the partial diffusion operation. Grover's algorithm finds application in the field of cryptography.

Deutsch-Jozsa Algorithm

David Deutsch and Richard Jozsa devised an algorithm to determine whether a function is constant or balanced. Deutsch-Jozsa algorithm implements the n bits $x_1, x_2, \ldots, x_n$ taken as inputs and $f(x_1, x_2 \ldots, x_n)$ as output such that the function is known to be constant if (returns 0 or 1 on all inputs) or balanced (returns 0 or 1 for two halves of the domain). In worst case conventional deterministic algorithm requires 2^{n-1} evaluations of $f()$ whereas the Deutsch-Jozsa algorithm requires just 1 evaluation of $f()$.

Astoundingly good data security can be achieved using these algorithms in Quantum Computers. Presently the backbone of quantum networking is Quantum Key Distribution.

Quantum Key Distribution

Discovered by Charles Bennett and his associates at IBM **[BB84]**, Quantum Key Distribution is a Protocol which is probably secure, by which private key bits can be created between two parties. The security of QKD is conditioned only on the laws of physics being correct! Since information is coded as a quantum state of a particle such as light polarization, electrons spin, etc… and according to the Heisenberg's Uncertainty Principle it is impossible to discover both the momentum and position of a particle at any given instant in time. Therefore basically an eavesdropper, an intruder, can't discover a cryptographic key based on particle state information; he/she would need the actual particle to decipher any data encrypted with the key.

The system is able to generate photons using lasers, and send them in one of two modes, vertical/horizontal or +45°/-45° such that within each mode one orientation represent value 0 and other 1. The sender, whom cryptographers term *Alice*, sends each photon with randomly chosen mode

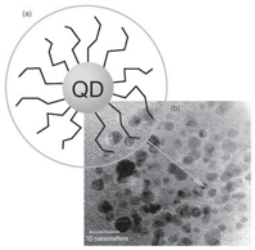
and value. The receiver called *Bob* randomly chooses between the two modes when he tries to detect a photon. Only the matching modes on both sections determine the correctness of the received photon. Alice uses classical communication channel to tell Bob about the modes she selected for each photon. Bob then replies with the modes of those photons he measured correctly so that Alice can discard the ones with false measurement. Hence the correct measurement constitutes the encryption key that Alice and Bob now share.

If someone referred as *Eve* tries to eavesdrop on the transmission, she will not be able to 'read' it without altering it. The photons is destroyed as soon as it is converted to electrical signals, so Eve must generate new signals to send to Bob. Thus she must guess, causing error in the secured key linked between Alice and Bob, hence they discard the key opting to eavesdropper. This provides better security than conventional Data Encryption Standard or the newer Advanced Encryption Standard.

Some breakthrough applications in the field of Quantum Computing are enumerated below.

Nuclear Magnetic Resonance (NMR) Quantum Computers

NMR refers to spectroscopic studies of transition between the Zeeman levels of an atomic nucleus in magnetic field. Researchers at IBM, MIT, Berkeley and Oxford have demonstrated powerful quantum search algorithm using NMR computers. Initially NMR had a slow development due to the fact that low signals of NMR need to be amplified by using several copies from the large number of molecules in the solution but it was difficult to obtain the correct results since all copies may not start calculating at the same instant in time. However in 1997 two separate solutions were published both illustrating on how to 'distill' an effectively pure starting state from a complex mixture. Chuang and co-workers, founders of one of the solution, have implemented the Grover's Search Algorithm for two Qubits on their ^{1}H and ^{13}C nuclei isotopically labelled chloroform computer. The next step is to implement complex algorithm on the NMR computer in future but may be difficult since the efficiency of distillation of pure initial solutions decreases rapidly with increase in number of Qubits.



Quantum Dots

Quantum Dots are the nanometric devices containing tiny droplet of free electrons such that their properties depend on the number of electrons. Quantum Dots become quantized due to the confined electrons. Quantum Dots are interconnected using quantum wires which are molecular tethers made of organic compounds. At Lawrence Livermore National Laboratory, scientists have been able to make Silicon and Germanium Quantum Dots that emit light throughout the visible spectrum and have ability to tune their luminescence to any wavelength over spectral range with stability. Quantum Dots LED, particularly those that provide the hard to reach blue end of the spectrum, appear to be a key to opening number of exciting technological advances in the fields of full-colour, flat panel displays; ultra high density optical memories and data storage; backlighting; and chemical and biological sensing.

Quadruple Ion Trap

The quadruple Ion Trap functions both as an ion store and as a mass spectrometer. As a storage device the Ion Trap acts as an 'electric field test tube' for the confinement of ions positively or negatively charged in the absence of solvent due to the formation of a trapping 'potential well' when appropriate potentials are applied to the electrodes. The confinement of ions permits the study of gas-phase ion chemistry and elucidation of ion structures.

Finale

It is entirely upon us to either accept or reject this fast approaching tomorrow. Quantum computing has been lurking round the corner ever since Babbage piped it to the pole. For over half a century now the concept of QC has been trying to outdo conventional computing. But the romanticism has ended now. We are now looking at the concept banging our wits out, trying to out pace the development of simple microprocessors.

The time has come now that the advances in the field be secured and we try and make tomorrow brighter using QC.

At this very moment obstacles are being surmounted that will provide the knowledge needed to thrust quantum computers up to their rightful position as the fastest computational machines in existence. Error correction has made promising progress to

date, nearing a point now where we may have the tools required to build a computer robust enough to adequately withstand the effects of decoherence. Quantum hardware, on the other hand, remains an emerging field, but the work done thus far suggests that it will only be a matter of time before we have devices large enough to test Shor's and other quantum algorithms. Thereby, quantum computers will emerge as the superior computational devices at the very least, and perhaps one day make today's modern computer obsolete. Quantum computation has its origins in highly specialized fields of theoretical physics, but its future undoubtedly lies in the profound effect it will have on the lives of all mankind. ■

References

1. Rod Van Meter, *Open University Classroom on Quantum Computing*
2. Ajoy Ghatak, *An Introduction to Quantum Mechanics*
3. http://www.qubit.org/
4. http://quantum.fis.ucm.es/
5. *Articles on Quantum Dot, NMR, Ion Traps, QKD*

During the heat of the space race in the 1960's, NASA decided it needed a ball point pen to write in the zero gravity confines of its space capsules. After considerable research and development, the Astronaut Pen was developed at a cost of $1 million. The pen worked and also enjoyed some modest success as a novelty item back here on earth.

The Soviet Union, faced with the same problem, used a pencil.

The future of display technology:

OLED

Anjan Narsingh Rayamajhi
2060 Electronics

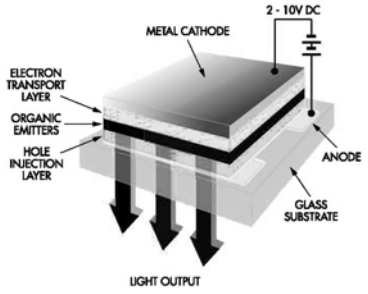
Today every sci-fi movie character is equipped with the thin, high resolution, pocket sized rollup monitor that perhaps may look real; but the reality is not far. In the near future we are sure to be bloomed with the blessings of such rollup TV, colour changeable cloths, wallpaper-like panels that curl around the architectural column and many more, not just by the grace of god but by the use of an electronic display device called the **Organic Light Emitting Diode** or **OLED.**

What is an OLED?

Well, OLED is organic and it emits light. An OLED is a monolithic, solid-state device that typically consists of a series of organic materials sandwiched between two thin-film electrodes. The choice of organic materials and the layer structure determine the device's performance features such as color emitted, operating lifetime and power efficiency. It operates by the principle of *electrophosphorescence* defined as the emission of light from a substance exposed to electric potential and persisting as an afterglow after the exciting potential has been removed; the extra energy is stored in metastable states and re-emitted later.

What does it look like?

It looks like *a traditional club sandwich with a pair of bread slices on top with ham and pastrami layers in the middle*. The actual structure is shown in the figure. OLED consists of a pair of electrode films squeezing in organic films of conductive (e.g. polyaniline.) and emissive (e.g. polyfluorene.) layers. The layers are mounted on a substrate made up of plastic or glass with the CMOS control circuitry drawn on it. The deposition of the organic layer on the substrate is done by i) *Vacuum Thermal Evaporation*, ii) *Organic Vapour Phase Deposition*, or iii) *Inkjet Printing processes.*



Structure of OLEDs

How do they work?

They work by making electrons emit light as they recombine with holes. Conduction electrons in semiconductors are more energetic than the valence ones. On providing sufficient energy valence electrons can jump to conducting layer and they emit luminous energy on recombining with the holes. For the production of visible light, organic materials should have an energy gap in a relatively small range, about two to three electron volts. Due to the electric potential applied across the layers, cathode gives electrons to the emissive layer and anode removes electrons from (or gives holes to) the conductive layer of organic molecules. At the boundary between the layers, electrons meet holes; as a result they fall in to an energy level of atom that is missing an electron thereby emitting photon of light as energy. The colour depends on the organic material; the potential applied and current determine the brightness or intensity. Doping or enhancing organic material helps control brightness and color of light. And manufacturers use a variety of organic materials to ensure different colours.

In OLED electrical energy injected onto a host molecule is often transferred to luminescent "guest" molecules which then light up. When an electron combines with a hole, an energetic, chargeless and free to move unit is formed called Exciton. The quantum-mechanical rules of interaction of the electron and hole spins dictate that for most materials only one in four excitons formed will be able to give up its energy as a photon, whereas the others will lose their energy as heat, giving only 25% efficiency. But the problem was overcome in 1998 by developing an OLED with emitting layer made up of heavy metal such as Platinum or Iridium.

Since the outer electrons of the heavy metals have relatively larger angular momentum of rotation, this momentum interacts with the spin of other electrons, essentially creating conditions where all the excitons can emit light rather than heat, raising the theoretical efficiency to nearly 100 percent. These new emitters are referred to as *phosphorescent OLEDs (PHOLED)*, to distinguish them from the more common fluorescent OLEDs.

Are there any types?

Yup! A lot, actually.

On the basis of the materials used OLEDs are categorized as:

Small molecules OLED

Invented in 1987, the p-n type OLED consisted of film of very good electron and hole conductors such that the emission of the light occurred at the junction of the layers. Such small molecules OLED merits in brighter light emission especially red, blue and green but the manufacturing process is expensive.

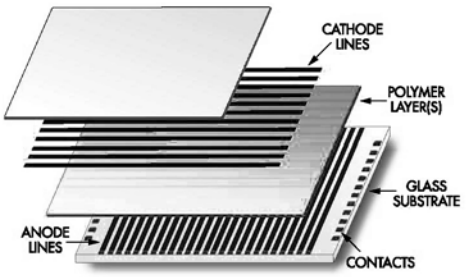
Polymer type OLED (PLED)

Reported in 1990 by Burroughes and his colleagues at the University of Cambridge, PLED is formed by the chain-like unions of a large number of smaller organic molecules. They are manufactured more economically and have low voltage requirement due to higher conductivity causing better efficiency. In addition, due to the stiffness of polymers the electron and hole emitters are strongly held, hence large flat panels are easily manufactured.

On the basis of the technique used for driving current the OLEDs are divided as:

Passive matrix OLEDS (PMOLEDs)

In PMOLEDs the conductors are arranged orthogonally in rows and columns forming a matrix such that every row-column intersection is a pixel. On supplying current to the n^{th} row and m^{th} column the [n, m] pixel can be activated to emit light with brightness depended on current supplied. Thus they consume more power and are suitable for smaller displays including cell phones, MP3 players and portable games.



Structure of POMLEDs

Active-matrix OLEDs (AMOLEDs)

AMOLEDs use transistor switches to control the current in each pixel, hence controlling its brightness. The organic layer is sandwiched between cathode and anode along with the thin film transistors (TFT) backplane. For each pixel, one TFT controls the start and stop of the storage capacitor charging period and other provides a source voltage for the constant current to the pixel. Therefore the pixel remains *on* for all times. Polysilicon and Amorphous Silicon TFT Backplane Technology are used for the production of TFTs and Organic TFTs are also on their way. The AMOLEDs consume less power making them applicable for large panel displays.

we are sure to be bloomed with the blessings of rollup TVs, colour changeable cloths, wallpaper-like panels that curl around the architectural column and many more

In addition, other types of OLEDs fill the spectrum:

- Transparent OLEDs have all the components 'transparent', useful for heads-up displays.
- Top-Emitting OLEDS have opaque or reflective substrate, applicable for smart cards.
- Foldable OLEDs have substrate made up of flexible metallic foils or plastic. They can be sewn into clothing making 'smart' cloths with integrated computer, GPS receiver, cell phone etc…
- White OLEDs which emit more energy efficient, uniform and brighter white lights that can be used in daily lighting.

How good is it?

It is better than the Liquid Crystal Displays which are inorganic and non luminous. They block/allow light reflected from backlight system. Such backlights consume half the

power making the system less efficient.

OLEDs do not need such backlighting, they are self emissive.

Comparing with the Cathode Ray Tubes, CRTs operate by striking electrically accelerated electron beam on phosphorescent screen causing a bright spot at the point of impact. This technology also needs more energy, hence less efficient and the tube is bulky.

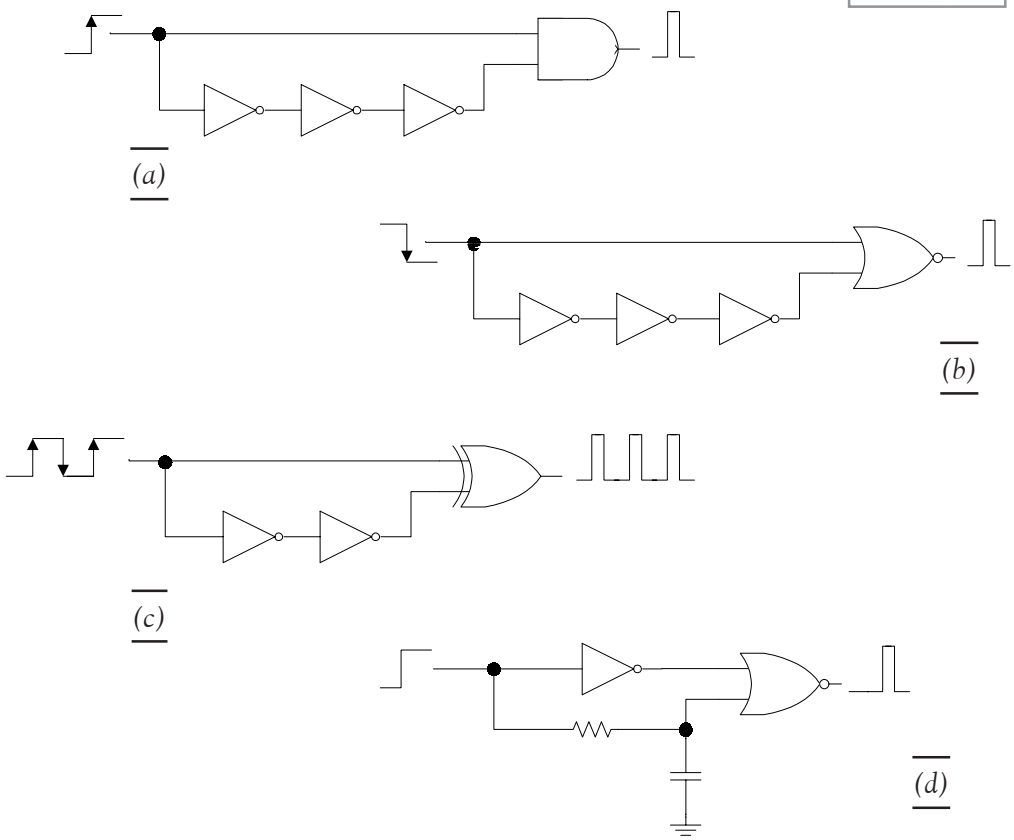
In addition OLEDs have large angle of view, easily perceivable, each pixel glows like light bulb hence extra bright, high clarity, the pixel turn on/off quickly forming life-like picture quality, portable and energy efficient etc.

The blah…blah so far! Just not enough for the OLEDs to come and certainly not enough for the applicability they ensure. I truly hope that the above text introduced the subject well because tomorrow we are sure to be wearing cloths that make us invisible simply by projecting back to front and the front to back; our cloths being the screen. ■

References

1. www.howstuffworks.com
2. www.kodak.com
3. www.oled-display.net
4. www.universaldisplay.com
5. *Wave Report*
6. *Today's chemist at work articles*

circuit ideas



edge sensitive circuits: rising edge detector (a), falling edge detector (b), bi-edge detector (c), & rising edge detector with delay (d) (Source: AoE)

Printing Objects

Shristi N Pradhan
2062 Electronics

Nearly everyone is familiar with printers that print documents, but how about printers that actually make things? As this ScienCentral News report explains, such printers are here now.

Layer by Layer

At first glance, it looks like a desktop printer. Looking through the clear plastic case, one sees the print head moving back and forth. The screen on the computer monitoring the printer says "Printing: 82.5% complete." But, this is no ordinary printer. It's designed to accommodate a special kind of "ink" that could be anything from metal to organic materials. What comes out of the printer, is not a drawing or picture, it is a real object; in this case an electronic circuit.

If that sounds like science fiction, then you should know that such printers are in use today. "We're talking about devices that could actually fit on a table that could be used to create, for example, a circuit or even a small (computer) display," says John Batterton, CEO and President of Dimatix of Santa Clara, California, which is marketing such a printer. He explains, "That's sort of the first building block that's going to be available in the industry now. Once you get past the actual circuit, now you can start jetting (printing) things like transistors and things that are actually used to create the display, itself."

Dimatix is one of several groups exploring how printing technology could be adapted to make electronics. For example, Motorola is exploring printing electronics using high-grade commercial printing presses, while universities including Cornell and MIT are also trying to bring fabrication to the desktop. Cornell's Hod Lipson is excited about the new opportunities opened up by devices like the one they're using. Lipson says, "You can design anything you want, without worrying about how it's going to be fabricated. And that would really free up the ability to design things."

Key to this new style of electronics manufacturing is the fluids that act as the ink. These fluids are engineered at the molecular level to be applied like ink, but once applied, do things such as conduct electricity. What happens in these printers is that they first put down one layer of material. And then another layer of material on top of that, and you continue to build up layers of material until you've actually created the object that you want.

what comes out of the printer, is not a drawing or picture, it is a real object; in this case an electronic circuit.

While it's possible today to print simple circuits, there's a big difference between that and the next step of printing circuits and the electronic devices in them. The biggest obstacle in place for the industry today is the actual creation of these new fluids that have a function. So, it's the fluid to make a transistor, the fluid to make a display.

If that can be done, not only can electronics possibly be made more cheaply, but, Electronics suddenly can become far more customizable, such that custom circuits, or custom displays, can be made for a variety of applications.

An additional benefit is that electronics would no longer have to sit in rigid boxes that protect inflexible, even brittle, electronics circuits. With bendable electronics, you have the ability to create things like displays that can be put in places that you never thought before. So, for example at an airport you could have a display that wraps around a round post and is providing advertising in a constant new way… Or it could be a new kind of television set that's actually like a window shade. You just simply roll it out when you want to look at it and roll it back up into the shade.

So, while this new application of nanotechnology is still just in it's infancy, this new style of printing electronics just may be news that is "hot off the presses" that could forever change the way electronics are made. ■

Bluetooth in Brief

Anup Bajracharya
2061 Computer

Bluetooth is an industrial specification for wireless personal area networks (PANs), developed by a group of electronics manufacturers that provides a way to connect and exchange information between devices like personal digital assistants (PDAs), mobile phones, laptops, PCs, printers or digital cameras via a secure, low-cost, globally available short range radio frequency. Bluetooth allows these devices to make their own connections, without wires, cables or any direct action from a user.

The technology of Bluetooth centers around a small microchip, which functions as a low cost and short range radio link. The Bluetooth radio chip functions at 2.45 GHz, which is in the license-free ISM (Industrial Scientific Medical) band. In order to avoid interfering with other protocols which use the same frequency band, the Bluetooth protocol divides the band into 79 channels (each 1 MHz wide) and changes channels up to 1600 times per second.

Bluetooth lets devices 'talk' to each other when they come in range, even if they are not in the same room, as long as they are within up to 100 metres (328 feet) of each other, dependent on the power class of the product. Products are available in one of the three power classes:

- **Class 3 (1 mW)** is the rarest and allows transmission of 10 centimetres (3.9 inches), with a maximum of 1 metre (3.2 feet)
- **Class 2 (2.5 mW)** is most common and allows a quoted transmission distance of 10 metres (32 ft)
- **Class 1 (100 mW)** has the longest range at up to 100 metres. This class of product is readily available.

Network Arrangements

A Piconet is a network of devices connected in an ad hoc fashion, that is, not requiring predefinition and planning, as with a standard network. Two to eight devices can be networked into a piconet. It is a peer network, that is, once connected, each device has equal access to the others. However, one device is defined as master, and the others as slaves.

A Bluetooth master device can communicate with up to 7 slave devices. At any given time, data can be transferred between the master and one slave; but the master switches rapidly from slave to slave in a round-robin fashion. (Simultaneous transmission from the master to multiple slaves is possible, but not used much in practice). Either device may switch the master/slave role at any time.

The network arrangements can be either point-to-point or point-to-multipoint. The Bluetooth specification also allows connecting two or more piconets together to form a Scatternet, with some devices acting as a bridge by simultaneously being a master one piconet and a slave in another piconet.

Setting up Connections

Any Bluetooth device will transmit the following sets of information on demand:

- Device Name
- Device Class
- List of services
- Technical information eg: device features, manufacturer, Bluetooth specification, clock offset, etc.

Any device may perform an "inquiry" to find other devices to which to connect, and any device can be configured to respond to such inquiries. However, if the device trying to connect knows the address of the device, it will always respond to direct connection requests and will transmit the information shown in the list above, if requested for it. Use of the device's services however, may require pairing or its owner's permission, but the connection itself can be started by any device and be held until it goes out of range. Some devices can only be connected to one device at a time. If they're already connected, they will be prevented from connecting to other devices and showing up in inquiries until they disconnect the other device.

Every device has a unique 48-bit address. However, these addresses are generally not shown in inquiries and instead friendly "Bluetooth names" are used which can be set by the user. These names will appear when another user scans for devices, and in lists of paired devices. Most phones have the Bluetooth name set to the manufacturer and

model of the phone by default. Most phones and laptops will only show the Bluetooth names and special programs are required to get additional information about remote devices.

Every device also has a 24-bit class identifier. This provides information on what kind of a device it is (Phone, Smartphone, Computer, Headset, etc), which will also be transmitted when other devices perform an inquiry. On some phones, this information is translated into a little icon displayed beside the device's name.

Bluetooth devices will also transmit a list of services if requested by another device; this also includes some extra information such as the name of the service and what channel it is on. These channels are virtual and have nothing to do with the frequency of the transmission.

Transmission Types and Speeds

Data can be transmitted both synchronously and asynchronously. The Synchronous Connection Oriented (SCO) method is used primarily for voice, and Asynchronous Connectionless (ACL) is primarily for data. Within a piconet, each master-slave pair can use a different transmission mode, and modes can be changed at any time. Because of the need for smoothness in data transmission, SCO packets are generally delivered via reserved intervals, that is, the packets are sent in groups without allowing other transmissions to interrupt. As noted earlier, a different hop signal is used for each packet. SCO packets can be transmitted without polling by the sending unit. ACL links support both symmetric and asymmetric transmissions.

Bandwidth is controlled by the master unit, which determines how much of the total each slave unit can use. Slaves cannot transmit data until they have been polled by the master, and the master can broadcast messages to the slave units via ACL link

Circuit switching can be either synchronous or asynchronous. Up to three synchronous (voice) data channels, or one synchronous and one asynchronous data channel, can be supported on a single channel. Each synchronous channel can support a 64 kbps transfer rate, which is fully adequate for several human voice conversations. An asynchronous channel (which is needed when connecting to a computer printer, for example) can transmit as much as 721 kbps in one direction and 57.6 kbps in the opposite direction. If the use calls for the same speed in both directions, an asynchronous connection with 432.6 kbps capacity in both directions can be made. The later version of Bluetooth, Version 2.0 implementations feature Bluetooth Enhanced Data Rate (EDR), which can support transmission rates of upto 2.1 Mbit/s. Technically version 2.0 devices have a higher power consumption, but the three times faster rate reduces the transmission times, effectively reducing consumption to half that of 1.x devices

Avoiding Interference

every Bluetooth transmitter uses spread-spectrum transmitting automatically, it's unlikely that two transmitters will be on the same frequency at the same time

With many different Bluetooth devices in a room, you might think they'd interfere with one another, but let's see why there aren't any problems:

Low Power

One of the ways Bluetooth devices avoid interfering with other systems is by sending out very weak signals of 1 milliwatt. By comparison, the most powerful cell phones can transmit a signal of 3 watts. The low power limits the range of a normal Bluetooth device to about 10 meters, cutting the chances of interference between your computer system and your portable telephone or television. Even with the low power, the walls in your house won't stop a Bluetooth signal, making the standard useful for controlling several devices in different rooms.

Hopping

Even if several Bluetooth-capable devices are in the same room, it is unlikely that multiple devices will be on the same frequency at the same time, because Bluetooth uses a technique called **spread-spectrum frequency hopping**. In this technique, a device will use 79 individual, randomly chosen frequencies within a designated range, changing from one to another on a regular basis. In the case of Bluetooth, the transmitters change frequencies 1,600 times every second, meaning that more devices can make full use of a limited slice of the radio spectrum. Since every Bluetooth transmitter uses spread-spectrum transmitting automatically, it's unlikely that two transmitters will be on the same frequency at the same time. This same technique minimizes the risk that portable phones or baby monitors will disrupt Bluetooth devices, since any interference on a particular frequency will last only a tiny fraction of a second.

When Bluetooth-capable devices come within range of one another, an electronic conversation takes place to determine whether they have data to share or whether one needs to control the other. The user doesn't have to press a button or give a command — the electronic conversation happens automatically. Once the conversation has occurred, the devices — whether they're part of a computer system or a stereo — form a network. Bluetooth systems create a personal-area network (PAN), or **piconet**, that may fill a room or may encompass no more distance than that between the cell phone on a belt-clip and the headset on your head. Once a piconet is established, the members randomly hop frequencies in unison so they stay in touch with one another and avoid other piconets that may be operating in the same room.

A Networks Example

Let's take a look at how the Bluetooth frequency hopping and personal-area network keep systems from becoming confused. Let's say you've got a typical modern living room with the typical modern stuff inside. There's an entertainment system with a stereo, a DVD player, a satellite TV receiver and a television; there's a cordless telephone and a personal computer. Each of these systems uses Bluetooth, and each forms its own piconet to talk between main unit and peripheral.

The cordless telephone has one Bluetooth transmitter in the base and another in the handset. The manufacturer has programmed each unit with an address that falls into a range of addresses it has established for a particular type of device. When the base is first turned on, it sends radio signals asking for a response from any units with an address in a particular range. Since the handset has an address in the range, it responds, and a tiny network is formed. Now, even if one of these devices should receive a signal from another system, it will ignore it since it's not from within the network. The computer and entertainment system go through similar routines, establishing networks among addresses in ranges established by manufacturers. Once the networks are established, the systems begin talking among themselves. Each piconet hops randomly through the available frequencies, so all of the piconets are completely separated from one another.

Now the living room has three separate networks established, each one made up of devices that know the address of transmitters it should listen to and the address of receivers it should talk to. Since each network is changing the frequency of its operation thousands of times a second, it's unlikely that any two networks will be on the same frequency at the same time. If it turns out that they are, then the resulting confusion will only cover a tiny fraction of a second, and software designed to correct for such errors weeds out the confusing information and gets on with the network's business.

Future Bluetooth uses

One of the ways Bluetooth technology may become useful is in Voice over IP. When VoIP becomes more widespread, companies may find it unnecessary to employ telephones physically similar to today's analogue telephone hardware. Bluetooth may then end up being used for communication between a cordless phone and a computer listening for VoIP and with an infrared PCI card acting as a base for the cordless phone. The cordless phone would then just require a cradle for charging. Bluetooth would naturally be used here to allow the cordless phone to remain operational for a reasonably long period. In May 2005, the Bluetooth Special Interest Group (SIG) announced its intent to work with UWB manufacturers to develop a next-generation Bluetooth technology using UWB technology and delivering UWB speeds. This will enable Bluetooth technology to be used to deliver high speed network data exchange rates required for wireless VoIP, music and video applications.

Bluetooth may also be used for remote sales technology, allowing wireless access to vending machines and other commercial enterprises. ■

Face recognition

An Eigenface approach

Mahesh Subedi
2058 Computer

Introduction

The detection of faces and facial features from an arbitrary uncontrived image is a critical precursor to recognition. A robust scheme is needed to detect the face as well as determine its precise placement to extract the relevant data from an input image. This is necessary to properly prepare the image's 2D intensity description of the face for input to a recognition system. This detection scheme must operate flexibly and reliably regardless of lighting conditions, background clutter in the image, multiple faces in the image, as well as variations in face position, scale, pose and expression. The geometrical information about each face in the image that we gather at this stage will be used to apply geometrical transformations that will map the data in the image into an invariant form. By isolating each face, transforming it into a standard frontal mug shot pose and correcting lighting effects, we limit the variance in its intensity image description to the true physical shape and texture of the face itself.

Over the last ten years, face recognition has become a popular area of research in computer vision and one of the most successful applications of image analysis and understanding. Because of the nature of the problem, not only computer science researchers are interested in it, but neuroscientists and psychologists also. It is the general opinion that advances in computer vision research will provide useful insights to neuroscientists and psychologists into how human brain works, and vice versa.

it's a high-level pattern recognition problem in which humans are very adept, whereas it can be quite challenging to teach a machine to do it.

A general statement of the face recognition problem (in computer vision) can be formulated as follows: Given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces.

Research directions

- Recognition from outdoor facial images.
- Recognition from non-frontal facial images.
- Recognition at low false accept/alarm rates.
- Understanding why males are easier to recognize than females.
- Greater understanding of the effects of demographic factors on performance.
- Development of better statistical methods for understanding performance.
- Develop improved models for predicting identification performance on very large galleries.
- Effect of algorithm and system training on covariate performance.
- Integration of morphable models into face recognition performance.

History

Research efforts towards of faces recognition start in 1878 by English scientist **Sir Francis Galton**. His research involved the combination of people's photos, by means of superimposing face images. **Galton** proposed the photos alignment from human faces, depending on its main characteristic, putting some of them upon the other. The main difficulty was to describe the personal similarities, the types of faces and personal characteristics. To overcome this difficulty, biometrics characteristic were extracted from the face image to compare with the measures of another face in order to succeed in face recognition.

Biometrics and Face Recognition

Face Recognition is part of a larger context called Biometrics that give us the notion of life measure. Biometrics can be defined as some characteristics

that can be used to verify an individual's identity. The biometrics system is essentially a pattern recognition system that makes a personal identification determining the authenticity of some characteristic of an individual. In fact, an automatic people identification system based exclusively on fingerprints or on face recognition does not satisfy all the functionality requirements. Face Recognition is natural and non-intrusive, but it is not trustworthy as compared to the fingerprint verification which is relatively more trustworthy but can be intrusive and can cause resistance to the users, depending on the application.

Why Face Recognition is Interesting?

Face recognition is interesting to study because it is an application area where computer vision research is being utilized in both military and commercial products. Much effort has been spent on this problem, yet there is still plenty of work to be done.

Basic research related to this field is currently active. For example, research searching for a fundamental theory describing how light and objects interact to produce images was recently published in April 2004. Often, practical applications can grow out of improvements in theoretical understanding and it seems that this problem will continue to demonstrate this growth.

Personally, I'm interested in this project because it's a high-level pattern recognition problem in which humans are very adept, whereas it can be quite challenging to teach a machine to do it. The intermediate and final visual results are interesting to observe in order to understand failures and successes of the various approaches.

How Computers See

Although we are able to visually recognize complex objects from an early age, visual recognition is very difficult to automate. A single object may be viewed from a number of different angles, in different lighting conditions, and with other objects partially obscuring view of the object.

Purpose of a vision system: To move from an initial digitized image to useful analysis of the scene. Goal is to recognize objects in a scene, given one or more images of that scene.

Ultimate goal: Develop a system with capabilities comparable to human capabilities.

Stages of the vision process

Begin with a digitized image, which gives the brightness at each point (pixel) in the image. The stages that follow:

Low-level Processing

Simple features are identified for example, lines or edges in the image. Output is something like a line drawing of the objects in the image, the lines separating the image into regions corresponding to object surfaces.

Medium-level Processing

Next, determine how far away the regions are and what their orientation is. Output referred to as "2 -D sketch."

High-level Processing

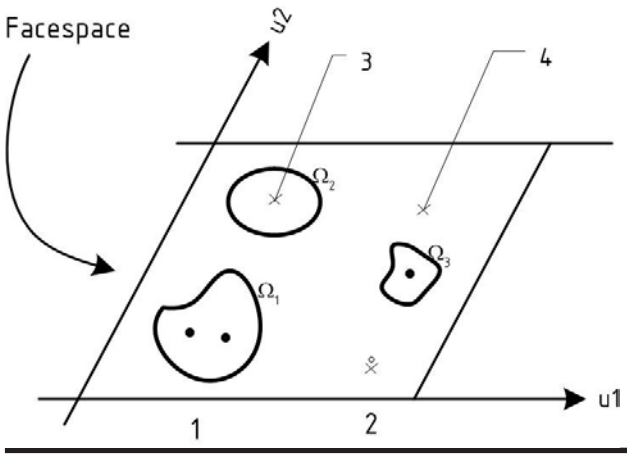
Obtain a useful high level description/representation of the scene. Work out the 3-D models of objects in the scene given the depth and orientation information above. Next, attempt is to recognize what sorts of objects occur in the scene.

Face Recognition Using Eigenfaces

Introduction

The information theory approach of encoding and decoding face images extracts the relevant information in a face image, encode it as efficiently as possible and compare it with database of similarly encoded faces. The encoding is done using features which may be different or independent than the distinctly perceived features like eyes, ears, nose, lips, and hair.

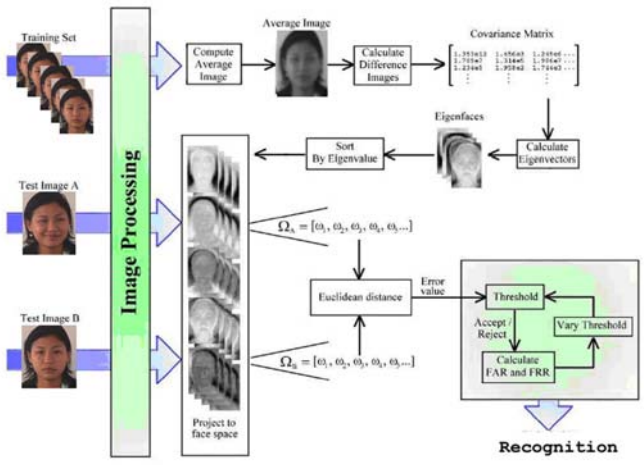
Mathematically, principal component analysis approach will treat every image of the training set as a vector in a very high dimensional space. The eigenvectors of the covariance matrix of these vectors would incorporate the variation amongst the face images. Now each image in the training set would have its contribution to the eigenvectors (variations). This can be displayed as an '*eigenface*' representing its contribution in the variation between the images. In each eigenface some sort of facial variation can be seen which deviates from the original image.



	Face Space	Known face class	Result
1	near	near	Recognized as Ω_1
2	near	far	Who are you ?
3	far	near	?False positive?
4	far	far	No face

Figure: a simplified version of face space to illustrate the four results of projecting an image into face space. In this case, there are two eigenfaces (u_1 and u_2) and three known individuals (Ω_1, Ω_2, and Ω_3).

The high dimensional space with all the eigenfaces is called the image space (feature space). Also, each image is actually a linear combination of the eigenfaces. The amount of overall variation that one eigenface counts for, is actually known by the eigenvalue associated with the corresponding eigenvector. If the eigenface with small eigenvalues are neglected, the image can be a linear combination of reduced number of these eigenfaces. For example, if there are M images in the training set, we would get M eigenfaces. Out of these, only M' eigenfaces are selected such that they are associated with the largest eigenvalues. These would span the M'-dimensional subspace 'face space' out of all the possible images (image space).



Overall Block diagram of Eigen Face Approach of Face Recognition

When the face image to be recognized (known or unknown), is projected on this face space (figure 1), you can get the weights associated with the eigenfaces, that linearly approximate the face or can be used to reconstruct the face. Now these weights are compared with the weights of the known face images so that it can be recognized as a known face in used in the training set. In simpler words, the Euclidean distance between the image projection and known projections is calculated; the face image is then classified as one of the faces with minimum Euclidean distance.

How Eigenface algorithm works?

The task of facial recognition is discriminating input signals (image data) into several classes (persons). The input signals are highly noisy (e.g. the noise is caused by differing lighting conditions, pose etc.), yet the input images are not completely random and in spite of their differences there are patterns which occur in any input signal. Such patterns, which can be observed in all signals, could be - in the domain of facial recognition - the presence of some objects (eyes, nose, mouth) in any face as well as relative distances between these objects. These characteristic features are called *eigenfaces* in the facial recognition domain (or *principal components* generally). They can be extracted out of original image data by means of a mathematical tool called *Principal Component Analysis* (PCA).

By means of PCA one can transform each original image of the training set into a corresponding eigenface. An important feature of PCA is that

one can reconstruct any original image from the training set by combining the eigenfaces. Remember that eigenfaces are nothing less than characteristic features of the faces. Therefore one could say that the original face image can be reconstructed from eigenfaces if one adds up all the eigenfaces (features) in the right proportion. Each eigenface represents only certain features of the face, which may or may not be present in the original image. If the feature is present in the original image to a higher degree, the share of the corresponding eigenface in the 'sum' of the eigenfaces should be greater. If, contrary, the particular feature is not (or almost not) present in the original image, then the corresponding eigenface should contribute a smaller (or not at all) part to the sum of eigenfaces. So, in order to reconstruct the original image from the eigenfaces, one has to build a kind of weighted sum of all eigenfaces. That is, the reconstructed original image is equal to a sum of all eigenfaces, with each eigenface having a certain weight. This weight specifies, to what degree the specific feature (eigenface) is present in the original image. If one uses all the eigenfaces extracted from original images, one can reconstruct the original images from the eigenfaces *exactly*. But one can also use only a part of the eigenfaces. Then the reconstructed image is an approximation of the original image. However, one can ensure that losses due to omitting some of the eigenfaces can be minimized. This happens by choosing only the most important features (eigenfaces). Omission of eigenfaces is necessary due to scarcity of computational resources.

How does this relate to facial recognition? The clue is that it is possible not only to extract the face from eigenfaces given a set of weights, but also to go the opposite way. This opposite way would be to extract the weights from eigenfaces and the face to be recognized. These weights tell nothing less, as the amount by which the face in question differs from "typical" faces represented by the eigenfaces. Therefore, using this weights one can determine two important things:

1. Determine if the image in question is a face at all. In the case the weights of the image differ too much from the weights of face images (i.e. images, from which we know for sure that they are faces), the image probably is not a face.
2. Similar faces (images) possess similar features (eigenfaces) to similar degrees (weights). If one extracts weights from all the images available, the images could be grouped to clusters. That is, all images having similar weights are likely to be similar faces.

Calculation of Eigenfaces with PCA

In this section, I am presenting the original scheme for determination of the eigenfaces using PCA.

Step 1: Prepare the data

In this step, the faces constituting the training set ($\mathbf{\Gamma}$) should be prepared for processing.

Step 2: Subtract the mean

The average matrix $\mathbf{\Psi}$ has to be calculated, then subtracted from the original faces ($\mathbf{\Gamma}$) and the result stored in the variable:

$$\mathbf{\Psi} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{\Gamma}_i. \qquad (1)$$

$$\mathbf{\Phi}_i = \mathbf{\Gamma}_i - \mathbf{\Psi}; i = 1,\ldots,M. \qquad (2)$$

Step 3: Calculate the covariance matrix

In the next step the covariance matrix C is calculated according to

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{\Phi}_i\mathbf{\Phi}_i^T \qquad (3)$$

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

In this step, the eigenvectors (eigenfaces) u_i and the corresponding eigenvalues $\boldsymbol{\lambda}_i$ should be calculated. The eigenvectors (eigenfaces) must be normalized so that they are unit vectors, i.e. of length 1. The description of the exact algorithm for determination of eigenvectors and eigenvalues is omitted here, as it belongs to the standard arsenal of most math programming libraries.

Step 5: Select the principal components

From M eigenvectors (eigenfaces) u_i, only M' should be chosen, which have the highest eigenvalues. The higher the eigenvalue, the more characteristic features of a face does the particular eigenvector describe. Eigenfaces with low eigenvalues can be omitted, as they explain only a small part of characteristic features of the faces. After M' eigenfaces u_i are determined, the "training" phase of the algorithm is finished.

Improvement of the Original Algorithm

There is a problem with the algorithm described in section 2.5. The covariance matrix **C** in step 3 has a dimensionality of $N^2 \times N^2$, so one would have N^2 eigenfaces and eigenvalues. For a 256 × 256 image that means that one must compute a 65,536 × 65,536 matrix and calculate 65,536 eigenfaces. Computationally, this is not very efficient as most of those eigenfaces are not useful for our task. So, the step 3 and 4 is replaced by the scheme:

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T \qquad (4)$$

$$= \mathbf{AA}^T$$

$$\mathbf{L}=\mathbf{A}^T\mathbf{A}\mathbf{L}_{n,m}=\Phi_m{}^T\Phi_n \qquad (5)$$

$$u_i = \sum_{k=n-1}^{M} v_{1k}\Phi_k,\ i = 1,\ldots,\ldots,M$$

where **L** is a $M \times M$ matrix, v are M eigenvectors of **L** and u are eigenfaces. Note that the covariance matrix **C** is calculated using the formula $\mathbf{C} = \mathbf{AA}^T$, the original (inefficient) formula is given only for the sake of explanation of **A**. The advantage of this method is that one has to evaluate only M numbers and not N^2. Usually, $M << N^2$ as only a few principal components (eigenfaces) will be relevant. The amount of calculations to be performed is reduced from the number of pixels ($N^2 \times N^2$) to the number of images in the training set (M).

In the step 5, the associated eigenvalues allow one to rank the eigenfaces according to their usefulness. Usually, we will use only a subset of M eigenfaces, the M' eigenfaces with the largest eigenvalues.

Classifying the Faces

The process of classification of a new (unknown) face Γ_{new} to one of the classes (known faces) proceeds in two steps.

First, the new image is transformed into its eigenface components. The resulting weights form the weight vector $\Omega_{new}{}^T$

$$\mathbf{w}_k = \mathbf{u}_k{}^T(\mathbf{T}_{new}-\Psi),\ \mathbf{k} = 1,\ldots,\ldots,M' \qquad (6)$$

$$\Omega_{new}{}^T = [\mathbf{w}_1,\mathbf{w}_2,\mathbf{w}_3,\ldots,\mathbf{w}_{M'}] \qquad (7)$$

The Euclidean distance between two weight vectors $d(\Omega_i, \Omega_j)$ provides a measure of similarity between the corresponding images i and j. If the Euclidean distance between Γ_{new} and other faces exceeds - on average - some threshold value , one can assume that Γ_{new} is no face at all. $d(\Omega_i, \Omega_j)$ also allows one to construct "clusters" of faces such that similar faces are assigned to one cluster.

For comparison, two methods are used to describe a face class in the face space. The first method, referred to as the averaging representation, calculates the class vector by averaging the projected vectors from the training images of the corresponding individual. The second method, the point-set representation, describes a face class by a set of vectors projected from all the training images of an individual.

A distance threshold, θ_c, that defines the maximum allowable distance from a face class as well as from the face space, is set up by computing half the largest distance between any two face classes:

$$\theta_c = \frac{1}{2}\max_{j,k}\left\{\left\|\Omega_j - \Omega_k\right\|\right\};\ j,k = 1,\ldots,N_c. \qquad (8)$$

In the recognition stage, a new image, Γ, is projected into the face space to obtain a vector, Ω :

$$\Omega = \mathbf{U}^T(\Gamma - \Psi) \qquad (9)$$

The distance of Ω to each face class is defined by

$$\varepsilon_k^2 = \left\|\Omega - \Omega_k\right\|^2;\ k = 1,\ldots,N_c. \qquad (10)$$

For the purpose of discriminating between face images and non-face like images, the distance, ε, between the original image, Γ, and its reconstructed image from the eigenface space, Γ_f, is also computed:

$$\varepsilon^2 = \left\|\Gamma - \Gamma_f\right\|^2, \qquad (11)$$

where

$$\Gamma_f = \mathbf{U}\cdot\Omega + \Psi. \qquad (12)$$

These distances are compared with the threshold given in equation (8) and the input image is classified by the following rules:

- IF $\varepsilon \geq \theta_c$
  THEN input image is not a face image;
- IF $\varepsilon < \theta_c$ AND $\forall k, \varepsilon_k \geq \theta_c$
  THEN input image contains an unknown face;
- IF $\varepsilon < \theta_c$ AND $\varepsilon_{k^*} = \min_k \{\varepsilon_k\} < \theta_c$
  THEN input image contains the face of individual k^*.

A Euclidean Distance

Let an arbitrary instance x be described by the feature vector

$$x = [a_1(x), a_2(x), ..., ..., a_n(x)] \quad (13)$$

where $ar(x)$ denotes the value of the r^{th} attribute of instance x. Then the distance between two instances x_i and x_j is defined to be $d(x_i, x_j)$:

$$d(x_i, x_j) = \sqrt{\sum \left(ar(x_i) - ar(x_j)\right)^2} \quad (14)$$

The distance measure at a given image location is

$$\varepsilon^2 = \left\| \Phi - \Phi_f \right\|^2 \quad (15)$$

Conclusion

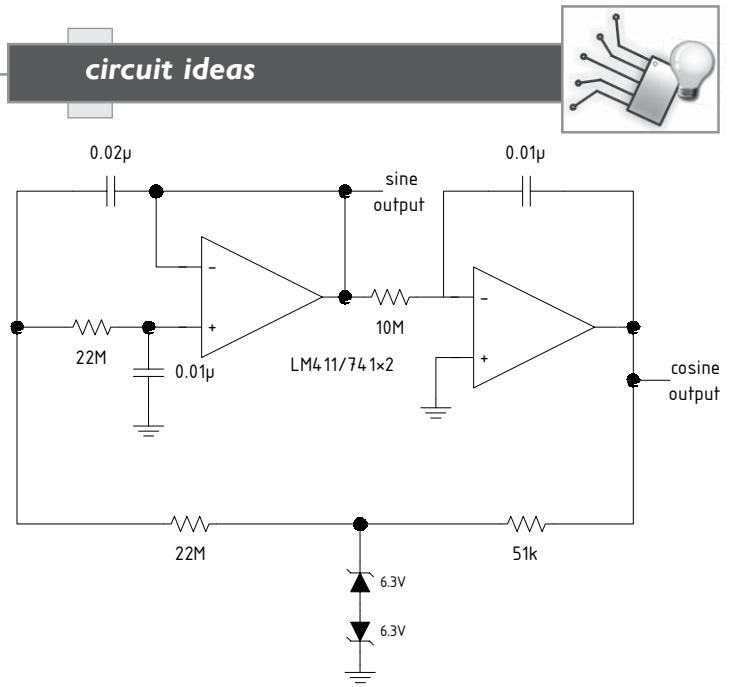
Face recognition system is a system that identifies taken or given pictures face matches or not with the faces from database. If the face is matched corresponding action is taken. Various techniques are available for face recognition process. Eigenface algorithm is faster, simple and efficient among various available algorithms.

This kind of project is applicable for variety of situations starting for security of home, offices, airports. Another approach is to automatically keep track of workers in office. When they come and leave the office can be tracked.

One limitation for eigenface approach is in the treatment of face images with varied facial expressions and with glasses. Also as images may have different illumination conditions. It is quite efficient and simple in preprocessing of image to verify the face geometry or the distances between the facial organs and its dimensions. The application of the symmetrization procedure improves significantly the Eigenface algorithm performance concerning images in unsuitable illumination conditions. ■

(Note: The writer is performing research on face recognition and its implementation issues.)

circuit ideas



1 Hz quadrature oscillator
(Source: Art of Electronics)

Fingerprint Recognition
and
its use in authorization

Dil Kumar Shrestha
2059 Electronics

Everyone prefers better authentication method than a mere password for access to critical information (data) and want to set up a more secure environment for his/her employees and assets. These days we frequently hear the news regarding the security systems being compromised and broken into. Now there is a new technology for authentication and its called Biometrics.

Biometrics is the science and technology of authentication by measuring the person's physiological or behavioral features. The term is derived from the Greek words "bios" for life and "metron" for degree. Biometrics is a technology that is used in the security industry and is integrated with other authentication applications and technologies, like domain access, single sign-on, smart cards, encryption, remote access, and digital signatures. Biometrics authenticates personals based on their unique physical body demographics or behavioral characteristics. In other words, a person can be considered as his/her own password. Biometrics gives us an alternative and higher security compared to conventional alphanumeric passwords or PIN identification due to the fact that passwords and PIN's can be easily compromised.

automatic fingerprint matching is becoming increasingly popular in systems, which control access to physical locations, computer/network resources, bank accounts or register employee attendance time in enterprises

Biometrics technology currently implements measuring and analyzing of human physiological characteristics such as fingerprints, eye retinas and iris, voice patterns, facial patterns, and hand measurements, for authentication purposes. In a typical Biometric security system, a person registers with the system when one or more of his physiological characteristics are obtained, processed by a numerical algorithm, and stored into a database. Ideally, when the person logs in to the system, all of his features will match 100% with the information in the database and thus authenticating him. However an unauthorized person will not fully match the information in the database, so the system will not allow him to access the system. However, current technologies are nowhere close to matching this ideal scenario.

Nowadays, automatic fingerprint matching is becoming increasingly popular in systems, which control access to physical locations, computer/network resources, bank accounts, or register employee attendance time in enterprises. Straightforward matching between the fingerprint patterns to be identified and many already known patterns would not serve well due to its high sensitivity to errors. Various noises, damaged fingerprint areas, or the finger being placed in different areas of fingerprint scanner window and with different orientation angles, finger deformation during the scanning procedure etc are the major problems that prevent 100% fingerprint matching.

A more advanced solution of to this problem is to extract features of so called 'minutiae points' from the fingerprint image, and check matching between the sets of fingerprint features. However, the above outlined solution requires sophisticated algorithms for reliable processing of the fingerprint image, noise elimination, minutiae extraction, rotation and translation-tolerant fingerprint matching. At the same time, the algorithm must be as fast as possible for comfortable use in applications with large number of users. For developers who intend to implement the fingerprint recognition algorithm into a microchip, compactness of algorithm and small size of required memory may also be important. Though many fingerprint identification algorithms are proposed, in reality, achieving satisfactory fulfilment of all the discrepant requirements is still important problem.

Neurotechnologija has developed fingerprint identification algorithm VeriFinger, for biometric system integrators. VeriFinger has the capabilities of the most powerful fingerprint recognition algorithms: Fingerprint matching speed is one of the highest among the competing identification algorithms. Fingerprint enrolment time is 0.2 - 0.4 sec., and VeriFinger can match 30,000 fingerprints per second in 1:N identification mode. VeriFinger is available as a software development kit (SDK), but source code may also be made available for developers.

VeriFinger fingerprint recognition algorithm follows the commonly accepted fingerprint identification scheme, which uses a set of specific fingerprint points (minutiae). However, it contains many proprietary algorithmic solutions, which enhance the system performance and reliability.

Adaptive image filtration algorithm allows to eliminate noises, ridge ruptures and stuck ridges, and extract minutiae points reliably even from poor quality fingerprints, with processing time of about 0.2 - 0.4 seconds (all times are given for Pentium 4, 3 GHz processor). VeriFinger functions can be used in 1:1 matching (verification), as well as 1: N mode (identification). VeriFinger is fully tolerant to fingerprint translation and rotation. Such tolerance is usually reached by using Hough transform-based algorithms, but this method is quite slow and unreliable. VeriFinger uses a proprietary fingerprint matching algorithm instead, which currently enables to match 30,000 fingerprints per second and identify fingerprints even if they are rotated, translated and have only 5 - 7 similar minutiae (usually fingerprints of the same finger have 20 - 40 similar minutiae). VeriFinger does not require presence of the fingerprint core or delta points in the image, and can recognize a fingerprint from any part of it.

VeriFinger can use the database entries, which were pre-sorted using certain global features. Fingerprint matching is performed first with the database entries having global features most similar to those of the test fingerprint. If matching within this group yields no positive result, then the next record with most similar global features is selected, and so on, until the matching is successful or the end of the database is reached. In most cases there is fairly good chance that the correct match will be found already in the beginning of the search. As a result, the number of comparisons required to achieve fingerprint identification decreases drastically, and correspondingly, the effective matching speed increases. VeriFinger has the fingerprint enrolment with features' generalization mode. This mode generates the collection of the generalized fingerprint features from three fingerprints of the same finger. Each fingerprint image is processed and features are extracted. Then the three collections of features are analyzed and combined into a single generalized features collection, which is written to the database. This way, the enrolled features are more reliable and the fingerprint recognition quality considerably increases.

Hence Fingerprint Recognition provides a great solution for doors, computer room access, desktop login authentication, and application integration. Fingerprint recognition provides a low cost biometrics solution and with its small designs makes it a prime choice when setting up a high security solution. Most probably within a few year it would be the most common security system, that is preferred by most of the industries, institutions and individuals those require to maintain several statements, data, information etc more secure. Then you are what your thumb says! ■

RADAR

(Radio Detection and Ranging)

Nilesh Man Shakya
2059 Electronics

Brief History

The credit for the development of the RADAR technology goes to Heinrich Hertz, who first succeeded in generating and detecting the radio waves experimentally although the prediction of the existence of electromagnetic waves accomplished by James Clerk Maxwell.

In 1904, Christian Huelsmeyer patented the device called '*telemobiloscope*' for the detection of the presence of ships at the distance of about 3 km only. In 1917, Tesla proposed principles regarding frequency and power levels for the early RADAR. Scientists at the Naval Research Lab (NRL) in US used continuous wave interference RADAR to detect an aircraft but it could not determine its location and velocity. In 1936, first pulse RADAR was demonstrated at the range of 2.5 miles which was able to determine the location and the velocity of the aircraft.

In Germany, Hans Eric Holliman worked in the field of microwaves, which were to become the basis of all modern RADAR systems. In 1934, the company called 'GEMA', established by him and his partner Hans-Karl Von Willisen built the first commercial RADAR system for detecting ships at the range of 10 km.

The research and development in RADAR technology increased enormously during the World War II for the detection of enemy aircrafts and ships. Since then, the RADAR has found its utility not only in defense but also in weather observation, space exploration and many others.

Principle of RADAR

RADAR is the remote detection system used in locating and identifying distant objects. The detection of the objects is done by first transmitting a powerful radio waves and analyzing the echoes (the reflected signals) that bounce off the objects in their paths. RADAR can be used to find out the different properties of the reflecting objects such as its shape, distance, speed and direction of motion.

The basic principles used in all RADAR systems are the echo and the Doppler shift. RADAR uses both echo and Doppler Shift to locate and identify the objects. These two terms can be easily explained using the example of a sound wave.

RADAR systems try to get the information about the distant objects from the reflections of the deliberately generated electromagnetic waves at radio frequency

We experience echo all the time. When we shout into a well, we can hear our voice a little later. This is due to the reflection of sound waves from some objects in its path. This phenomenon is called Echo.

Doppler shift is also an equally common phenomenon. If music is played inside a moving car which is coming toward us, we can observe that the music gets louder and louder as the car approaches us and diminishes as the car gets away. This effect is known as the Doppler shift caused due to the change in the wavelength of the wave leaving the moving object.

In RADAR system, instead of sound waves, radio waves are used.

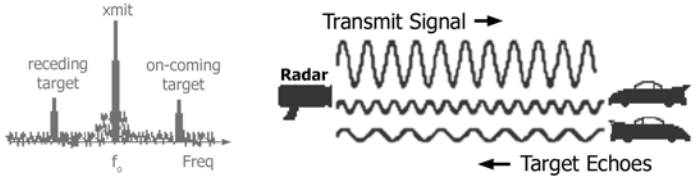


Illustration of Echo and Doppler Shift

Components of a RADAR system

RADAR system basically consists of a transmitter system, antenna, receiver system and the display.

Transmitter System

The main task of the transmitter system is to transmit the desired RADAR signals to the target regions. Basically, the transmitter system consists of an oscillator, a modulator and a transmitter itself. The oscillator produces the RADAR signal of required frequencies. The oscillator produces an accurate/stable reference frequency for the calculation of the Doppler shift. The job of the modulator is to modulate or vary the signals received from the oscillator. For example, in a simple pulse RADAR system, the modulator simply turns the signal ON and OFF. In the transmitter, the power of the signal is amplified. High power amplification is needed in the RADAR signals because only less amount of signals get reflected back. After the amplification, the signal is fed to the focus of the antenna, which is usually parabolic or concave shaped metal dish. Most antennas have servo-mechanism so that they can point to desired directions and scan the whole area.

A common antenna for RADAR system

Receiver System

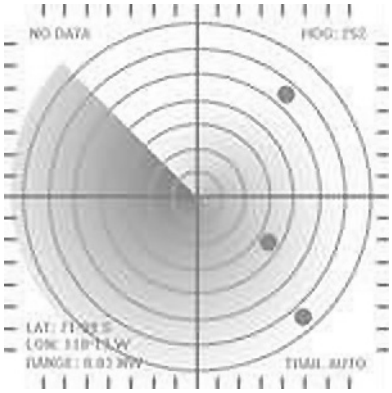
The receiver system basically detects and analyzes the faint echoes of the RADAR signals received due to the reflection from the distant objects. The main components of the receiver system are the antenna, the duplexer, the receiver and the display unit.

A common antenna can be used for the transmission as well as reception of the RADAR waves. The antenna gathers the weak returning RADAR signals and converts it to the electric current. Since the same antenna is used to transmit and receive signal so there is a need of duplexer which enables a RADAR system to transmit high power RADAR signals and also receive the faint echoes.

The receiver processes the received signals and sends to the display. First, the analog signal is digitized using analog to digital converter (ADC). The noise and other interferences are then removed and the low power signal is amplified to the required range. The digital signal is then further processed using Doppler filtering. The Doppler filtering helps to identify the multiple targets and their orientations as well as the speed.

The final stage of the receiver system is the display in which the processed signals are transformed to the useful information. In early days, a simple amplitude scope, which displayed the received signal strength as the function of distance from the antenna, was used. Modern displays use the plan position indicator (PPI).

The PPI displays the direction of the target in relation to the RADAR system as an angle measured from the top of the display, while the distance to the target is represented as a distance from the center of the display.



RADAR Screen

RADAR frequencies

RADAR system can work on various signal frequencies. The frequency of operation is usually adjusted to find the optimal propagation path to the desired target region of the globe as ionosphere condition changes.

The standard RADAR-frequency band nomenclature as per the IEEE standard is given below:

Applications of RADAR

RADAR systems try to get the information about the distant objects from the reflections of the deliberately generated electromagnetic waves at radio frequency. The information obtained is the detection of the presence of target objects in the

Band Designation	Nominal Frequency Range	RADAR Frequency Ranges*
HF	3-30 MHz	None
VHF	30-300 MHz	138-144 MHz 216-255 MHz
UHF	300-1000 MHz	420-450 MHz 890-942 MHz
L	1000-2000 MHz	1215-1400 MHz
S	2000-4000 MHz	2300-2500 MHz 2700-3700 MHz
C	4000-8000 MHz	5250-5925 MHz
X	8000-12000 MHz	8500-10680 MHz
K_u	12-18 GHz	13.4-14.0 GHz 15.7-17.7 GHz
K	18-27 GHz	24.05-24.25 GHz
K_a	27-40 GHz	33.4-36.0 GHz
V	40-75 GHz	59-64 GHz
W	75-110 GHz	76-81 GHz 92-100 GHz
Mm	110-300 GHz	126-142 GHz 144-149 GHz 231-235 GHz 238-248 GHz

**based on ITU Assignments for North and South America, ITU std 521-1984*

presence of other obstacles, recognition of the targets and estimation of the target parameters such as the range, elevation, orientation, velocity or acceleration.

The application of modern RADAR can be found in the military, civilian and in the scientific fields. Military applications include the search and surveillance of enemy targets; navigation, control and guidance of guided missiles; battlefield surveillance; and anti-aircraft fire control. For the civilian purpose, it is used in aircraft navigation, collision avoidance, air craft control, ship navigation, tracking of vehicles, traffic law enforcement, and many others. Similarly, scientific applications involve remote sensing of the Earth's environment from aircraft and satellites; weather RADAR for study and monitoring of precipitation, clouds, and major weather disturbances; ground mapping; ground penetrating RADAR for detection of buried objects; foliage penetrating RADAR for detection of hidden targets and high resolution imaging of objects by synthetic aperture imaging RADARs. Hence, RADAR has wide utility in almost every field of science and technology and its usage is increasing day by day. ■

References

1. M.I. Skolnik, *Introduction to RADAR Systems,* New York: McGraw Hill, 1980.
2. *Encyclopedia of Electronics Engineering*
3. www.howstuffworks.com
4. www.wikipedia.com
5. www.copRADAR.com
6. *Microsoft Encarta*

The War of the Worlds

HD DVD vs. Blu-ray Disc

Saurav Dhungana
2060 Electronics

The battle lines are drawn

It seems like yesterday when watching a home video meant renting those bulky video cassettes and then feeding into our VHS system. The whole experience was accompanied by a fuzzy, gloomy video and a less than pleasing sound quality. But the scenario is a lot different today. The advent of the optical media starting with the CD and more recently, the DVD has certainly taken our home video experiences a notch. The digital revolution brought with it a new level of clarity and refinement unseen and unheard in the VHS era. The modern home video experience has attained such levels of technological sophistication with its crystal clear, high definition video formats and theatre like digital surround sounds, that it is well on its way to eclipse the traditional movie theatre experience.

But just when all you movie buffs thought that it couldn't get any better, well it just did. This time however there's a twist. Two decades after the competing video formats battled for supremacy in our living rooms (see next para); a new war is looming between two incompatible types of high-definition video discs. One, called HD DVD, is the official choice of the group that backs conventional DVDs, and with the support of technological powerhouses like, Microsoft and Toshiba. The other, called Blu-ray, is spearheaded by more than a dozen big-name consumer-electronics and high-tech companies

This however is like the history repeating itself. The home video market has endured two format wars already, starting with the battle between Sony's Betamax and JVC's VHS in the mid-1970s. The fight lasted a little more than a decade, with the VHS share growing from about 75 percent of the market in 1980 to 95 percent in 1988, despite Betamax's reputation for better picture quality. Sony finally abandoned its Betamax product line in 2002.

In the mid-1990s, Sony and Philips Electronics backed a new format for video discs, while Toshiba and Warner Bros. supported a more radical shift to a higher-capacity approach. Sony and Philips eventually backed a compromise approach based largely on Warner and Toshiba's technology, and the DVD format was announced in December 1995.

But a format war broke out anyway when a handful of consumer-electronics manufacturers and a few of the major studios offered – briefly – a pay-per-play approach called DivX.

Things as they are

a new war is looming between two incompatible types of high-definition video discs

This time around, a split-the-baby compromise is virtually impossible, both sides acknowledge. The two camps are still trying to strike a last-minute deal and agree on common technical standards. However, in addition to the money and egos involved, the physical differences in the two disc formats are keeping the two sides far apart.

That is because the core difference lies with a single aspect of the disc – a thin layer of plastic that sits just above the metal surface on which data is written. An HD-DVD disc calls for a 0.6 millimeter coating, while a Blu-ray disc requires 0.1 millimeters.

While that does not seem like much, the half-millimeter gap amounts to a technological chasm. HD DVD's thicker coating is the same as current DVDs, which allows manufacturers to use existing disc-stamping equipment to make the new discs. That gives HD DVD a significant cost advantage and more predictability about what those costs will be.

HD DVD players can also rely on some of the same technology as conventional DVDs, making it easier to build players that can handle both generations of disc. That is an important feature, given how many conventional DVDs movie buffs already own.

Blu-ray's thinner coating requires all new manufacturing equipment, but it's the secret behind the disc's higher capacity. Because the laser travels through a

thinner layer of resin, it is able to focus more sharply and write 67% more data onto the disc itself.

The Studio Factor

Major Hollywood studios exacerbate the problem by splitting their support between the two formats, each of which promise to deliver richly detailed pictures and cinema-quality sound. Guided by differing visions for the high-definition future, half of the studios have announced plans to release HD DVD discs, and the other half are expected to back Blu-ray. Should the format war reach consumers, the battle could be over quickly. Sony plans to include a Blu-ray drive in its highly anticipated PlayStation 3 video-game console, which could put Blu-ray in several million homes in a matter of months. The worst casualties could be the video enthusiasts who spend close to $1,000 on a new disc player only to have it become quickly obsolete. Analysts say a format war would also slow the transition to high-definition discs, reducing sales for consumer-electronics manufacturers and studios alike.

Disney is backing Blu-ray, which offers at least 25 gigabytes per disc, compared with 15 gigabytes for basic HD DVD discs and 4.7 gigabytes for conventional DVDs.

Executives at Warner Bros., which has announced plans to release HD DVD discs, counter that the Blu-ray group has not been able to answer critical questions about manufacturing costs, the discs' resistance to warping and other reliability issues. They say the HD DVD group has proved its ability to mass-produce double-layer discs and hybrids that combine a conventional DVD on one side with a high-definition movie on the other – a key product for movie fans who have yet to buy an HDTV.

In spite of the format dilemma, many consumer-electronics executives are eager to shift to high-definition discs because profit margins have shrunk dramatically on conventional DVD players and sales have started to drop. The worldwide sales of DVD players peaked in 2004 at $20.1 billion and are expected to drop in 2005 for the first time by 1 percent to $19.8 billion, falling to $15.3 billion in 2010.

While the studios are leery of disturbing that cash cow, they also want to replace DVDs with a format that is less vulnerable to piracy.

Where we go from here

It's perfectly obvious that no one wants this format war. Not the device manufacturers. Not the studios. Not the consumers. Consumers will delay their purchase if there is confusion, and that results in market stagnation for everyone. The problem is that both believe that their solution is the best. A better solution would be for the two sides to use one camp's disc structure and the others software, generating royalties for both. But doing so would require one side or the other to give up the core advantages of its format — the cost, compatibility and reliability, strengths of HD DVD, or the capacity of Blu-ray. However, it can certainly be said that there'll be only one winner in the end and the other has to give way. Let us hope as consumers that this conflict is settled sooner rather than later and we get to enjoy the best of both worlds. ■

	Current DVD	HD-DVD	Blu-Ray
Data Capacity (per layer)	4.7 GB	15 GB	25 GB
Maximum Image Resolution (pixels)	640 × 480	1920 × 1080	1920 × 1080
Thickness of Recorded layer	0.6 mm	0.6 mm	0.1 mm
Key Patent Holders	10 Electronics Companies & Time Warner	Toshiba, NEC, Time Warner	Sony, Philips, Matsushita, Pioneer
Studio Backers	All	Warner Bros., Universal, Paramount	Sony & Disney
Retail Launch	1997 Christmas	2005	Spring 2006

Comparative analysis of the formats in question

Unicode and Our Perspective

Aashish Poudel
Bikash Sharma
2060 Electronics

Background

After many years of developing scores of fonts on ASCII or EBCDIC format, the people and the world at large, felt the need to develop a common ground so that the individual languages can be supported, no matter what the platform, program or the language is. The need was urgent and would mean a lot to billions of people across the globe who were fed up with the conversion system of any normal font yielding many errors and lapsing some characters. The answer was definitely universal code that could systematize and standardize the use of fonts. There were many attempts for encoding the characters in different part of the world like "JIS" encoding, "KOI8" for Russian, various "ISCII" standards for the Indian languages, and so on. Moreover, there were proprietary encodings dreamed up by operating system makers such as Apple, and Microsoft with its "code pages". The late 80s saw two independent developments to create a single unified character set viz. ISO 10646 and Unicode Consortium. They, however, shortly realized that two parallel paths don't yield uniformity which is the main basis for 'Unicode' concept. They not only agreed to join efforts, but also keep the code tables of the Unicode and ISO 10646 standards compatible and they closely coordinated any further extensions. Unicode versions 1.1, 3.0, 3.2 and 4.0 (with 96,382 assigned characters are the direct result of this synergy. All Unicode versions since 2.0 are compatible, only new characters will be added; no existing characters will be removed or renamed in the future. Hence, it is backward compatible.

How it works

UCS (Universal Character Set) characters U+0000 to U+007F (ASCII) are encoded simply as bytes 0x00 to 0x7F (ASCII compatibility). This means that files and strings which contain only 7-bit ASCII characters have the same encoding under both ASCII and UTF-8. All UCS characters above U+007F are encoded as a sequence of several bytes, each of which has the most significant bit set. Therefore, no ASCII byte (0x00-0x7F) can appear as part of any other character. The first byte of a multibyte sequence that represents a non-ASCII character is always in the range 0xC0 to 0xFD and it indicates how many bytes follow for this character. All further bytes in a multibyte sequence are in the range 0x80 to 0xBF

'Microsoft XP (Nepali)' and 'Nepalinux' has been recently launched which is a gift of Unicode System

In ASCII or EBCDIC you store your character one-per-byte in memory, ASCII using 7 (or 8, 8th bit for graphic characters) and EBCDIC 8 bits of each byte. Unicode characters are identified by number or "code point", usually given in hexadecimal, so for example the Devanagari letter " द " is 0926 (2342 in decimal), usually written U+0926.

Unicode currently defines just under 100,000 characters, but has space for (65,536×17=) 1,114,112 code points. They are organized into 17 "planes" of 65,536 characters, numbered 0 through 16. Plane 0 is called the "Basic Multilingual Plane" or BMP and contains pretty well everything useful. In particular, it contains every character that had ever been available to a computer programmer before Unicode came along. A Unicode code point between U+0000 and U+FFFF is defined as Basic Multilingual Plane (BMP) Code Point. Past the BMP, planes 1 through 16 are used for exotic, rare, and historically important characters. Obviously, all Devanagari (U+0900-U+097F) fall under BMP.

Along with the characters, Unicode also defines methods for storing them in byte sequences in a computer. There are three approaches; named UTF-8, UTF-16, and UTF-32 (although encoding schemes are extended to UTF-16BE, UTF-16LE, UTF-32, UTF-32BE, UTF-32LE). "UTF" may be

explained as standing for Unicode Transformation Format, or the UCS Transformation format

UTF-32 is about the simplest imaginable way of storing characters. As the name suggests, you use 32 bits or four bytes for each character. So each of the example characters would be stored as a 4-byte number with values 38, 1046, 20013, and 66374 respectively. This corresponds to the way most modern C compilers store characters if only brief extension of ASCII is needed; using 32 bits to store characters seems utterly wasteful. Also, the old fashioned C-language routines like strcpy, strcmp, and so on won't work with this because they go a byte at a time and there are lots of bytes filled with zeros.

UTF-16 stores Unicode characters in sixteen-bit chunks. All the characters in the BMP appear as themselves, but past BMP plane characters don't fit in sixteen bits. To handle this, Unicode applies what it calls the "Surrogate" blocks which aren't normally used for ordinary characters. UTF-16 is probably what most people thought most programmers would use for Unicode; this is reflected in the fact that the native character type in both Java and C# is a sixteen-bit quantity. Of course, it doesn't really represent a Unicode character, exactly (although it does most times), it represents a UTF-16. UTF-16 is about the most efficient way possible of representing Asian character strings.

Files containing only 7-bit ASCII characters are unchanged when viewed in the Unicode UTF-8 encoding, so plain ASCII files are already valid Unicode files. With UTF-8, up to four 8-bit bytes may be required to access all defined Unicode characters.

None of the three UTF approaches (-32, -16, -8) are really better than any of the others. UTF-8 works better with traditional C programming practice, while Java and C# share a sort-of-UTF-16 culture. These variable-width encoding schemes have been developed to minimize the number of bytes required to store Unicode characters.

Our perspective

Every now and then, we have been suffering with the Nepali font problems. It's because various locally developed Nepali fonts such as Himali, Preeti, Kantipur etc. has been developed and used to fulfil the need of documents required in Nepali language. All these fonts use the Devanagari system as their base but have different coding system. This brings about a lot of complication in downloading the documents from one PC to other, especially when the document prepared in a particular font doesn't get downloaded in the other computer in the absence of the same font in the latter one. To view the document, the exact font had to be installed in the receiving computer. Sometimes even a single page of document may contain several kinds of fonts which make the downloading process even more complicated and time consuming. But with the Unicode Devnagari font installed in the keyboard, this problem is completely minimized. There are umpteenth of sites waiting for your searches and free downloads. The 'Mangal' font comes with Microsoft's Windows XP and Windows 2000. The 'Arial Unicode MS' (arialuni.ttf) is Arial Unicode MS font is a full Unicode font. It contains all of the characters, ideographs, and symbols defined in the Unicode 2.1 standard. So, it supports unicode in multi-languages including Devnagari. It is worthy here to mention that 'Madan Puraskar Pustakalaya' in collaboration with United Nations Development Program and the Ministry of Science and Technology have accomplished a Font Standardization Project by writing a Unicode-based Nepali version for the two different Nepali keyboard layouts drivers viz. '**Nepali Unicode traditional'** and '**Nepali Unicode Romanized' with input language as 'Sanskrit'.**

All Unicode CD's come with instructions for local Unicode setup.

1. To install Arial Unicode MS Font, you should opt for 'Choose advance customization of applications' during the installation of MS Office. If you have Office already, go to 'Add/Remove Programs' of Control Panel to change the features, Choose 'advance customization of applications' and Click Next. Next to Office Shared Features, click the plus sign (+). Next to International Support, click the plus sign (+). Click the icon next to Universal Font, and then select the installation option you want.
2. To input characters that are not on your keyboard, Press and hold the ALT key, and then press the keys on the numeric keypad that represent the decimal code value of the character you want to input.

Unicode has started to replace ASCII, ISO 8859 and EUC (EU code) at all levels

Microsoft Windows is built on a base of Unicode; AIX, Solaris, HP/UX, and Apple's MacOS all offer Unicode support. All the new web standards: HTML, XML, etc. are supporting or requiring Unicode. The latest versions of Netscape Navigator and Internet Explorer both support Unicode. Sybase, Oracle, DB2 all offer or are developing Unicode support. Most of Microsoft's Office applications supported Unicode for several versions now. Meanwhile, both 'Microsoft XP (Nepali)' and 'Nepalinux' has been recently launched which is a gift of Unicode System. ■

VSAT

An overview

Anup Dhital
2059 Electronics

VSAT stands for "**V**ery **S**mall **A**perture **T**erminal" and refers to receive/transmit terminals installed at dispersed sites connecting to a central hub via satellite using small diameter antenna dishes (0.6 to 3.8 meter). It is used as a means for one way (receive-only) or two way (interactive) communications. VSAT technology represents a cost effective solution for users seeking an independent communications network connecting a large number of geographically dispersed sites. VSAT networks offer value-added satellite-based services capable of supporting the Internet, data, LAN, voice/fax communications, and can provide powerful, dependable private and public network communications solutions.

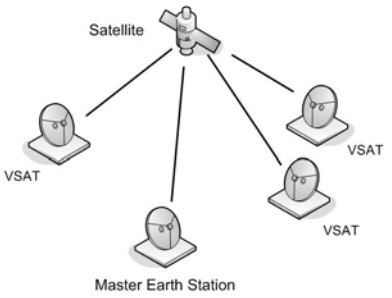
VSAT Components

A VSAT network basically consists of three components:

1. Master Earth (Ground) Station or Hub Station

This is the spacecraft and network control center for the entire VSAT network. The configuration, monitoring and management of the VSAT network as well as recording spacecraft and system data are done at this location. Traffic requests are processed and set up and traffic channels are assigned as an on-demand basis.

The Master Earth Station has a large antenna (generally 5 to 8 meters in diameter), a fully redundant electronics, self-contained backup power system, and a regulated air conditioning system.



General VSAT Communication System

2. The VSAT remote earth station

This is the hardware installed at the customer's premises that includes the outdoor unit (ODU), the indoor unit (IDU) and the interfacility link (IFL). The VSAT outdoor unit consists of a standard 1.8 meter offset feed antenna, a solid state power amplifier (SSPA), a Low Noise Amplifier (LNA), and a Feed horn. The indoor unit is a VCR-sized unit that houses the communications electronics that includes interface with the customer's equipment such as computers or telephones. The IFL consists of coaxial cables that connect the outdoor unit to the indoor unit.

VSAT technology represents a cost effective solution for users seeking an independent communications network connecting a large number of geographically dispersed sites

The Outdoor Unit (ODU)

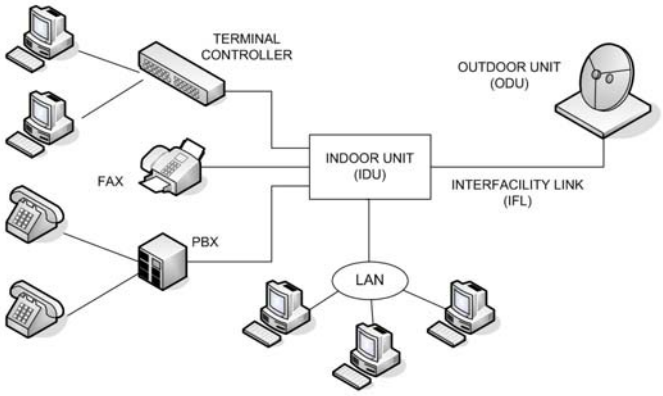
It consists of reflector dish and radio electronics on the dish.

- The satellite dish is a reflector that redirects the satellite signal and focuses it toward the feed horn. The antenna size may range from 70 cm to 90 cm in diameter for C-band and 1.8 or 2.4 meters for Ku-band.
- The feed horn is a snow cone shaped instrument that helps to cut down the ambient signals so that only signals from satellite dish reach the LNB (Low Noise Block).
- The Low Noise Amplifier (LNA) filters noise and directs the signal to the transceiver. The LNB receives satellite signal from space and 2 Watt transmitter transmits to the satellite orbit. For example the transceiver, CODAN in Nepal, basically down converts the incoming 4 GHz signal to 70 MHz IF signal and up converts the outgoing 70 MHz signal to 6 GHz signal.

- Waveguide is provided as a portion of satellite dish electronics to capture the incoming signal.
- A Solid State Power Amplifier (SSPA) is also connected to the waveguide which amplifies the incoming signal even at higher frequencies.

3. Satellite

All signals sent between the VSAT earth stations are beamed through the geostationary satellite, which is orbiting at nearly 36,000 km above the ground. The hub station or the remote earth station sends the signals to be transmitted towards the geostationary satellite where the signal is processed and directed to the station/s where the data is to be sent. The VSAT operators have to buy bandwidth with the satellite owners before they can use the satellite.



Remote Earth Station of a VSAT System

Modulation and Coding

The modulation and coding in VSAT communication depends on number of factors such as antenna diameter, received or transmitted power, bandwidth, transmission rate, cost etc. Unfortunately most of these factors work against each other. For example a system with a modulation scheme like coded QPSK has advantages in terms of C/N (channel to noise ratio) and bandwidth but this is expensive and sensitive to phase noise. So different modulation schemes are used according to the importance of various factors but at a cost or sacrifice of some other factors. Different modulation schemes used in VSAT communication are PSK, QPSK, OQPSK, BPSK, FSK, MSK etc.

Regulation Issues for VSAT

It is necessary and important that VSATs are operated in a manner that doesn't cause excessive interference to other users. For this the International Telecommunication Unit (ITU) has laid down some broad conditions which must be followed by every VSAT system users and vendors. Besides ITU some other organizations also work for the regulation of such communication in a region or country. But these organizations must be associated with ITU. Radiocommunication Agency (RA) is one of such organization that operates in Europe.

One of the most important issues that are conducted by ITU is the frequency allocation. VSAT space segment providers offer three types of satellite beams: spot, hemisphere and global. Spot beams are available in both Ku-band (12-16 GHz) and C-band (4-6 GHz). Spot beams are generally high power thus allowing smaller antenna size (for e.g. 1.8 m for Ku-band and 3 m for C-band). Hemisphere and Global beams have a much larger footprint and thus weaker signal. However C-band is mostly used for these two beams because of low rain induced signal degradation (1-2 dB) compared to Ku-band (5-10 dB) at a cost of greater interference among the signals and larger antenna size for its lower frequency range (i.e., lower gain).

Advantages and Applications

The following advantages and applications of VSAT would be very helpful in understanding its features:

- The service cost is distance insensitive i.e. cost doesn't depend upon the distance between the hub station and remote stations. The customer has to pay only for the data throughput used.
- With the availability of bigger bandwidth for transmission, VSAT supports video and audio communications. The most happening example of which in today's date is Video Conferencing for education, training, corporate communications, marketing etc.
- With the popularity and development of VSAT system, its installation is no longer a matter of problem. Nowadays whenever any organization or person wants to have a VSAT system, the service providers send their field technicians who will erect the antenna in hours and usually complete the installation of the Indoor Unit on the same day. The wireless nature of the VSAT has made the installation process

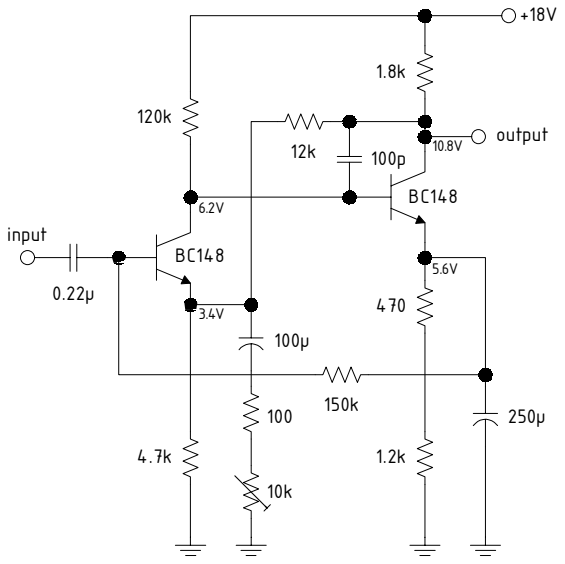
even simpler and almost at any location within the footprint of the satellite. The job of taking (buying) frequencies from the satellite belongs to the service providers.

- VSAT networks offer excellent security against unauthorized access. All transmission over the system is scrambled in digital formats and gaining access to such VSAT system without authorization is virtually impossible.
- A VSAT network supports thousands of VSAT remotes and a new site can be added easily by just installing a remote unit. Different add-on services such as internet, voice capabilities, video applications etc can be added to the existing VSAT remote unit by installing additional modules to the Indoor Unit.
- One of the most important and popular service that is been taken through VSAT is the internet service. With VSAT, internet speed/ bandwidth is no longer a topic to bother about. With the aid of VSAT we can have mobile satellite Internet system that can be mounted on a truck, RV, or a trailer. VSAT terminals can transmit data to the network hub at speeds up to 2 Mbps, with data downloads of up to 60 Mbps, satisfying bandwidth-intensive applications using IP data. ■

References

1. Prof. Charles J. Huges, Prof. David Parsons, Prof. Gerry White, *Satellite Communication System*, 3/e, IEE Telecommunication Series 38
2. grc.nasa.gov
3. telesat.ca
4. vsat-systems.com
5. spacenet.gov
6. viasat.com
7. intelsat.com
8. vsatus.com
9. skycasters.com
10. satsig.com
11. bcsatellite.net
12. tiscsat.com
13. satcoms.org.uk
14. telesindo.com

circuit ideas



Dynamic mic preamp with adjustable gain, 13dB-40dB (Source: Philips Application Book,Audio Amplifier Systems)

Wi-Fi Wireless Networking

Praswish Maharjan
Nijjal Nyachhyon
2059 Electronics

Computer Networking is the interconnection of the different computers. The development of the computers along with the Internet has made information field grow exponentially. Notebooks have made computing mobile. Almost as soon as the notebook computers appeared, many people had a dream of walking into an office and magically having their notebooks connected to the Internet. This gave rise to the Wireless Fidelity system, also known as the Wi-Fi system. Wireless Fidelity system is the wireless way to handle networking. It is also known as 802.11 networking and wireless networking. The computers connect to the network using radio signals. It has made networking mobile.

WECA (Wireless Ethernet Compatibility Alliance) is an industry group that includes all of the major manufacturers of 802.11b equipment. Their twin missions are to test and certify that wireless network devices from all their member companies can operate together in the same network and to promote 802.11 networks as the worldwide standard for wireless LANs. WECA's marketing geniuses adopted the more "friendly" name of Wi-Fi (short for Wireless Fidelity) for the 802.11 specifications and changed their own name to the Wi-Fi Alliance. Wi-Fi is compatible with Ethernet; it should be possible to send an IP packet over the wireless LAN the same way a wired computer sent an IP packet over Ethernet.

802.11 standard Specification

Originally 802.11 ran at either 1 Mbps or 2 Mbps. But since this was relatively slow, immediately work began on faster standards. The 802.11b was the first version to reach the marketplace. It is the slowest and least expensive of the three. 802.11b transmits at 2.4 GHz and can handle up to 11 megabits per second (although 7 Mbps is more typical, and 802.11b may fall back as low as 1 or 2 Mbps if there is a lot of interference). It uses **Complementary Code Keying (CCK)** technique. The 802.11a uses wider frequency band and runs at the speed up to 54 Mbps (although 30 Mbps is more typical). 802.11a transmits radio signals in the frequency range above 5 GHz. This range is "regulated," meaning that 802.11a gear utilizes frequencies not used by other commercial wireless products like cordless phones. It uses much more efficient coding techniques that also contribute to the much higher data rates. The technique used is known as **orthogonal frequency division multiplexing (OFDM).** The 802.11g is a mix of both worlds. It operates at 2.4 GHz (giving it the cost advantage of 802.11b) but it has the 54 mbps speed of 802.11a. Like 802.11a, 802.11g also uses OFDM technique.

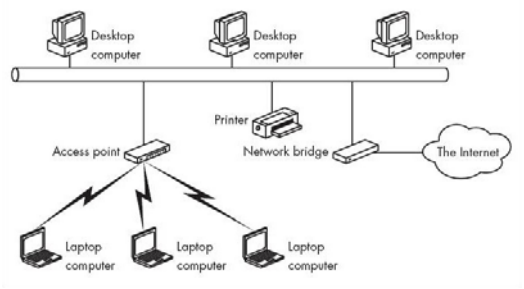


Fig 1. Wireless LAN with Base station

Working of Wi-Fi system

Wireless LANs can operate in one of the two configurations i.e. with a base station and without a base station. In the former case, all the communication goes through the base station called an **access point** in 802.11 lingo. An example of this type of networking will be, all the laptop being connected to internet through the common access point. Usually the base stations are connected to each other through Ethernet. In the later case, the computer would just send signals to one another directly. This mode is sometimes called **ad hoc** networking. A typical example is two or more people sitting down together in a room not

Almost as soon as the notebook computers appeared, many people had a dream of walking into an office and magically having their notebooks connected to the Internet

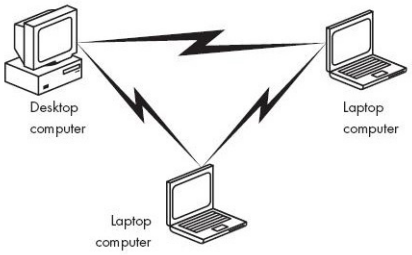


Fig 2. Wireless LAN without Base Station

equipped with a wireless LAN and having their computer just communicate directly.

A Wi-Fi System is immune to interference because it uses spread spectrum for signal transmission. Spread spectrum is a family of methods for transmitting a single radio signal using a relatively wide segment of the radio spectrum. Wireless Ethernet networks use two different spread-spectrum radio transmission system, called **FHSS (frequency-hopping spread spectrum)** and **DSSS (direct-sequence spread spectrum).** Some older 802.11 networks use the slower FHSS system, but the current generation of 802.11b and 802.11a wireless Ethernet networks uses DSSS.

Spread-spectrum radio offers some important advantages over other types of radio signals that use a single narrow channel. Spread-spectrum is extremely efficient, so the radio transmitter can operate with very low power. Because they operate on a relatively wide band of the frequencies, they are less sensitive to interference from other radio signals and radio noises, which mean that the signals are often able to get through environments where a conventional narrow band signal would be impossible to receive and understand. And because a frequency hopping spread-spectrums shifts among multiple channels, it can be extremely difficult for an unauthorized listener to intercept and decode the content of the signal. 802.11a used **OFDM (Orthogonal Frequency Division Multiplexing)** for high data rate while 802.11b uses **HR-DSSS (High Rate Direct Sequence Spread Spectrum).**

Applications

- Internet connection to the notebooks computers through the common central points know as **hotspot**.
- Many Wi-Fi Hotspots re available in restaurants, hotels, libraries and airports.
- It can replace traditional Networks.
- "**WPS**" **(Wi-Fi Positioning System)** is positioning system based on 802.11.

Comparison with other technologies

Broadband Wireless (802.16) and Wi-Fi and both wireless systems but differ in some major ways. The 802.16 is designed to provide service to buildings which are not mobile where as 802.11 deals with the mobility. 802.16 is more elaborate and expensive. So 802.16 have better radios and they can use full duplex communication which 802.11 avoids, keeping the cost down. Similarly **Bluetooth (802.15)** is other technology with works on the master/slave architecture and connects computer to its peripherals. It is finding its application in mobiles and is used as an interface between mobile and computer. It can be inferred that Wi-Fi lies in between Broadband Wireless and Bluetooth in terms of cost and power.

Security system for Wi-Fi

The security system used in Wi-Fi is WEP (Wired Equivalent Privacy) which has either 40 bit or 128 bit key. This is not a robust system. A new protocol called WPA (Wi-Fi Protected Access) solves this problem. Even WPA2 will soon hit the market.

Conclusion

Hence we are able to conclude that Wi-Fi is the definite way to handle Wireless Networking. It is most feasible solution when mobility is the primary concern. It frees network devices from the cable and provides a reliable transfer of the data. Simplicity is one of its attribute that makes it popular. Wi-fi can be of two types ones with the base station and another one without the base station. Although both have got their own applications and advantages, it seems that one that uses base station is more reliable as there is a central control over the data flow. It operates in the unlicensed bit of the broadcast spectrum which helps to make the device cheap. It uses spread spectrum transmission which makes it less prone to interference. Although the interference caused by the system that operate in the same frequency range degrades its performance.

802.11 has the potential to replace many system provided that its security system is better than what is used today (WEP). WPA and WPA2 definitely seem to be the possible solutions. So we can say that, **"The Future is Wireless."** ■

References

1. Andrew S. Tanenbaum, *Computer Networks 4/e*
2. www.howstuffworks.com
3. www.nostarch.com

Google's Advanced Search Operators

Rajendra K Bhatta
2059 Electronics

All of us are very much familiar with the very popular search engine, Google, aren't we? But I bet most of us do not know Google's advanced search features. In addition to the basic boolean operators AND, OR, NOT and quoted strings, Google offers an extensive feature of special syntaxes for honing your searches. Here I have made and attempt to list out some of the advance searching features offered by Google:

Google limits the searches to only the first ten words entered, so keep your search phrases as short as possible. For better results one can use " " marks or + operator.

INTITLE

Tells Google to search for such words found in web page titles. Example: **intitle:nepali celebrity.** For better and accurate results you can use ALLINTITLE instead of INTITLE. (Note: **intitle:nepali intitle:celebrity** is same as **allintitle: nepali celebrity**)

INURL

Tells the Google to look only in URLs of web pages. Example: **inurl:wallpapers.** For better and accurate results you can use ALLINURL instead of INURL.

FILETYPE

There are a number of file types that Google can search for you in addition to standard HTML-formatted pages. Example: Typing **filetype:doc html tutorials** brings back only Microsoft Word documents (i.e files with .doc as extension) on the subject of html tutorials. Similarly for other file types, only files created with the corresponding program are returned. The file types supported by Google are: Adobe Acrobat (pdf), Microsoft Excel (xls), Microsoft PowerPoint (ppt), Rich Text Format (rtf), Shockwave Flash (swf), Text (ans, txt), Images (jpeg, gif)

INTEXT

Tells the Google to search only in the body text of Web pages-not in links, URLs, or titles. Example: **intext:Nepal.** For better n accurate results you can use ALLINTEXT instead of INTEXT.

INANCHOR

Searches for text in a page's link anchors. A link anchor is a descriptive text of a link. For example, the link anchor in the HTML code O'Reilly is "O'Reilly".

SITE

Use this bit of syntax when you want to limit Google's search to a particular site. Example: **site:.com.np wallpapers** would offer results of wallpapers from .com.np sites only not from .com or .org or .edu etc. sites.

DEFINE

This operator can be used to find definitions of a word or phrases. For example: **define:computer** lists out the definition of the word computer.

LINK

Using this operator one can know the particular site being linked by other sites. For example: **link:www.ioe.edu.np** gives the list of the sites which links to the www.ioe.edu.np.

INFO

This operator will present some information that Google has about that web page. For Example: **info:www.google.com** will show information about the Google homepage.

RELATED

Using the related operator will list web pages that are "similar" to a specified web page. Example: **related:www.hotscripts.com**

DATERANGE

This operator is used to limits your search to a particular date or range of dates that a page was indexed. It's important to note that the search is not limited to when a page was created, but when it was indexed by Google. So a page created on February 2 and not indexed by Google until April 11 could be found with daterange: operator search on April 11

CACHE

This syntax causes the Google to display the content from its cache. For Example **cache: www.dc.com.np** gives the homepage of www.dc.com.np from the Google's cache.

PHONEBOOK

The **phonebook** operator searches for U.S. street address and phone number information. For Example: "**phonebook:***John+CA*" will list down all names of person having "*john*" in their names and located in "*California (CA)*"

GOOGLE AS CALCULATOR

Google can be used as a calculator. For Example: **4*2+2** gives the result **10.** Other operators that can be used are –(for subtraction), /(for division), %(for remainder after division), ^(for exponentiation), **th root of** (calculates the nth root of a number Example**: 5th root of 32**), **choose**(X choose Y determines the number of ways of choosing a set of Y elements from a set of X elements Example:**18 choose 4**), **% of** (X % of Y computes X percent of Y Example: **20% of 150**), **!**(for factorial Example: **5!**), **sqrt** (for square root Ex:**sqrt(9)**).Besides these **log( )**, **ln( )**, **sin( ), cos( )**, etc can be used.

Conclusion

Sometimes increase in sophistication in the systems creates new problems. Google, although being a sophicated system in itself, it can still be used by any Ram, Shyam & Ghanshyam on internet to dig out useful information which is normally neither visible nor reachable to anyone.

Finally, I hope these searching techniques will help you a great deal. Remember be optimistic think positive and thank Google for what it has provided the netizens of the World Wide Web. ■

Grid Computing

Rajendra Banjade
2059 Computer

Grid computing is a kind of distributed computing, consisting of many computers operating together remotely and often simply using the idle processor power of normal computers.

The idea of grid in 'grid computing' is analogous to electric power network (grid) where power generators are distributed, but users are able to access electric power without bothering about the source of energy and its location. Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed "autonomous" resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements. Grids are built with low cost modular components, so we can start small and preserve our investment as business grows. For example, one business corporate initially may not be able to afford a mainframe, later as business grows –turning 100 (say) small servers into a giant mainframe, it's fast, cheap and it never breaks. When a server goes down, the system keeps on running.

Computing has taken the grid concept in order to achieve,

- Higher availability
- Improve the resource usage and overall performance of the system
- Ability to add (and remove) resources on demand.

Cluster Computing

Cluster is an independent system, with its own operating system, private memory, and, in some cases, its own file system. Because the processors of one node cannot directly access the memory on the other nodes, programs or software run on clusters usually employ a procedure called "message passing" to get data and execution code from one node to another. Cluster computing is combination of many nodes under one roof.

- Tightly coupled computers
- Single system image
- Centralized Job management & scheduling system

Cluster computing is used for high performance computing, high availability computing.

SETI uses the computational power of over five million home computers to utilize computational power far in excess of even the fastest of supercomputers

Grid Computing

In case of Grids, each node has its own resource manager and does not aim for providing a single system view. Grid computing is combination of many cluster or nodes under the globe i.e. many organizations may be involved. Grid computing utilizes a network of many computers, each accomplishing a portion of an overall task, to achieve a computational result much more quickly than with a single powerful computer.

- Loosely coupled
- No single system image
- Distributed job management & scheduling

It is used for high throughput computing as well as high performance computing depending on the underlying installation setup.

Some usages are molecular modeling for drug design, brain activity analysis, high energy physics etc.

Way to utilize Idle CPU cycle

At any time, some computers may be overloaded, while on the other hand some computer may be displaying screen saver only. If the load is distributed among these computers dynamically, the

performance will be increased. While running some application, a computer can regularly contribute some CPU cycles for a huge load of a grid, like as a micro-hydropower plant contributes to national grid of electricity.

Grid Computing Use - An Example

The most noteworthy project utilizing grid computing is the project SETI@home -'Search for Extra-Terrestrial Intelligence (SETI)'. SETI uses the computational power of over five million home computers to utilize computational power far in excess of even the fastest of supercomputers. SETI@home makes available a free piece of software, a home user may install on a computer. The software runs when the computer is left idle, and each computer with the software contacts a central server in Berkeley and downloads a file (some KB to MB) which tells it what to analyze and sends the result obtained to the server. If the computer is busy with another task, it uses less else fully utilizes the CPU cycles in well mannered way. The grid computing system then analyzes this data for specific patterns. It would normally cost millions of dollars to achieve that type of power on one or even two supercomputers. Another example is 'Human Proteome Folding' - research for cancer and other diseases, which divides the cDNA and genome data and distributes the computation of them for many computers in grid via Globus toolkit which is one of grid middleware.

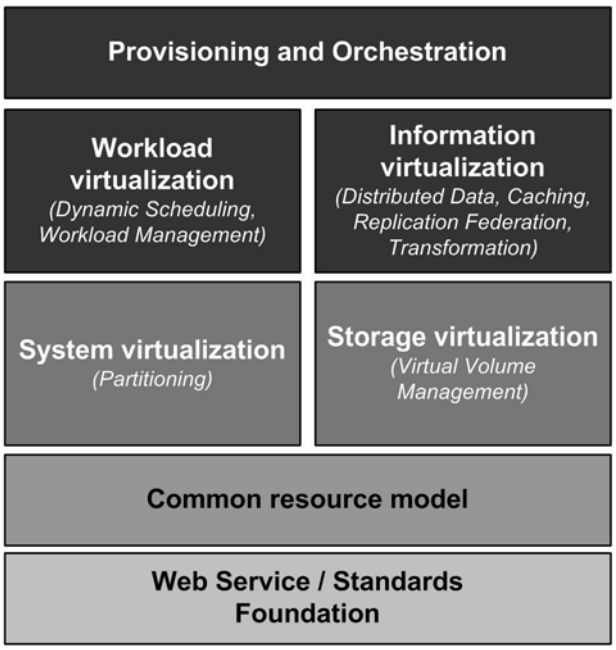
Database Designed for grid computing

Oracle database 10g, the first relational database designed for Grid Computing, lowers the cost of ownership through automated management while providing the highest possible quality of service. Availability and scalability with grid computing, Oracle database 10g delivers the response times on users demand and reduces cost of downtime. Oracle offers non-stop availability, scalability, and low-cost clustering with Oracle Real Application Clusters (RAC), the foundation of grid computing. With 10g, we can simply plug another server into the cluster. And if do not need that server, simply unplug it. With a grid, if a component of the grid goes down, we do not lose any availability.

Programming difficulties for GC

Grid network is in collaboration with colleagues from around the globe, it is important to choose grid technologies that support and work on a wide variety of resources, which are heterogeneous in terms of various factors including architecture, instruction set, configuration, node operating system, and local resource managers. Several Grid middleware software systems are available for building high-end computing grids and their applications. A program for the grid computing system is the most important resource among all, than like hardware. ■

"The whole is greater than the sum of the parts".



Components of grid computing

Computer Professional

Q. What's the difference between an amateur programmer and a professional?

A. An amateur thinks that 1 kilobyte is 1000 bytes. A professional thinks that 1 kilometer is 1024 meters.

Internet

What makes it possible?

Subharoj Dahal
Network Engineer
CIT, IOE

The correct way to imagine the Internet is not a hierarchical model like a company with divisions, or even a football team with a coach and players. Instead, it is a loose confederation like the United Nations, where each member has full authority over their own "territory," although they may work together. Since it's beginning in 1969, the Internet has grown from four host computer systems to tens of millions. However, just because nobody owns the Internet, it doesn't mean it is not monitored and maintained in different ways. The Internet Society, a non-profit group established in 1992, oversees the formation of the policies and protocols that define how we use and interact with the Internet.

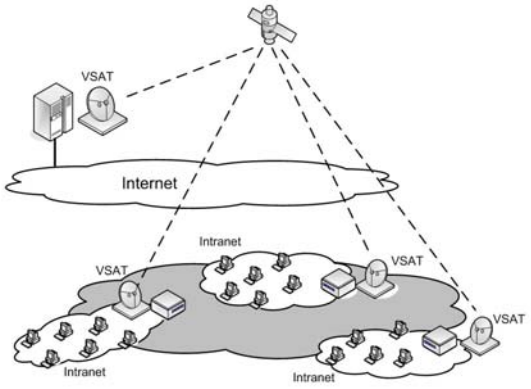


Fig. 1: Internet topology

Internet is collection of individual network having independent functionalities and resources and is governed by set of rules and guidelines, called **Protocol**, so that one computer in the one network can properly and reliably communicate with other computer in any other network. The main objective of the Internet is to share resources among the computer even though they are geographically far from each other. Internet means not only downloading web pages from web sites and checking your e-mail rather it can be put to use for paying your telephone bills, buying a flat in New York, making a call to friend in London, switching on your lights in drawing room from your office using your laptop and many more.

To know process behind downloading a typical page from a remote computer to your machine, some key terminology need to be explained:

Internet is collection of individual network having independent functionalities and resources and is governed by set of rules and guidelines

Domain Name: The unique name that identifies a web site and it always has two or more parts, separated by dots. The part on the left is the most specific, and the part on the right is the most general. A given machine may have more than one Domain Name but a given Domain Name points to only one machine. For example yahoomail.com, mail.yahoo.com points to same machine (computer).

IP Address: Computer or any computing device works in terms of binary number system so every computer connected to the Internet is assigned a unique number known as an Internet Protocol (IP) address. Since these numbers are usually hierarchically arranged, any machine having specific IP address can be monitored easily.

URL: The World Wide Web address of a web site in the Internet. eg: www.hotmail.com

DNS Server: It is the software, which is used to map domain name of websites to their corresponding IP address. eg: www.ioe.edu.np => 202.10.45.213

Web Browser: It is the software running on local machine, which helps to view the information from web sites.

Web Server: The software running in remote machine, which is used to response to the request messages originated by web browser.

Router: It is a hardware device, which is used to deliver data packets from one networks of computer to remote network as shown in fig 2.

When you hit URL like www.hotmail.com in address bar of your web browser (like Internet Explorer) then your web browser will encapsulate the specified URL (address) with necessary headers (extra information) and send it to nearest DNS server. Then DNS server will send IP address corresponding to specified domain name you have type in your web browser. Again your web browser will make a data packet, which consists of IP address of source host, destination host, request message and other necessary information and deliver it into the network. The collection of router in the Internet will forward that data packet to appropriate destination according to the IP address of remote machine. When the request packet will reach in other side (remote side), web server, which is listening in close loop whether any request packet are there or not, will catch it and process accordingly. If it finds one it will respond to that request by delivering requested information. The next job of web server is fetching requested file from file system and preparing response data packet, which will be delivered in the network. The delivered data packets consist of IP address of the client (the machine that originates the request) machine, IP address of server machine and other fields as well. Now the delivered response data packets are in the hand of router, which will eventually, delivered to client machine. Web Browser in the client machine displays the received message in front of you, which may be the log in page of www.hotmail.com

This is how web pages are retrieved from its source to when ever and where ever you want it to. ■

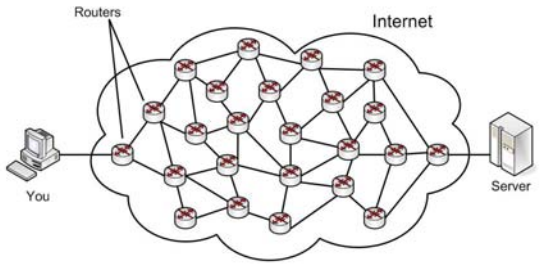
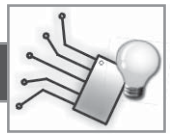
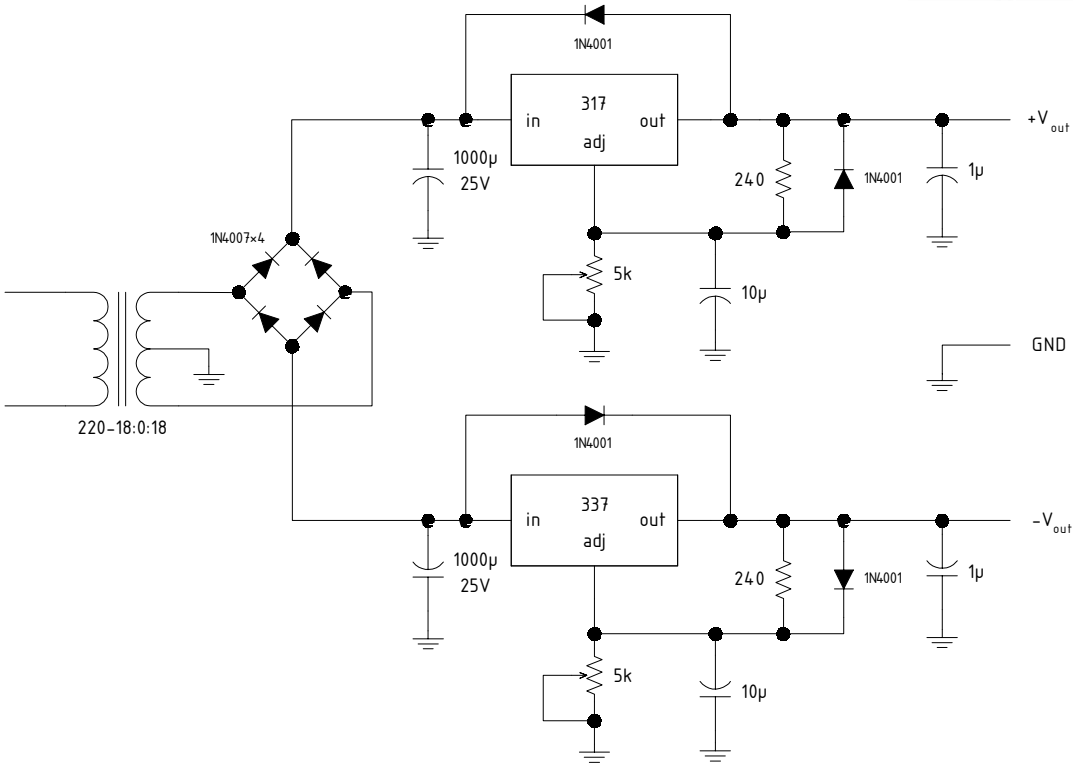


Fig. 2: Inter-connected routers.

circuit ideas





a simple regulated variable dual dc supply (1.2V-18V, 1 A max)

Online or Invisible ?

Rajendra Bahadur Thapa
2060 Electronics

Articles freely available online are more highly cited. For greater impact and faster scientific progress, authors and publishers should aim to make research easy to access. The volume of scientific literature typically far exceeds the ability of scientists to identify and utilize all relevant information in their research. Improvements to the accessibility of scientific literature, allowing scientists to locate more relevant research within a given time, have the potential to dramatically improve communication and progress in science. With the web, scientists now have very convenient access to an increasing amount of literature that previously required trips to the library, inter-library loan delays, or substantial effort in locating the source. Evidence shows that usage increases when access is more convenient, and maximizing the usage of the scientific record benefits all of society.

Although availability varies greatly by discipline, over a million research articles are freely available on the web. Some journals and conferences provide free access online, others allow authors to post articles on the web, and others allow authors to purchase the right to post their articles on the web.

This article investigates the impact of free online availability by analyzing citation rates. It do not discuss the methods of creating free online availability, such as time-delayed release or publication/membership/conference charges. Online availability of an article may not be expected to greatly improve access and impact by itself. For example, efficient means of locating articles via web search engines or specialized search services is required, and a substantial percentage of the literature needs to be indexed by these search services before it is worthwhile for many scientists to use them. Computer science is a forerunner in web availability -- a substantial percentage of the literature is online and available through search engines such as Google (google.com), or specialized services such as Research Index (researchindex.org). Even so, the greatest impact of the online availability of computer science literature is likely yet to come, because comprehensive search services and more powerful search methods have only become available recently.

An analysis was performed on 119,924 conference articles in computer science and related disciplines, obtained from DBLP (dblp.uni-trier.de). In computer science, conference articles are typically formal publications and are often more prestigious than journal articles, with acceptance rates at some conferences below 10%. Citation counts and online availability were estimated using Research Index. The analysis excludes self-citations, where a citation is considered to be a self-citation if one or more of the citing and cited authors match.

Figure 1 shows the probability that an article is freely available online as a function of the number of citations to the article, and the year of publication of the article. The results are dramatic. There is a clear correlation between the number of times an article is

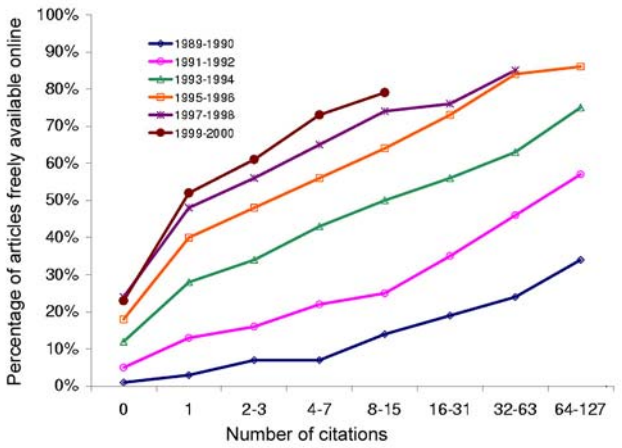


Figure 1: Probability that an article is freely available online against number of citations to the article

cited, and the probability that the article is online. More highly cited articles, and more recent articles, are significantly more likely to be online.

More highly cited articles, and more recent articles, are substantially more likely to be freely available on the web. The actual percentage of articles available online is greater due to limitations in the extraction of article information from online documents, and limitations in locating articles on the web. Only points with greater than 100 articles are computed.

The mean number of citations to offline articles is 2.74, and the mean number of citations to online articles is 7.03, or 2.6 times greater than the number for offline articles. These numbers mask variations over time -- in particular, older articles have more citations on average, and older articles are less likely to be online. When considering articles within each year, and averaging across all years from 1990 to 2000, it is found that online articles are cited 4.5 times more often than offline articles.

Differences within each publication venue was also analysed, where multiple years for the same conference are considered as separate venues. A percentage increase in the average number of citations to online articles compared to offline articles was noticed. When offline articles were more highly cited, negative of the percentage increase for offline articles was used. For example, if the average number of citations for offline articles is 2, and the average for online articles is 4, the percentage increase would be 100%. For the opposite situation, the percentage increase would be -100%. Figure 2 shows the results. Averaging the percentage increase across 1,494 venues containing at least five offline and five online articles results in an average of 336% more citations to online articles compared to offline articles published in the same venue [the first, second (median), and third quartiles of the distribution are 58%, 158%, and 361%].

The graph shows the distribution of the percentage increase for the average number of citations to online articles compared to offline articles. The analysis covers 1,494 publication venues containing at least 5 online and 5 offline articles. For 90% of venues, online articles are more highly cited on average. On average there are 336% more citations to online articles compared to offline articles published in the same venue [the first, second (median), and third quartiles of the distribution are 58%, 158%, and 361%].

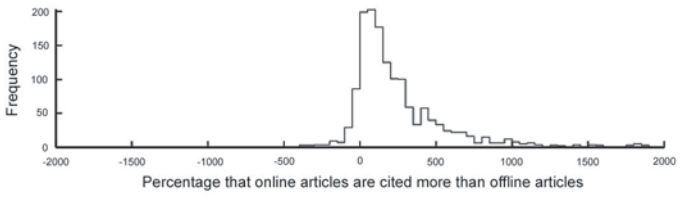


Figure 2:Analysis of citation rates within publication venues.

The preceding data does not allow us to make conclusions as to the cause of the correlation between high citation rates and online availability. Online articles may be more highly cited because they are easier to access and thus more visible and more likely to be read, or because higher quality articles are more likely to be made available online. Intuitively, it seems likely that the easier availability and improved visibility of online articles plays a significant role. If it is assumed that articles published in the same venue are of similar quality, then the analysis by venue suggests that online articles are more highly cited because of their easier availability. This assumption is likely to be more valid for top-tier conferences with very high acceptance standards. Restricting the above analysis to the top publication venues by average citation rate results in a similarly dramatic increase in citation rates for online articles. For example, when restricting to the top 20 venues, the average increase in the citation rate for online articles is 286% [the first, second (median), and third quartiles of the distribution are 66%, 284%, and 471%].

There is a clear correlation between the number of times an article is cited, and the probability that the article is online.

Free online availability facilitates access in multiple ways, including online archives, direct connections between scientists or research groups, hassle-free links from email, discussion groups, and other services, indexing by web search engines, and the creation of third-party search services. Free online availability of scientific literature offers substantial benefits to science and society. To maximize impact, minimize redundancy, and speed scientific progress, author and publishers should aim to make research easy to access. ■

References

1. Steve Lawrence, C. Lee Giles, Kurt Bollacker, *Digital Libraries and Autonomous Citation Indexing*, IEEE Computer, 32(6):67-71, 1999.
2. A. Odlyzko, *The Rapid Evolution of Scholarly Communication*, Learned Publishing, 2001.

An Overview of Sniffing

Roshan Sharma
2060 Electronics

What is Sniffing?

Sniffing is a method by which an attacker can compromise the security of a network. A sniffer, in network security, is a program or tool that monitors a computer network for key information such as authentication information (username and passwords) which can be used to gain access to system or resource.

How does it Work?

These days computers are connected by switches. Rather than distribute network traffic to all ends of the network (as done anciently), switches filter traffic at a hub. This prevents the computer from seeing anybody else's traffic even when it puts the adapter into "promiscuous mode" (disabling filtering of traffic and allowing all the traffic to pass through the network is the promiscuous mode). So, an attacker must actively attack the switch/router in order to redirect traffic flow. When network traffic enters the machine, it is first handled by Ethernet driver. The driver then passes the traffic to the Transmission control protocol/Internet protocol (TCP/IP) stack, which will in turn pass it to applications. Sniffing software connects directly to the Ethernet driver, making a copy of it.

Let's now see some of the useful port numbers and their functions before entering into depth for easy understanding of topics that are described later. Only few ports are described due to lack of writing space.

i) rexec (Port 512)

The rexec service, otherwise called "rexecd" on almost all UNIX based operating systems is a legacy used for executing commands remotely. The service performs authentication (username and passwords) information passed to the server by a client. The service receives a buffer from the client consisting of the necessary data. If authentication was successful, a NULL byte is returned by the server; otherwise, a value of 1 is returned in addition to an error string. So, it's useful to monitor this port for sniffing purpose.

ii) rlogin (Port 513)

The rlogin protocol provides much the same functionality as the Telnet protocol additionally combined with the authentication mechanism of the REXEC protocol, with some exceptions. It supports trust relationship, which is specified via a file called "rhosts" in the user's home directory. This file contains a listing of users and the hosts on which they reside, who are allowed to log into the specified account without a password. Authentication is performed instead by trusting that the user is who the remote rlogin client says he or she is. Sad but true this authentication mechanism works only among UNIX systems, and can be flawed in many ways. So it's useful to monitor this port for sniffing purpose.

iii) POP (Port 110)

The Post Office Protocol (POP service is a network server by which client-based e-mail programs are connected to access a user's e-mail on a central server. POP servers appear commonly on an Internet service providers (ISP's) network, to provide e-mail delivery to customers. POP traffic is often not encrypted, even if encrypted it can be decoded. So, it is also very useful to monitor this port for sniffing purpose.

iv) X11 (Port 6000+)

The X11 Window system uses a "magic cookie" to perform authorization against clients attempting to connect to a server. A randomly generated 128-bit cookie is sent by X11 clients when connecting to the X Window server. By sniffing this cookie, an attacker can use it to connect to the same X Window server. Normally, this cookie is stored in a file named ".Xauthority" within a user's home directory. This cookie is passed to the X Window server by the xdm program at logon.

Windows NT Authentication:

The use of weak Windows NT authentication mechanism creates one of the weakest links in Windows NT security. Password hashes are encrypted in a weak fashion so that original hash can be sniffed from the network and cracked quite easily. However, NTLMv2 which was introduced with the release of Service Pack 4 for Windows NT 4.0 uses comparatively stronger password hashes encryption though it can also be cracked.

Windows NT stores the LANMAN and Administrator password hashes in the file called SAM

(security account manager). The SAM file cannot be accessed while Windows NT is running. So, one requires to simultaneously boot the machine and access the SAM file.

Various Windows NT password crackers are available for free download in the web. So after accessing the SAM, it can be used to crack the password hashes. Usually the method used to crack the hashes is the dictionary attack followed by brute force attack. This type of attack can crack any type of password. The only thing required is patience as the brute force attack can take several days to complete its task. Though weak passwords can be cracked within a minute too.

Popular Sniffing software:

There are various sniffing software made available in the net. But the ones which I personally think to be useful are listed below:

i) Wild Packet
ii) TCP Dump
iii) Ettercap
iv) Sniffit
v) Dsniff
vi) Carnivour

Due to lack of writing space, I am going to deal only Dsniff in some detail.

Dsniff:

Dsniff is a sniffing toolkit provided by Dug Song. Dsniff is most famous for its authentication (usernames, passwords) sniffing capabilities. The current version of dsniff will decode authentication information for the following protocols: AOL Instant Messenger, Citrix Winframe, Concurrent Versions System (CVS), FTP, HTTP, ICQ, IMAP, Internet Relay Chat (IRC), Lightweight Directory Access Protocol (LDAP), RPC mount requests, Napster, NNTP, Oracle SQL Net, Open Shortest Path First (OSPF), PC Anywhere, POP, PostgreSQL, Routing Information Protocol (RIP), Remote Login (rlogin), Windows NT plaintext (SMB), Network Associates Sniffer Pro (remote), Simple Network Management Protocol (SNMP), Socks, Telnet, X11, and RPC yppasswd.

With today's switched networks and encrypted protocols, password sniffing doesn't always work as well as we might hope. Dsniff contains several redirect and man-in-middle (MITM) utilities to redirect the flow of traffic and decrypt sessions.

The first utility is "arpspoof". Address Resolution Protocol (ARP) is used by hosts to find the local router's Media Access Control (MAC) address. By spoofing ARP packets, you can convince other nearby computers that you are the router. Your machine has to forward them onto the legitimate router after receiving them, but in the meantime, the dsniff password sniffer has a chance to process the packets. This runs well not only on local switched networks, but also cable-modem networks. This tool isn't completely foolproof; you are essentially fighting with the router, trying to convince other machines of the local MAC address. As a result, traffic flows through your machine are sometimes intermittent. This technique is easily detected by networks-based intrusion detection systems (IDSs).

a program that monitors a computer network for key information such as authentication information

The "dnsspoof" utility is another way of redirecting traffic. In this case, it spoofs responses from the local Domain Name System (DNS) server. When you go to a website such as www.example.com, your machine sends out a request to your local DNS server asking for the IP address of www.example.com. This usually takes a while to resolve; dnsspoof quickly sends its own response faster. The victim will take the first response and ignore the second one. The spoofed response will contain a different IP address than the legitimate response, usually the IP address of the attacker's machine. The attacker will likely be using one of the other dsniff man-in-the-middle utilities.

The name man-in-the-middle comes from cryptography and describes the situation when somebody intercepts communications, alters it, and then forwards it. The Dsniff utilities for these attacks are "webmitm" for HTTP traffic (including SSL) and "sshmitm" (for SSH). Normally, SSH and SSL are thought to be secure, encrypted protocols that cannot be sniffed. The way the MITM utilities work is that they present their own encryption keys to the SSL/SSH clients. This allows them to decrypt the traffic, sniff passwords, and then reencrypt with the original server keys.

Dsniff can sniff not only passwords, but also other cleartext traffic. The "mailsnarf" utility sniffs e-mails, reassembles them into an mbox format that can be read by most mail readers. The "msgsnarf" utility sniffs message from ICQ, IRC, Yahoo! Messenger and AOL IM. The "filesnarf" utility sniffs files transferred via NFS (network file system, a popular fileserver protocol used on UNIX systems). The "urlsnarf" utility saves all the URLs it sees going across the wire. The "webspy" utility sends those URLs to a Netscape Web browser in real time-essentially allowing to watch in real time what the victim sees on their Web browser.

The "macof" utility sends out a flood of MAC address. This is intended as another way of attacking Ethernet switches. Most switches have limited tables that can hold only 4000 MAC address. When the switch overloads, it "fails open" and starts repeating every packet out every port, allowing everyone's traffic to be sniffed.

The "tcpkill" utility kills TCP connections. It can be used as a denial of service (DoS) attack. For example, it can be used to kill every TCP connection made by other persons. The "tcpnice" utility is similar to tcpkill, but rather than killing connections, it slows them down. For example, you could spoof from your neighbor's cable modems so that you can get a higher percentage of the bandwidth for your downloads.

Taking Protective Measures

So you probably think that all is lost and that there is nothing you can do to prevent sniffing from occurring on your network, right? All is not lost, as you will see in this section.

Secure Shell (SSH):

Secure Shell is a cryptographically secure replacement for the standard Telnet, rlogin, rsh, and rcp commands. It consists of both client and server that use public key cryptography to provide session encryption. It also provides the ability to forward arbitrary ports over an encrypted connection, which comes in very handy for forwarding of X11 Window and other connections. SSH has received wide acceptance as the secure mechanism to access a remote system interactively. The original SSH, written by Tatu Ylonen, is available from ftp://ftp.cs.hut.fi/pub/ssh/. A completely free version of SSH-compatible software, OpenSSH, developed by the OpenBSD operating system project can be obtained from www.openssh.com.

If you can't use encryption on your network for some reason, then you must rely on detecting any network interface card (NIC) that may be operating in a manner that could be invoked by a sniffer.

Local detection can be done by using the "ifconfig" command on any UNIX system. But it is important to note that if an attacker has compromised the security of the host on which you run this command, he or she can easily change the output of this command. An important part of an attacker's toolkit is a replacement "ifconfig" command that does not report interfaces in promiscuous mode.

DNS Lookups:

Most programs that are written to monitor the network perform reverse DNS lookups when they produce output consisting of the source and destination hosts involved in a network connection. In the process of performing this lookup, additional network traffic is generated; mainly, the DNS query to look up the network address. It is possible to monitor the network for hosts that are performing a large number of address lookups alone; however, this may be coincidental, and not lead to a sniffing host.

An easier way, which would result in 100% accuracy, would be to generate a false network connection from an address that has no business being on the local network. We would then monitor the network for DNS queries that attempt to resolve the faked address, giving away the sniffing host.

Finally, before putting down the pen, I would like to say that if you want to be a good sniffing man and want to write a good sniffer, then you've got to know the ups and downs of each and every port. Above all the most important part which is fully required is the *keen interest* in sniffing purpose. ■

Artificial *lack of* Intelligence

Om Chandra Rimal
2059 Computer

Artificial Intelligence (AI), is the science of making computers do things that may require intelligence when done by humans. Alternate definition of AI takes it as the intelligence exhibited by an artificial entity and such a system is usually called a computer. As for the term - "Artificial Intelligence", it was coined by John McCarthy in 1956.

AI has had some success in limited, or simplified domains. However, the five decades since the inception of AI has brought only very slow progress, and early optimism concerning the attainment of human-level intelligence has given way to an appreciation of the profound difficulty of the problem.

What is Intelligence ?

There is no proper definition of the term - "Intelligence". However, the components of intelligence - learning, reasoning, problem-solving, perception, and language-understanding have been chief areas of research in AI. So, they are:

Learning

There are generally two types of learning - trial-and-error learning and rote learning. Trial-and-error learning is improving by previous experience. While, rote learning is learning by memorizing. It is simpler to implement rote learning to the computers.

Reasoning

To reason is to draw inferences appropriate to the situation in hand. Inferences are classified as either deductive or inductive. There has been considerable success in programming computers to draw inferences, especially deductive inferences. One of the hardest problems confronting AI is that of giving computers the ability to distinguish the relevant from the irrelevant.

Problem - solving

Problem-solving methods divide into special-purpose and general-purpose. A special-purpose method is tailor-made for a particular problem, and often exploits very specific features of the situation in which the problem is embedded. A general-purpose method is applicable to a wide range of different problems. Problems have the general form: given such-and-such data, find x. A huge variety of types of problem is addressed in AI. Some examples are: finding winning moves in board games, identifying people from their photographs, etc.

Perception

There are successful implementations and there are lapses. Lapses are where human have failed to fulfill the lack of intelligence in machines.

In perception the environment is scanned by means of various sense-organs, real or artificial, and processes internal to the perceiver analyses the scene into objects and their features and relationships. Analysis is complicated by the fact that one and the same object may present many different appearances on different occasions, depending on the angle from which it is viewed, whether or not parts of it are projecting shadows, and so forth. At present, artificial perception is sufficiently well advanced to enable a self-controlled car-like device to drive at moderate speeds on the open road and a mobile robot to roam around busy offices searching for and clearing away empty soda cans.

Understanding languages

A language is a system of signs having meaning by convention. Traffic signs, for example, form a mini-language. An important characteristic of full-fledged human languages, such as English, which distinguishes them from, bird calls and systems of traffic signs, is their productivity. A productive language is one that is rich enough to enable an unlimited number of different sentences to be formulated within it. It is relatively easy to write computer programs that are able, in severely re-

stricted contexts, to respond in English, seemingly fluently, to questions and statements. An appropriately programmed computer can use language without understanding it, in principle even to the point where the computer's linguistic behaviour is indistinguishable from that of a native human speaker of the language. What, then, is involved in genuine understanding if a computer, that uses language indistinguishably from a native human speaker, does not necessarily understand? There is no universally agreed answer to this difficult question.

There are successful implementations and there are lapses. Lapses are where human have failed to fulfil the lack of intelligence in machines. But, nevertheless, we are trying to remove the "Artificial Lack of Intelligence".

What is AI ?

AI can generally be divided into two schools of thoughts - Conventional AI (movies have mostly focused on it) and Computational Intelligence (CI) (implemented in making movies).

Conventional AI

mostly involves methods now classified as machine learning, characterized by formalism and statistical analysis. This is also known as symbolic AI, logical AI, neat AI and Good Old Fashioned Artificial Intelligence (GOFAI). Conventional methods include: Case Based Reasoning, Bayesian Networks, Behaviour Based AI and also:

Expert Systems: Systems that can apply reasoning capabilities to reach a conclusion. An expert system can process large amounts of known information and provide conclusions based on them.

Computational Intelligence (CI)

involves iterative development or learning (e.g. parameter tuning in connectionist systems). Learning is based on empirical data and is associated with non-symbolic AI, scruffy AI and soft computing. CI methods mainly include:

Neural networks: Systems with very strong pattern recognition capabilities.

Fuzzy systems: Techniques for reasoning under uncertainty and has been widely used in modern industrial and consumer product control systems.

Evolutionary computation: Applies biologically inspired concepts such as population, mutation and survival of the fittest to generate increasingly better solutions to a problem. These methods most notably divide into evolutionary algorithms (e.g. genetic algorithms) and swarm intelligence (e.g. ant algorithms).

With hybrid intelligent systems, attempts are made to combine these two groups. Expert inference rules can be generated through neural network or production rules from statistical learning.

Artificial Lack of Intelligence

It has taken half a century for us to reveal the complexity of AI. It was already 50 years when the DeepBlue defeated Gary Kasparov in 11th May, 1997 and he said, "Sometimes quantity becomes quality." It took 512 RISC processors, which could examine 200 million board positions each second to defeat him. It would have taken 66 million seconds for Kasparov and since chess players get an average of 3 minutes to make a move, he would have to stare at the board for 360 years for DeepBlue's 3 minutes. Still, we are unaware of the lack of intelligence in machines. AI began with a thought to create intelligent entities in about a decade as is shown in sci-fi movies. We should have millions of Terminators around us by now. But, thanks to improper researches and the unrealization of "How machine and man are similar and are different?". Thanks to the 'artificial' assumption - "Researches are on, we will be successful."

Let's hope that the "Artificial Lack of Intelligence" in machines is diminishing. Let's work on helping those who are involved in removing "A Lack of I", those who are researching on AI. Let's hope to have more ASIMOs and not Terminators. ■

References

1. Jack Copeland, *What is AI?*, AlanTuring.net
2. *Artificial Intelligence*, en.wikipedia.org
3. *Lectures by Mr. Arun Timalsina, Faculty, DOECE* (The DeepBlue part).

"640K ought to be enough for anybody."

-- Bill Gates, 1981"

Cryptography: *An Essence*

Prajwol Kumar Nakarmi
Nirmal Thapa
2060 Computer

Introduction

Make any enquiry about computer security, and you will almost immediately fall over the terms *cryptography* and *encryption* (and *decryption*). This article aims to give fundamental concept, along with some practical methods of cryptology.

Cryptography is all about the study or analysis of codes and coding methods. In cryptography, **Encryption** is the process of obscuring information to make it unreadable without special knowledge. By this, we mean a process of converting information to a disguised form in order to send it across a potentially unsafe channel. The reverse process is called **Decryption.**

In today's information society, cryptography has become one of the main tools for privacy, trust, access control, electronic payments, corporate security, and countless other fields such as Internet e-commerce, mobile telephone networks and bank automatic teller machines etc. The use of cryptography is no longer a privilege reserved for governments and highly skilled specialists, but is becoming available for everyone. Most forms of cryptography in use these days rely on computers, simply because a human-based code is too easy for a computer to crack.

*<The most famous encryption machine invented was the '**Enigma**', used in the Second World War to send military messages.>*

Brief History

Cryptography is one of the oldest fields of technical study we can find records of, going back at least 4000 years. This article does not concentrate on history of cryptography, however, the name "**Claude Elwood Shannon**" is worth mentioning. He was one of the first modern cryptographers to attribute advanced mathematical techniques to the science of ciphers. In cryptography, **confusion** and **diffusion** are two properties of the operation of a secure cipher, which were identified by Shannon, and they form the basis for many modern cryptosystems because they tend to increase the workload of cryptanalysis.

Getting started

One of the best examples of early cryptography is the *Caesar cipher*, named after **Julius Caesar** because he is thought to have used it even if he did not actually invent it. It works by simply shifting the value of a letter either to right or to the left. Here is a slightly modified versionof the Caesar cipher, implemented in C++. Given functions are complementary i.e. they can be interchanged.

```
void Encipher ( int_8 string[ ],
   int_8 offset ){
for( int i=-1; string[++i]!='\0';
   string[i]+=offset );
}

void Decipher ( int_8 string[ ],
   int_8 offset ){
for( int i=-1; string[++i]!='\0';
   string[i]-=offset );
}
```

If you think about it, the Caesar cipher is not very brilliant. An *offset* of zero means that you do not actually encrypt the plain message. It is remarkably simple to break — in fact; most people can break the code in less than a minute.

Getting slightly more complex

In his issue of C/C++ User's Journal, **William Ward** presented an excellent encryption technique that uses **XOR**ing. Let's write a simple C++ implementation of it. Note that the same function works both for enciphering as well as deciphering.

```
void ProcessString ( int_8 str[ ],
   int_8 k_str[ ] ){
for ( int i=0, key_i=-1;
   str[i]!='\0'; ){
     if ( k_str[++key_i]!='\0' ) {
     str[i]=str[i++] ^ k_str[key_i];
     continue;
     }
     key_i=-1;
     }
}
```

It is important to note that, nowadays, almost all ciphers use XORing technique at least somewhere in their program. The primary difference between the simple encryption program shown here and an encryption program that provides true and strong encryption is in the algorithm used by the program or the generation of the keystream(*k_str*). Here's what that means, conceptually: Let's assume that the keystream is a simple cycling value: 1, 2, 3, 4,…, 255, 1, 2, 3, and so on. For any cryptanalyst, breaking the keystream under these circumstances is a piece of cake – they will begin to see repetition in the code quite quickly. Given a document of 10,000 characters or so, a good cryptographer can break the keystream in minutes.

Why XOR?

If ADD, or so, is used instead, extra effort should be employed to handle the overflowed values. i.e. 8-bits processed by ADD may not always produce 8-bits By that, restoring 8-bits may corrupt the original values. However, 8-bits processed by XOR is always 8-bits and restoring does not corrupt the original values. Moreover, [p XOR k = c] means that [c XOR k = p], which is always true unlike OR, AND, NOR and so on.

Cryptography, thus, cannot guarantee 100% security. But we can work toward 100% risk acceptance

Moving to Real World

Real world cryptosystems are really very complex. Throughout the past, we have seen so many algorithms rise and collapse. "***NO algorithm that depends on the secrecy of the algorithm, itself, is secure. For professionals, it is easy to disassemble and reverse-engineer the algorithm. Experience has shown that the vast majority of secret algorithms that have become public knowledge later have been pitifully weak in reality***."

In this article, we present brief discussion of two selected algorithms that have withstood decades of academic scrutiny and still are considered non-flawed. Generally speaking, modern encryption algorithms come in two flavors, symmetric and asymmetric. Symmetric algorithms such as Blowfish, CAST-128, IDEA, Rabbit, RC4 etc. use the same key for encryption and decryption. Asymmetric algorithms such as RSA, XTR, LUC, Elliptic Curve etc. use two keys – one for encryption and the other for decryption. Sounds strange? Keep reading!

Private Key (Secret Key / Symmetric Key) Ciphers

This type of cipher uses the same key for both encryption and decryption i.e. one is easily derivable from the other (hence called symmetric). As such, the key with which plain text has been encrypted should be kept secret (hence called secret/private), because it is all that is needed to decrypt the cipher-text.

Public Key (Asymmetric Key) Ciphers

Public-Key Cryptography is a form of modern cryptography, which allows users to communicate securely without previously agreeing on a shared secret key. It uses a combination of a **private** key and a **public** key. The private key is known only to your computer, while the public key is given by your computer to any computer (even the rival) that wants to communicate securely with it. With the public key, one could encrypt messages, and decrypt them with the private key. Thus, the owner of the private key would be the only one who could decrypt the messages, but anyone knowing the public key could send them in privacy.

Blowfish

It was **Bruce Schneier**, who designed Blowfish as a general-purpose algorithm, intended as a replacement for the aging DES and free of the problems associated with other algorithms. At the time, many other designs were proprietary, encumbered by patents or kept as government secrets. Schneier has stated that, "Blowfish is unpatented, and will remain so in all countries. The algorithm is hereby placed in the **public domain**, and can be freely used by anyone."

Blowfish is a symmetric encryption algorithm, meaning that it uses the same secret key to both encrypt and decrypt messages. Blowfish is also a block cipher, meaning that it divides a message up into fixed length blocks during encryption and decryption. The block length for Blowfish is 64 bits; messages that are not a multiple of eight bytes in size must be padded. The key can be up to 448 bits.

Algorithm Itself:

Encryption:

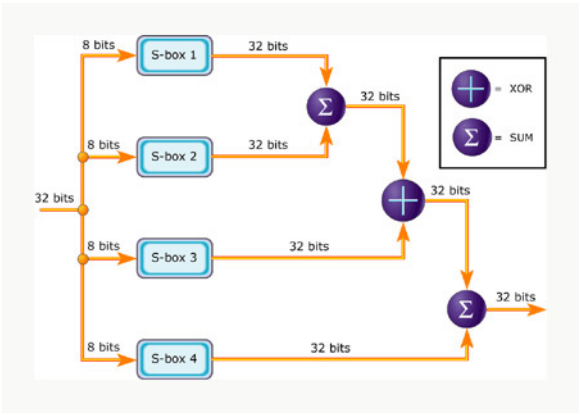
A 64-bit plaintext message is first divided into 32 bits. The "left" 32 bits are XORed with the first element of a P-array to create a value we'll call P', run through a transformation function called F, then XORed with the "right" 32 bits of the message to produce a new value we'll call F'. F' then replaces the "left" half of the message and P' replaces the "right" half, and the process is repeated 15 more times with successive members of the P-array. The resulting P' and F' are then XORed with the last two entries in the P-array (entries 17 and 18), and recombined to produce the 64-bit cipher text.

Decryption:

Blowfish cipher is so well designed, that the encryption algorithm is identical to the decryption algorithm step by step in the same order, only with the sub-keys applied in the reverse order. i.e. P1, P2, …, P18 are used in the reversed order.

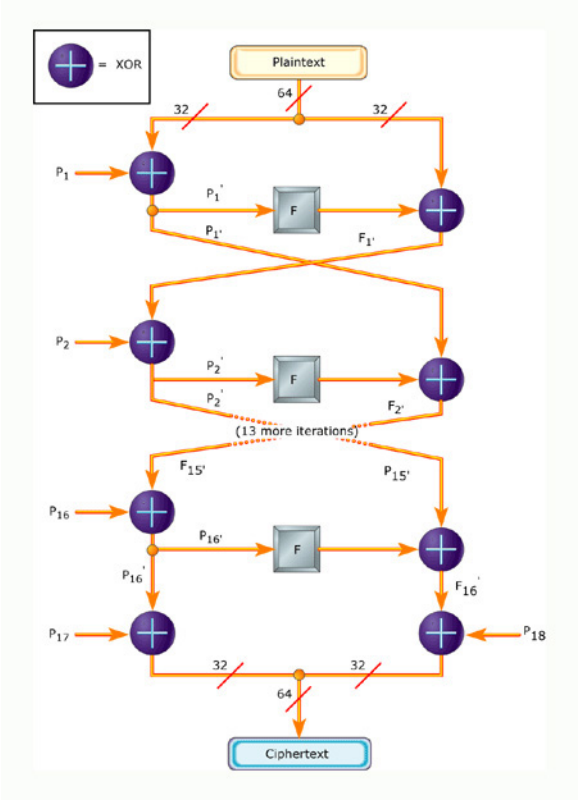
The round function F (A):

A graphical representation of F appears below. The function divides a 32-bit input into four bytes and uses those as indices into an S-array. The lookup results are then added and XORed together to produce the output.



Blowfish Key Schedule (Sub-Keys Generation)

The P-array and S-Box values used by Blowfish are pre-computed based on the user's key. Brief summary of the procedure is as follows:



- P is an array of eighteen 32-bit integers
- S is a two-dimensional array of 32-bit integer of dimension 4×256.
- Both arrays are initialized with constants, which happen to be the hexadecimal digits of ð (a pretty decent random number source).
- The key is divided up into 32-bit blocks and XORed with the initial elements of the P and S arrays. The results are written back into the array.
- A message of all zeros is encrypted; the results of the encryption are written back to the P and S arrays. The P and S arrays are now ready for use.

Security Concern:

With regard to brute force attacks, Blowfish is virtually invulnerable with suitable choice of key length, which can be as long as 448 bits. Blowfish is also impressive fast to execute. Following table, developed by Schneier, compares the number of clock cycles required on a Pentium for various algorithms implemented in C. Blowfish is clearly the fastest to execute.

Algorithm	# of clock cycles per byte encrypted
Blowfish	18
RC5	23
DES	45
IDEA	50
Triple-DES	108

Blowfish Conclusion

Blowfish encrypts 64-bits blocks of plaintext into 64-bit blocks of cipher text. Blowfish is implemented in numerous products and has received a fair amount of scrutiny. So far, the security of Blowfish is unchallenged. The S-boxes in Blowfish are key dependent. Both the sub keys and S-boxes are produced by a process of repeated applications of Blowfish itself. This thoroughly mangles the bits and makes cryptanalysis very difficult. So far, there have been a few published papers on Blowfish cryptanalysis, but no practical weaknesses have been found.

The only known attacks against Blowfish are based on its weak key classes. It has gained a fair amount of acceptance in a number of applications, including Nautilus and PGPfone. The blowfish algorithm was first introduce in 1993, and has not been cracked yet. It is also noteworthy to point out that this algorithm can be optimized in hardware applications, although it, like most other ciphers, is often used in software applications.

The RSA system

It is one of the most popular public key systems, named after the surnames of its designers **R**. L Rivest, A. **S**hamir and L. **A**dleman of *Massachusetts Institute of Technology***.** The RSA system is based on the fact that it is relatively easy to calculate the product of two prime numbers, but that determining the original prime numbers, given the product is far more complicated. The RSA cipher is a fascinating example of how some of the most abstract mathematical subjects find applications in the real world. Let us move on to the RSA algorithm, which can be discussed under the following headings.

Key generation

Suppose a user wishes to allow others to send him a private message over an insecure transmission medium. He takes the following steps to generate a *public key* and a *private key*.

a) Choose two large prime numbers 'p' and 'q', randomly and independently of each other, such that $p \neq q$
b) Compute $n=p*q$
c) Calculate $\phi(n)=(p-1)(q-1)$
d) Choose an integer $3<e<n$ which is co prime to $\phi(n)$ (*extended Euclidean algorithm*)
e) Compute d such that $ed \bmod \phi(n) = 1$ (*extended Euclidean algorithm)*

Let 'M' be the plain text and 'C' represent the cipher text.

Encryption	Decryption
Choose $M<n$.	*$M= C^d \pmod n$*
$C= M^e \pmod n$	

Lets illustrate this with an example.

Key generation

a) Let $p=3$ and $q=17$
b) $n=p*q=51$
c) $\phi(n)=(p-1)(q-1)=32$
d) $e=7$ (say)
e) $d=23$ ($ed \bmod 32=1$)

Now, we have two set of keys (n, e) and (n, d). One of them is distributed and is called *public key*. While the other one is kept secret and is called *private key*. There is no rule that only particular one should be made public. However we follow the convention and choose (n, e) i.e. (51, 7) as public.

For encipherment

- Let $M=2$
- $C=M^e \pmod n = 2^7 \pmod{51} = 26$

For decipherment,

- $M=C^d \pmod n = 26^{23} \pmod{51} = 2$: which is the initial value of M.

This is rather a simple example. Actually, the value of n should be represented by some 200 bits, i.e. N should be of the order of 2^{100}. P and q are very sensitive since, they are the factors of n and allow computation of d and e. So, they are often deleted or sometimes securely kept along with d in order to speed up the decryption. It is relatively easy to calculate the private and the public key but revealing one does not reveal any easy way to compute other: this property is known

as the *Trap Door Function* because it is easy to calculate the function in one direction but not in the other direction.

The security of the system relies on the fact that it is almost impossible to calculate the value of d if only the public key (n, e) is known. Since only n is publicly available, a cryptanalyst must determine the value of p and q. If n is of the order of some 200 decimal digits, this would take about *30 million years*, with current technology. Thus, the person who issues the private key is the only person who knows the precise value of d and therefore only person able to decipher the encrypted text.

<The security of any encryption scheme depends on the length of the key and the computational work involved in breaking the code. There is nothing in principle about either conventional or public-key encryption that makes one superior to another from the point of view of resisting cryptanalysis.>

Final Words

The good news about cryptography is that we already have the algorithms and protocols we need to secure our systems. The bad news is that implementing the protocols successfully requires considerable expertise. Most systems are not designed and implemented in concert with cryptographers, but by engineers who thought of cryptography as just another component. They don't realize that the systems cannot be made secure by tacking on cryptography as an afterthought. You have to know what you are doing every step of the way, from conception through installation.

Companies often get the easy part wrong, and implement insecure algorithms and protocols. But even so, practical cryptography is rarely broken through the mathematics; other parts of systems are much easier to break. Flaws can be anywhere: the threat model, the system design, the software or hardware implementation, the system management. Security is a chain, and a single weak link can break the entire system.

<Netscape's security fell to a bug in the random-number generator.>

Strong cryptography can withstand targeted attacks up to a point—the point at which it becomes easier to get the information some other way. Generally, the firmware upgrades are encrypted while transmission. Once they are fed into hardware, they are decrypted internally and the upgrade does its job. This sounds well because the encrypted firmware upgrade cannot be decrypted without knowing the proper key. However, a truly determined attacker with hardware skills is not bothered of whether the firmware upgrade is encrypted or not, simply because he can snatch the firmware upgrade from flash memory once it's decrypted.

Attackers don't follow rules; they cheat. They can attack a system using techniques the designers never thought of. Art thieves have burgled homes by cutting through the walls with a chain saw. Home security systems, no matter how expensive and sophisticated, will not stand a chance against this attack. Computer thieves come through the walls too. They steal technical data, bribe insiders, modify software, and collude. They take advantage of technologies newer than the system, and even invent new mathematics to attack the system with.

Cryptography, thus, cannot guarantee 100% security. But we can work toward 100% risk acceptance. Fraud exists in current commerce systems: cash can be counterfeited, checks altered, credit card numbers stolen. Yet these systems are still successful because the benefits and conveniences outweigh the losses. Privacy systems—wall safes, door locks, curtains—are not perfect, but they're often good enough. *A good cryptographic system strikes a balance between what is possible and what is acceptable.*

Security is different from any other design requirement, because FUNCTIONALITY does not equal QUALITY. If a word processor prints successfully, you know that the print function works. Security is different; just because a safe recognizes the correct combination does not mean that its contents are secure from a safecracker. No amount of general BETA testing will reveal a security flaw, and there is no such test possible that can prove the absence of flaws.

History has taught us: never underestimate the amount of money, time, and effort someone will expend to thwart a security system. *It is always better to assume the worst.* Assume your adversaries are better than they are. Assume science and technology will soon be able to do things they cannot yet. Give yourself a margin for error. Give yourself more security than you need today. When the unexpected happens, you will be glad you did. ■

Encryption through Cascaded Recursive Arithmetic Operation and Key Rotation of a session key

CRAOKR

Pawan Kumar Jha[1]
Dr. Subarna Shakya[2]

abstract

The technique considers a message as binary string on which a Cascaded Recursive Arithmetic Operation and Key Rotation (CRAOKR) of a session key is applied. A block of n bits is taken as an input stream, where n varies from 4 to 256, from a continuous stream of bits and the technique operates on it in two phases, in first phase plain text is encrypted by using recursive addition and then encrypts the output in the second phase to generate the intermediate encrypted streams. The same operation is performed repeatedly for different block sizes as per the specification of a session key of a session to generate the final encrypted stream.

It is a kind of block cipher and symmetric in nature hence, decoding is done following the same procedure. A comparison of the proposed technique with existing and industrially accepted RSA and TDES has also been done in terms of frequency distribution and homogeneity of source and encrypted files.

Key words: *Cascaded Recursive Arithmetic Operation and Key Rotation (CRAOKR), Cipher text, Block cipher, Session Key.*

1. Introduction

Cryptography is an essential tool in communication. To protect the data, information, messages from eavesdroppers each and every application use this tool and technique. Many algorithms are available each, of which has merits and demerits [2, 3, 4, 5, 7]. No single algorithm is sufficient for this application. As a result researchers are working in the field of cryptography to enhance the security further. The encryption becomes very useful at this point of time, when the people all over the world are engaged in communication through internet almost everyday. The encryption process converts the document into cipher text, which will not be legible to the intruder [6, 8, 9, 10, 11].

In this paper a new technique is proposed where the source message is considered as a stream of binary bits and can be conceived as block of 4/8/16/32/64/128/256 bits concatenated to each other. The technique transforms the document into unintelligible form from where the original message can also be recovered into the original form using the same technique. The technique has been implemented using C language.

Section 2 of the paper deals with the principle of Cascaded Recursive Arithmetic Operation and Key Rotation (CRAOKR). A proposal for key generation is described in section 3. Vulnerability and results are given in section 4 and 5. Analysis about the technique is made in section 6. Conclusions are drawn in section 7 and references are drawn in section 8.

[1] *School of Engineering and Technology Purbanchal University, Biratnagar, Nepal, e-mail: amp_jha@yahoo.com*

[2] *Department of Electronics and Computer Engineering, Institute of Engineering, T.U., e-mail: drss@ioe.edu.np*

2. Principle of Cascaded Recursive Arithmetic Operation and Key Rotation (CRAOKR)

This technique operates in two phases:

a. First phase encrypt the plaintext using Recursive Arithmetic Operation

The technique, considers the plaintext as a stream of finite number of bits N, and is divided into a finite number of blocks, each also containing a finite number of bits n, where, $1 <= n <= N$.

Let $P = s^0_0\ s^0_1\ s^0_2\ s^0_3\ s^0_4 \ldots s^0_{n-1}$ is a block of size n in the plaintext. Then the first intermediate block $I_1 = s^1_0\ s^1_1\ s^1_2\ s^1_3\ s^1_4 \ldots s^1_{n-1}$ can be generated from P in the following way:

$$s^1_0 = s^0_0$$

$$s^1_{n-1} = s^0_{n-1}$$

$$s^1_i = s^0_i \oplus s^0_{i+1},\ 1 < i < (n-1);$$

$\oplus$ stands for the exclusive-OR operation.

In the same way, the second intermediate block $I_2 = s^2_0\ s^2_1\ s^2_2\ s^2_3\ s^2_4 \ldots s^2_{n-1}$ of the same size (n) can be generated by:

$$s^2_0 = s^1_0$$

$$s^2_{n-1} = s^1_{n-1}$$

$$s^2_i = s^1_i \oplus s^1_{i+1},\ 1 < i < (n-1).$$

Any intermediate block in the recursive process may term as intermediate encrypted block for that source block and any block can be taken as the input for the second phase.

b. Second phase encrypt the output of the first phase by Recursive Key Rotation

The technique considers the encrypted message from the first phase in the form of blocks of bits with different size like 8/16/32/64/128/256. The rules to be followed for generating a cycle are as follows:

1. Consider any source stream of a finite number (where $N=2^n$, n =3 to 8) and divide it into two equal parts.
2. Consider any key value (key= 2^n, where n=1 to 7) depends upon the source stream that is, key value is the half of the source stream).
3. Make the modulo-2 addition (X-OR) with the key value to the first half of the source stream, to get the first intermediate block.
4. Make the modulo-2 addition with the key value (but now the key value is reversed) to the last half of the source stream to get the second intermediate block.

The same operation is performed for whole stream number of time with a varying block sizes. K such iteration is done and the any intermediate stream after k iterations generates the cascaded form of the encrypted stream. All of the different block sizes and k constitute the key for the session. This key may be considered as session key for that particular session. This process is repeated until the source stream is generated. Figure 2.1 and Figure 2.2 represents the first and second phase of the technique pictorially (single step of the iteration process)

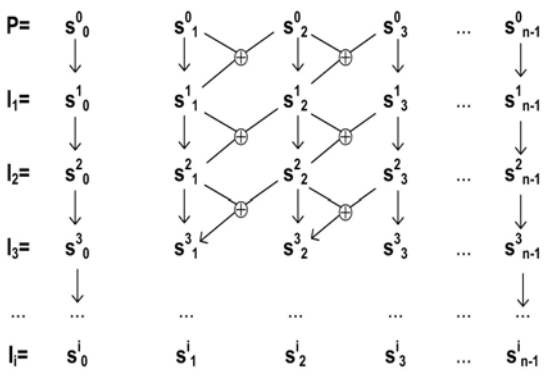


Figure 2.1 Pictorial representation of the first phase of the technique

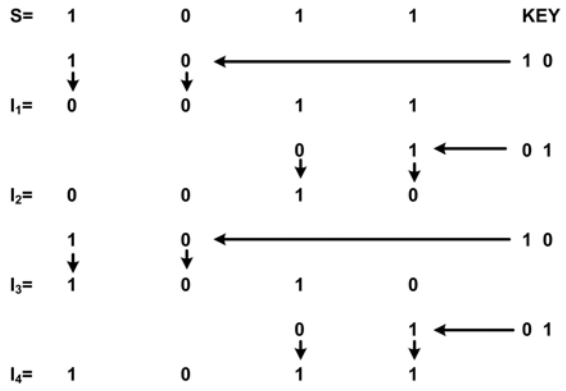


Figure 2.2 Pictorial representation of the second phase of the technique

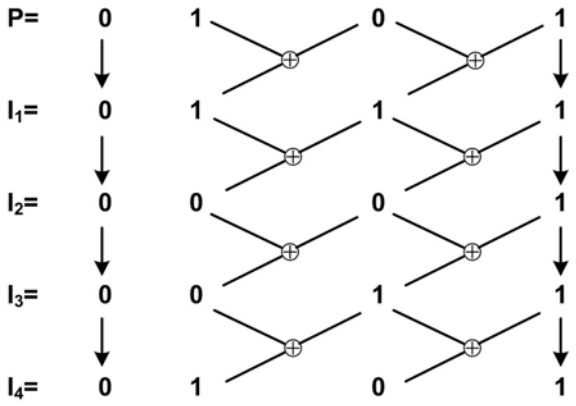


Figure 2.3 Pictorial Representation of the flow of the first phase of the technique for source block P=0101

Source Block			
0	0	1	1
1	**2**	**3**	**4**
Block After 1st Iteration			
1	0	1	1
1	**2**	**3**	**4**
Block After 2nd Iteration			
1	0	1	0
1	**2**	**3**	**4**
Block After 3rd Iteration			
0	0	1	0
1	**2**	**3**	**4**
Block After 4th Iteration			
0	0	1	1
1	**2**	**3**	**4**

Figure 2.4 Pictorial Representation of the flow of the second phase of the technique for source block P=0011

2.1 Example

To illustrate the technique (CRAOKR), let P=0101 be a 4-bit source block. Figure 2.3 shows the generation of the cycle for this sample block. Here, it requires 4 iterations to regenerate the source block. Here any intermediate block can be taken as the input stream for the second phase of the technique. For the second phase of encryption third intermediate block (0 0 1 1) is taken as the input stream.

In this way, for different blocks sizes in the plaintext corresponding cycles are formed for different number of iterations. If the blocks are taken of the same size, the number of iterations required in forming the cycles will be equal and hence that number of iterations will be required to complete the cycle for the entire stream of bits.

With respect to one single block of bits, any intermediate block during the process of forming the cycle can be considered as the encrypted block. If the total number of iterations required to complete the cycle is P and the ith block is considered to be the encrypted block, then a number of (P – i) more iteration will be required to decrypt the encrypted block, i.e., to regenerate the source block. If the process of encryption is considered for the entire stream of bits, then it depends on how the blocks have been formed. Out of the entire stream of bits, different blocks can be formed in two ways:

i. Blocks with equal size
ii. Blocks with different sizes.

In case of blocks with equal length, if for all blocks, intermediate blocks after a fixed number of iterations are considered as the corresponding encrypted blocks, then that very number of iterations will be required for encrypting the entire stream of bits. The key of the scheme will be quite simple, consisting of only two information, one being the fixed block size and the other being the fixed number of iterations for all the blocks used during the encryption. On the other hand, for different source blocks different intermediate blocks may be considered as the corresponding encrypted blocks. For example, the policy may be something like that out of three source blocks B1, B2, B3 in a source block of bits, the 4th, the 7th and the 5th intermediate blocks respectively are being considered as the encrypted blocks. In such a case, the key of the scheme will become much more complex, which in turn will ensure better security.

In the case of blocks with varying lengths, different blocks will require different numbers of iteration to form the corresponding cycle. So, the LCM value, say, P, of all these numbers will give the actual number of iterations required to form the cycle for the entire stream. If i number of iterations are performed to encrypt the entire stream, then a number of (P – i) more iterations will be required to decrypt the encrypted stream.

3. Key Generation

To ensure the successful encryption of the proposed technique with varying size of blocks, a

133 bit key format consisting of 14 different segment has been proposed here [1,11,6,8,9].

For the segment of rank the R, there can exist a maximum of N=216-R blocks, each of unique size of S=216-R, R starting from 0 and moving till 13.

- Segment with R=0 formed with the first maximum 65536 blocks, each of size 65536 bits
- Segment with R=1 formed with the next maximum 32768 blocks, each of size 32768 bits
- Segment with R=2 formed with the next maximum 16384 blocks, each of size 16384 bits
- Segment with R=3 formed with the next maximum 8192 blocks, each of size 8192 bits
- Segment with R=4 formed with the next maximum 4096 blocks, each of size 4096 bits
- Segment with R=5 formed with the next maximum 2048 blocks, each of size 2048 bits
- Segment with R=6 formed with the next maximum 1024 blocks, each of size 1024 bits
- Segment with R=7 formed with the next maximum 512 blocks, each of size 512 bits
- Segment with R=8 formed with the next maximum 256 blocks, each of size 256 bits
- Segment with R=9 formed with the next maximum 128 blocks, each of size 128 bits
- Segment with R=10 formed with the next maximum 64 blocks, each of size 64 bits
- Segment with R=11 formed with the next maximum 32 blocks, each of size 32 bits
- Segment with R=12 formed with the next maximum 16 blocks, each of size 16 bits
- Segment with R=13 formed with the next maximum 8 blocks, each of size 8 bits.

With such a structure, the key space becomes of 133 bits long and a file of the maximum size of around 699.05 MB can be encrypted using the proposed technique. The key structure is represented in figure 3.5

4. Vulnerability

We can consider the time required to use a brute force approach, which simply involves trying every possible key until an intelligent translation of the cipher text into plain text is obtained. On average, half of all possible key must be tried to achieve success. Table 3.1 shows how much time is involved for various key spaces. Results are shown for four binary sizes. The 56-bit key sizes used with the DES (Data Encryption Standard) algorithm, and the 168-bit key size is used for triple DES. The minimum key size specified for AES (Advanced Encryption Standard) is 128-bits. Results are also shown for what are called substitution codes that use a 26 characters key, in which all possible permutations of the 26 characters serves as keys. For each key size, the results are shown assuming that it takes 1 μs perform a single decryption, which is a reasonable order of magnitude for today's machine. Within the use of massively parallel organizations of microproces-

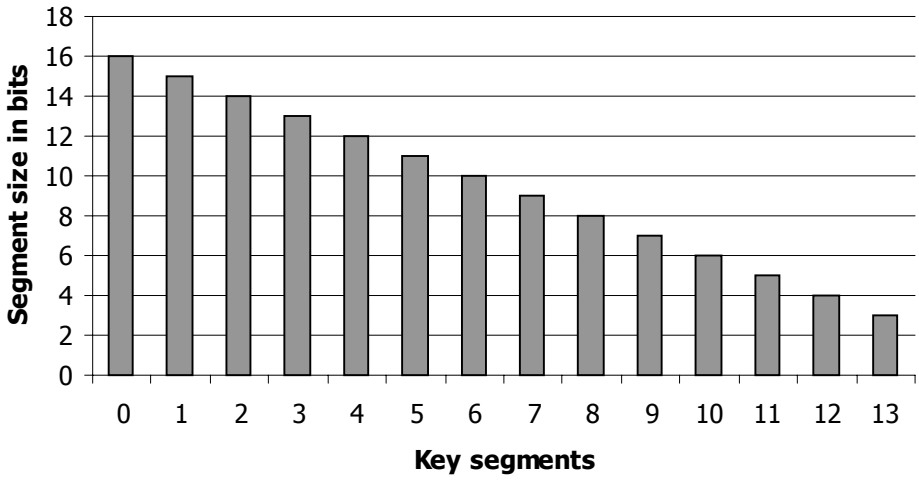


Figure 3.5 133-bit key format with 14 segments for CRAOKR Technique

sors, it may be possible to achieve processing rates may orders of magnitude greater. The final column of the table 4.1 considers the result for a system that can process a 1 million keys per microsecond. As one can see at this performance level, DES can no longer be considered computationally secure. In the proposed technique key size is 133 bits though this key may be changed to other size as per the requirements.[3, 14, 15, 16, 17, 18, 19]

Key Size(bits)	No. of Alternate Keys	Time required at 1 encryption / μs	Time required at 10^6 encryption /μs
32	$2^{32}=4.3*10^9$	2^{31}= 35.8 minutes	2.15 milliseconds
56	$2^{56}=7.2*10^{16}$	2^{55} μs=1142 years	10.01 hours
128	$2^{128}=3.4*10^{38}$	2^{127} μs=$5.4*10^{24}$ years	$5.4*10^{18}$ years
168	$2^{168}=3.7*10^{50}$	2^{167} μs=$5.9*10^{36}$ years	$5.1*10^{30}$ years
26 Characters (Permutations)	26! $4*10^{26}$ μs	$2*10^{26}$ μs=$6.4*10^{12}$ years	$6.4*10^6$ years
133 (Proposed CRAOKR technique)	$2^{133}=1.08*10^{40}$	2^{132}μs =$1.7*10^{26}$ years	$1.7*10^{20}$ years

Table 4.1 Average time required for exhaustive key search

5. Results

In this section the results of implementations are presented. The implementation is made through high-level language. Table 5.1 represents the encryption time, decryption time, and size before and after encoding and decoding. The encryption time varies from 0.054945 to 0.329070. The decryption time varies from 0.054945 to 0.274725 for the present implementation. These are shown in the figures 5.1 and 5.2 respectively. Figure 5.3 gives a relationship between encryption times against decryption time. The frequency distribution graphs for source and encrypted files for CRAOKR and existing RSA and TDES techniques are given in figure 5.4.

Chi-square test has also been done and presented in table 5.2 for source files and encrypted files. Results of the Chi square tests are compared with RSA technique for source files and encrypted files.

Source File	Source size (bytes)	Encryption time (sec.)	Output file name	Output file size (bytes)	Decryption time (sec.)
viewprev.cpp	30848	0.054945	deo1.cpp	30848	0.054945
olecli2.cpp	41023	0.054945	deo2.cpp	41023	0.054945
olecli1.cpp	61600	0.054945	deo3.cpp	61600	0.054945
inet.cpp	72980	0.10989	deo4.cpp	72980	0.054945
occsite.cpp	89786	0.10989	deo5.cpp	89786	0.054945
dbrfx.cpp	91269	0.10989	deo6.cpp	91269	0.10989
wincore.cpp	109141	0.16989	deo7.cpp	109141	0.10989
dbcore.cpp	115208	0.16989	deo8.cpp	115208	0.16989
daocore.cpp	135431	0.21978	deo9.cpp	135431	0.16989
book.cpp	143336	0.32907	deo10.cpp	147533	0.274725

Table 5.1 File size vs encryption times, decryption times for .CPP files

Graph in figure 5.4 shows the frequency of characters in a message and that frequency of characters in the encrypted message for CRAOKR technique and existing RSA and TDES techniques. The close observation reveals that the character frequencies are more evenly distributed for the CRAOKR technique. In connection with the Brute-force attack of decrypting the message by the hackers, it may be difficult to decrypt the message if the message is encrypted using the proposed key system or like manner.

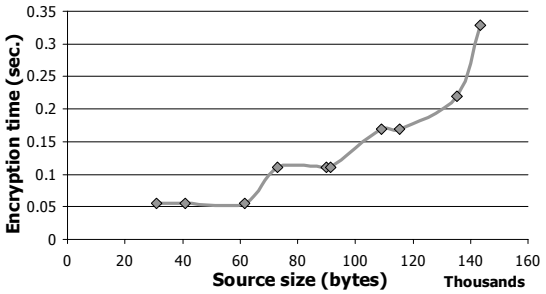


Figure 5.1 Encryption time against file sizes for CRAOKR technique

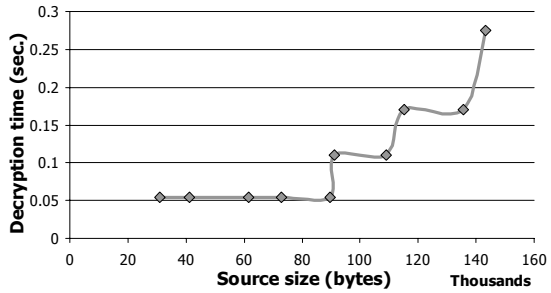


Figure 5.2 Decryption time against file sizes for CRAOKR technique

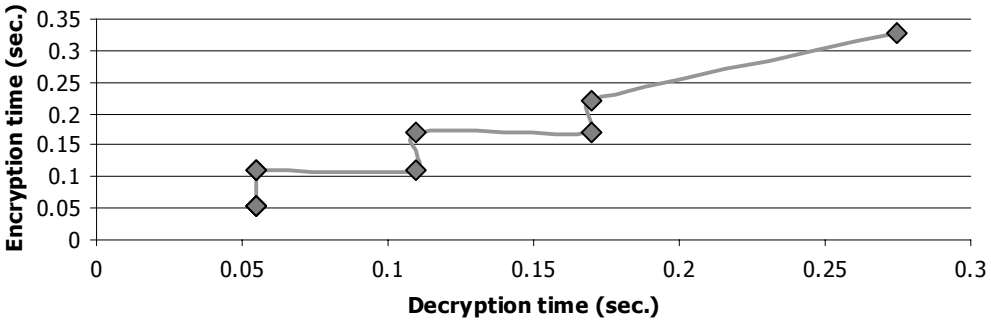


Figure 5.3 Encryption time against decryption time for CRAOKR technique

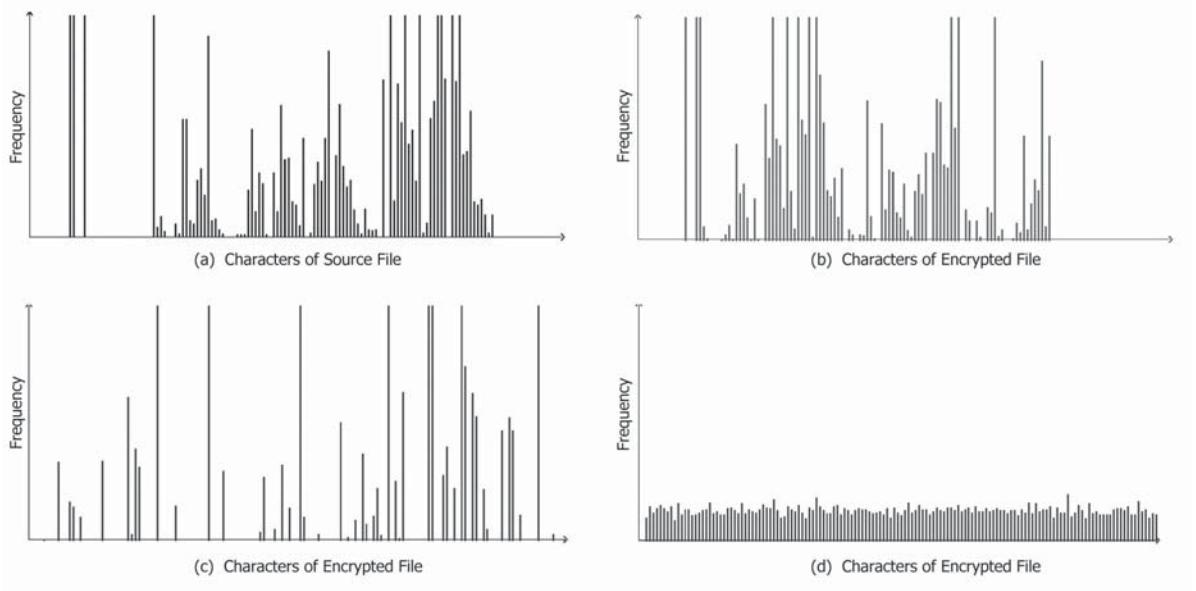


Figure 5.4 Frequency distribution for VIEWPREV.CPP: Source file (a), encrypted using CRAOKR technique (b), using RSA technique (c) and using TDES technique (d).

5.1 Tests for Homogeneity

The Chi-square test has also been performed using source file and encrypted files for CRAOKR and existing RSA and TDES Technique.

	Source file	File size (bytes)	Chi square CRAOKE	RSA	TDES
1.	viewprev.cpp	30,848	404,054	1,015,121	54,694
2.	olecli2.cpp	41,023	1,524,348	750,711	73,369
3.	olecli1.cpp	61,600	399,561	215,853	109,858
4.	inet.cpp	72,980	268,300	223,480	199,820
5.	occsite.cpp	89,786	437,527	302,856	159,044
6.	dbrfx.cpp	91,269	287,455	618,369	35,672
7.	wincore.cpp	109,141	807,080	411,302	194,013
8.	dbcore.cpp	115,208	1,216,993	401,363	204,655
9.	daocore.cpp	135,431	4,261,176	380,307	201,857
10.	book.cpp	143,336	13,680,462	19,977,116	269,585

Table 5.2 Value of Chi-square for different File sizes in CRAOKR Technique

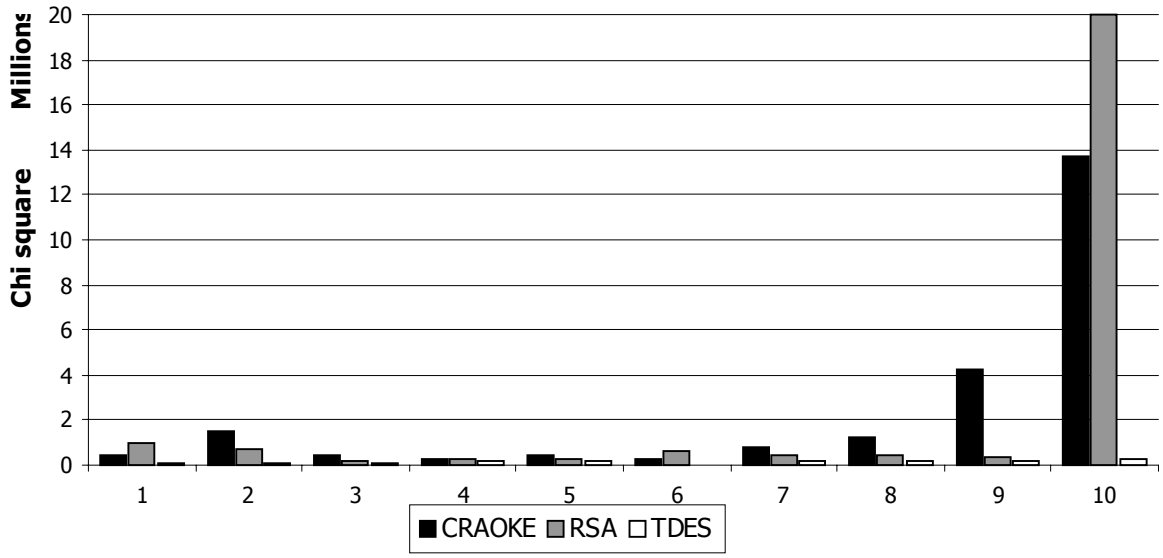


Figure 5.5 Comparison of Chi square values between CRAOKR, RSA, and TDES techniques

Table 5.2 shows the values of Chi-square for different file size, which shows that the value of Chi-square is increasing as file size is increasing. In case of CRAOKR the average value obtained for Chi-square test is 2328696. In case of existing RSA and TDES technique the average value of Chi-square test for present implementation is 2429648 and 150257 for the same source files. In all the cases the Chi-square is highly significant at 1 % level of significance. So we may conclude the source files and encrypted files are non-homogeneous in both CRAOKR and existing RSA and TDES techniques. The Chi-square values are nearer to each other for both of CRAOKR and existing RSA and TDES technique in present implementation. Hence, it may be inferred that the techniques are comparable. Figure 5.5 shows the results of comparison of Chi square values between proposed CRAOKR technique and existing RSA and TDES technique. The bars of gray shade represent the Chi square for RSA technique, black shade represent the proposed CRAOKR technique and that of white shed represents the TDES technique. It is clear from the graph and the table that Chi square values for CRAOKR shows either better or comparable results.

6. Analysis

Analyzing all the results presented in section 4, following are the points obtained on the proposed technique:

- The encryption time and the decryption time vary linearly with the size of the source file.
- There exist not much difference between the encryption time and the decryption time for a file, establishing the fact that the computation complexity of each of the two processes is of not much difference.
- For non-text files, such as .exe, .com, .dll, and .sys files there is no relationship between the source file size and the Chi square value.
- Chi Square values for text files, such as .cpp files are very high and vary linearly with the source file size.
- Out of the different categories of files considered here, Chi Square values for .cpp files are the highest.
- The frequency distribution test applied on the source file and the encrypted file shows that the characters are all well distributed. Chi square values for this proposed technique and those for the RSA and TDES system highly compatible. Except some cases the Chi square values in proposed technique either better or almost equal in most of the files. For low file sizes, proposed CRAOKR technique gives high Chi Square values.

7. Conclusion

The technique presented here is a simple, takes little time to encode and decode, though the block length is quite high. The encoded string will not generate any overhead bits. It can be easily implemented in any high level language for practical application purpose to provide security in message transmission.

Acknowledgement: The authors express a deep sense of gratitude to the Department of Computer Science and Application, University of North Bengal, Dist. Darjeeling, India for providing necessary infrastructure support for the work.. The scholarship to one of the author is provided by the UGC, Nepal. The authors also express their gratitude to the Dept. of Computer Application Kalyani Govt. Engineering College Kalyani, Nadia, India where the current processing is being done. ■

8. References

1. Jha P. K. Mandal, J. K., *A Bit Level Symmetric Encryption Technique Through Recursive Arithmetic Operation (RAO) to Enhance the Security of Transmission*, Proceedings of ICT Conference 2005, An International Conference of Computer Association of Nepal, held during 26-27th January, 2005 at BICC, Kathmandu, Nepal.
2. Jha, P. K., Mandal, J. K., *A Symmetric encryption technique through recursive modulo-2 operation of paired bits of streams (RMOPB)*. Proceedings of third international conference on innovative applications of information technology for developing world (AACC-2005), Imperial College Press, Covent Garden, London, WC2H 9HE, held during 10-12th December 2005 at Kathmandu.
3. Jha, P. K., Mandal, J. K., *A bit Level symmetric encryption technique through recursive key rotation (RKR) of a session key,* communicated and accepted to the Proceedings of ICT conference 2006, an international conference of computer association of Nepal held during 26-28th January, 2006 at BICC, Kathmandu, Nepal.
4. Jha, P. K., Mandal, J. K., *A bit Level symmetric encryption technique through recursive bitwise arithmetic manipulation (RBAM) to enhance the security of transmission*, communicated and accepted to the Proceedings of ICT conference 2006, an international conference of computer association of Nepal held during 26-28th January, 2006 at BICC, Kathmandu, Nepal.
5. Jha, P.K., Mandal, J.K, Shakya. S., *A Bit level symmetric Encryption Technique Through Recursive Transposition Operation (RTO) to Enhance the Security of Transmission*, Journal of Insitute of Engineering, Vol. 5, No. 1, PP-106-116, December 2005, Pulchowk, Nepal.
6. Dutta S et al., *Ensuring e-Security using a Private-key Cryptographic System Following Recursive Positional Modulo-2 Substitutions*, AACC 2004, LNCS 3285, Springer-Verlag Berlin Heidelberg, pp. 324-332, 2004.

7. Mandal J. K. and Dutta S., *A Space-Efficient Universal Encoder for Secured Transmission*, International Conference on Modeling and Simulation (MS' 2000-Egypt), Cairo, April 11-14,, pp-193-201, 2000.
8. Mandal J.K. and Dutta S., *A Universal Bit-Level Encryption Technique*, Proceedings of the 7th State Science and Technology Congress, Jadavpur University, West Bengal, India, February 28 - March 1, pp-INFO2, 2000.
9. D. Welsh, *Codes and Cryptography*, Oxford: Claredon Press, 1988
10. J. Seberry and J. Pieprzyk, *An Introduction to Computer Security*, Prentice Hall of Australia, 1989.
11. D. Boneh, *Twenty Years of Attacks on RSA Cryptosystem*, in notices the American Mathematical Society (AMS), vol 46,no 2, pp 203-213,1998.
12. B. Schneier, *Applied Cryptography*, Second Edition, John Wiley & Sons Inc, 1996.
13. C. Coupe, P. Nguyen and J. Stern, *The Effectiveness of Lattice Attacks against Low-Exponent RSA*, Proceedings of Second International Workshop on Practice and Theory in Public Key Cryptography, PKC'99, Vol. 1560, of lecture notes in Computer Science, Springer-Verlag, pp 204-218, 1999.
14. *RSA Vulnerabilities*, Journal of Cryptology, Vol. 10, pp 233-260,1997.
15. M. Wiener, *Cryptanalysis of Short RSA Secret Exponents*, IEEE Transactions on Information Theory, Vol. 36, pp 553-558, 1990.
16. V. P. Gulati, A Saxena and D Nalla, *On Determination of Efficient Key Pair*, Journal of the Institution of Engineers (India), Vol. 84, pp 1-3, May 2003.
17. William Stallings, *Cryptography and Network security*, Pearson Education (Singapore) Pte. Ltd., 2004.
18. *Recent Advances in Computing and Communication.* proceedings of the 12th international Conference on Advanced Computing and Communications, Dec 15-18, 2004, Ahmedabad, India.

Congratulations

The **Zerone** Team heartily congratulates

Dr. Subarna Shakya

on being awarded the prestigous

"श्री ५ युवराजाधिराज युवा विज्ञान तथा प्रविधि पुरस्कार"

in a program organised by the **RONAST**

LZW Coding

Lal Babu Sah
2058 Computer

LZW compression has been named after its developers, *Abraham **L**empel* and *Jakob **Z**iv (1977)*, with later modifications by *Terry A. **W**elch (1984)*. It is the foremost technique for general purpose data compression due to its simplicity and versatility. It is a proprietary lossless data-compression algorithm. A lossless data-compression algorithm is one in which the original message can be decoded exactly. A lossless data-compression can be called *data redundancy reduction*. Typically, we can expect LZW to compress text, executable code, and similar data files to about one-half their original size. LZW also performs well when presented with extremely redundant data files, such as tabulated numbers, computer source code, and acquired signals. Compression ratios of 5:1 are common for these cases.

LZW is a general compression algorithm capable of working on almost any type of data. It is generally fast in both compressing and decompressing data and does not require the use of floating-point operations. In addition, because LZW writes compressed data as bytes and not as words, LZW-encoded output can be identical on both *Big Endian* and *Little Endian* systems although we may still encounter bit order and fill order problems.

LZW is the basis of several personal computer utilities that claim to "double the capacity of our hard drive". Despite the fact that it must be licensed under United States Patent No. 4,558,302, granted December 10, 1985 to Sperry Corporation (now the Unisys Corporation), LZW compression has been integrated into a variety of mainstream imaging file formats, including the *graphic interchange format* (GIF), *tagged image file format* (TIFF), and the *portable document format* (PDF).

Introduction

LZW is a way of compressing data that takes advantage of repetition of strings in the data. Since raster data usually contains a lot of this repetition, LZW is a good way of compressing and decompressing it. LZW is referred to as a *substitution* or *dictionary-based encoding algorithm*. The algorithm builds a *data dictionary* (also called a *translation table* or *string table*) of data occurring in an uncompressed data stream. Patterns of data (*substrings*) are identified in the data stream and are matched to entries in the dictionary. If the substring is not present in the dictionary, a code phrase is created based on the data content of the substring, and it is stored in the dictionary. The phrase is then written to the compressed output stream. When a recurrence of a substring is identified in the data, the phrase of the substring already stored in the dictionary is written to the output. Because the phrase value has a physical size that is smaller than the substring it represents, data compression is achieved.

Decoding LZW data is the reverse of encoding. The decompress program reads a code from the encoded data stream and adds the code to the data dictionary if it is not already there. The code is then translated into the string it represents and is written to the uncompressed output stream.

we can expect LZW to compress text, executable code, and similar data files to about one-half their original size

LZW goes beyond most dictionary-based compressors in that it is not necessary to preserve the dictionary to decode the LZW data stream. This can save quite a bit of space when storing the LZW-encoded data. When compressing text files, LZW initializes the first 256 entries of the dictionary with the 8-bit ASCII character set (values 00h through FFh) as phrases. These phrases represent all possible single-byte values that may occur in the data stream, and all substrings are in turn built from these phrases. Because both LZW encoders and decoders begin with dictionaries initialized to these values, a decoder need not have the original dictionary and instead will build a duplicate dictionary as it decodes.

Thus, LZW compression starts with a *data dictionary* containing only the first 256 entries, with the remainder of the table being blank. As the encoding continues, the LZW algorithm identifies repeated sequences in the data, and adds them to the code table. Compression starts the second time a sequence is encountered. The key point is that a sequence from the input file is not added to the code table until it

has already been placed in the compressed file as individual characters (codes 0 to 255). This is important because it allows the decompression program to reconstruct the code table directly from the compressed data, without having to transmit the code table separately.

LZW compression works best for files containing lots of repetitive data. This is often the case with text and monochrome images. Files that are compressed but that do not contain any repetitive information at all can even grow bigger! LZW compression is fast. ■

LZW Compression: Pseudo code

```
Initialize String Table
Add next character from character_stream to String_Buffer
START LOOP here
     get Next_Character from character_stream
     if String_Buffer + Next_Character is in string table
          add Next_Character to String_Buffer
     else
          output code for String_Buffer
          add String_Buffer + Next_Character code to table
          clear String_Buffer and equal it to Next_Character
END LOOP when at end of character_stream
```

LZW Decompression: Pseudo Code

```
Initialize String Table
get First_Code from character_stream
output First_Code
START LOOP here
     get Next_Code from character_stream
     if Next_Code is NOT in the string table
          String_Buffer = translated First_code + first byte of First_Code
     else
          String_Buffer = Translation of Next_Code
          add translated First_code+first byte of First_Code to the table
          First_Code = Next_Code
          output String_Buffer
END LOOP when at end of charstream
```

String _Buffer	Next _Character	Code	Code value	Output
	a			
a	a	256	aa	a
a	a			
aa	b	257	aab	256
b	b	258	bb	b
b	b			
bb	c	259	bbc	258
c	c	260	cc	c
c	c			
cc				260

compresssion of the string "aaabbbccc"

String _Buffer	First _code	Next _code	Code	Code value	Output
a	a				
aa	256	256	256	aa	aa
b	b	b	257	aab	b
bb	258	258	258	bb	bb
c	c	c	259	bbc	c
cc	260	260	260	cc	cc

decode the previous output string "a 256 b 258 c 260"

References

1. Steve Blackstock, *LZW and GIF explained*
2. www.dspguide.com/datacomp.htm

Negative Frequency (-ω) Demystified

Saurav R. Tuladhar
2059 Electronics

A course in Signal Analysis or Communication Systems will have introduced to you the term *negative frequency*. A frequency domain representation of a signal (periodic/aperiodic) is shown with its positive and negative frequency components, although the negative side is symmetrical to its positive counterpart. The concept of negative frequency at times seem trivial, against the notion that frequency is a purely positive quantity, but at other times the concept can be quite troublesome in grasping the critical concepts. Here I present a discussion on the negative aspect of frequency.

First of all, *frequency* refers to the rate at which a phenomenon is repeated i.e repetitions per second. Its unit is Hertz. Simply from the definition '*frequency*' seems to be a purely positive quantity. How can a repetition be a negative quantity? However the frequency domain representation of any band limited signal (such as audio signal), will have symmetrical representation in positive and negative frequency. A typical frequency domain plot of a 50 Hz sinusoid is shown alongside.

The best way to understand the existence of negative frequency would be through mathematics behind it. The mathematical approach towards the existence of positive and negative frequency can be seen using *Euler's Identity* according to which:

$$\cos\theta = \frac{e^{j\theta} + e^{-j\theta}}{2}$$

$$\sin\theta = \frac{e^{j\theta} - e^{-j\theta}}{2}$$

Setting $\theta = \omega t + \varphi$, we see that both sine and cosine and hence all real sinusoids, consist of a sum of equal and opposite circular motion ($e^{j\omega}$ represents a unit circular motion). Mathematically, the complex sinusoid $Ae^{j(\omega t + \varphi)}$ is infact *simpler* than the real sinusoid $A\sin(\omega t + \varphi)$ because $e^{j\omega t}$ consists of

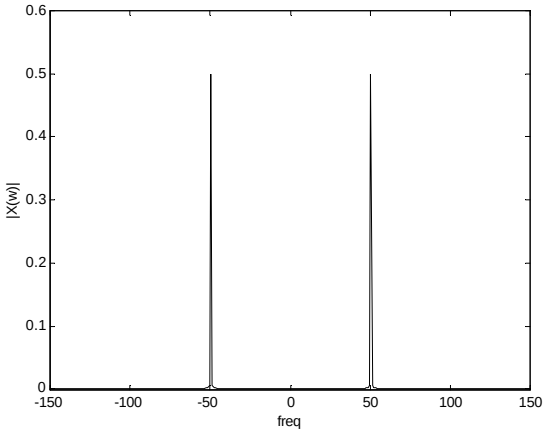


Figure 1. Magnitude spectrum of a 50 Hz sinusoid.

a single frequency ω (Fourier Transform of $e^{j\omega t}$ is a shifted delta function) while $\sin(\omega t)$ consists of two frequencies ω and -ω. This is a direct consequence of the definition from Euler's Identity. We may think of a real sinusoid as being the sum of a positive-frequency and a negative-frequency complex sinusoid. In other words, every real sinusoid consists of an equal contribution of positive and negative frequency components. And from Fourier theory, this is true of all real signals. When we perform spectrum analysis of such signals, we will find that every real signal contains equal amounts of positive and negative frequencies, *i.e.*, if X(ω) denotes the spectrum of the real signal x(t), we will always have $|X(\omega)| = |X(-\omega)|$

Further, since frequency is defined as the time derivative of phase, negative frequency may be seen as the rate of clockwise rotation in phase, and for negative frequency the phase φ decreases with time.

As stated earlier, a complex sinusoid has only one frequency (either positive or negative). Hence $x(t) = e^{-j\omega t} = \cos(\omega t) - \mathbf{j}\sin(\omega t)$ is a signal with only negative frequency and the signal is shown alongside. The real part (shown by a solid line) is a

cosine wave, and the imaginary part (shown by a dashed line) is a sine wave.

A parametric plot of real versus imaginary part of the signal, traces a circle, as the parameter (time)

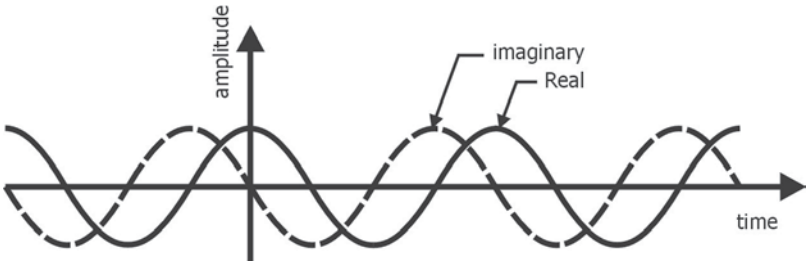


Figure 2. $x(t) = e^{j\omega t} = cos(\omega t) - jsin(\omega t)$

increases. The direction of rotation around the circle will be clockwise as time increases. Such a plot is called Corkscrew plot which is show alongside. Since positive angle is defined as counter-clockwise rotation, we have that the signal is moving in a direction of negative angle and hence the signal has a negative frequency.

So from above discussion the origin and existence of negative frequency has been established. Now coming back to spectrum analysis of signals, based upon the very mathematical basis explained above, the frequency domain plot consists of both positive and negative plots of a signal. The negative frequency components shown in spectral plot of signals like DSB, SSB may be thought of as pertaining to mathematical exactness. However, I must mention that besides mere mathematical exactness, the concept of negative frequency (only theoretical it may be) is used by signal processing engineers so as to simplify the signal processing work. Similarly in Doppler radar, the usual convention is that objects moving toward the radar are considered to induce a positive frequency, and objects going away are considered to induce a negative frequency.

I would like to conclude by saying that "negative frequency" is mathematically a valid concept, but it's not physically realizable. Finally, I leave you with a short story about negative frequency:

The story went something to the effect that a ham radio amateur had taken his ham radio receiver and rewound all of the coils, the oscillator coils and the IF transformers, in reverse direction in hopes of making a *negative frequency receiver.*

When he finished and turned the thing on, there was no reception at all. He ascribed this to having created a receiver tuned to negative frequencies on which nobody was transmitting. At least not yet!

He was delighted to have apparently doubled the usable spectrum of available frequencies for radio communications.

(This story was published in the American Radio Relay League's magazine, QST, written under the pseudonym "Larsen E. Rapp on an April Fool's day.) ■

...negative frequency components shown in spectral plot of signals like DSB, SSB may be thought of as pertaining to mathematical exactness

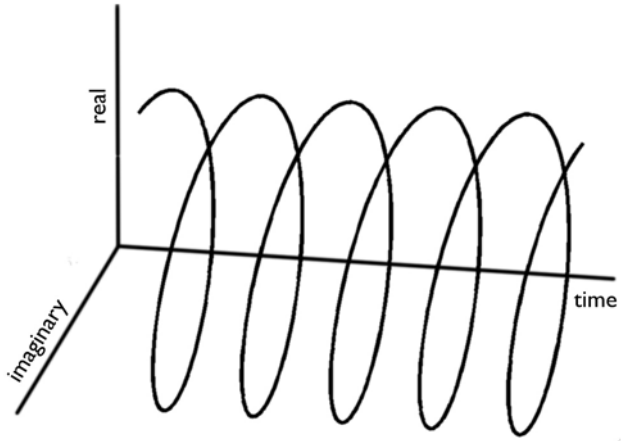


Figure 3.A parametric plot of real versus imaginary part of the signal

References

1. B.P. Lathi, *Modern Digital and Analog Communication Systems*,Oxford University Press,1998
2. Center for Computer Research in Music and Acoustics (CCRMA), Stanford University
3. EDABoard Discussion Forum
4. Mr. John Dunn, *Twysted Pair.com bulletin board.*
5. http://en.wikipedia.org, *Wikipedia*

The art of flanging

Ayush Shrestha
2060 Computer

Warm beats, funky sounds and enigmatic playing skills have always been gracefully rewarded by the world in the field of music. Amidst all these, we mostly focus on the musicians and tend to ignore the persisting gadgets and the technology behind the making of any music. There is a Latin proverb, *"Vinum et musica laetificant cor"* which means wine and music delight the heart, which I firmly acquiesce. Since I am not a connoisseur of wine and this is a technical magazine, the following text will focus more on the technical aspects of creating music; the art of flanging to be specific.

Taking a peek into the history, the classic flanging effect is believed to have been first perfected during 1966 by George Chkiantz, an engineer employed at Olympic Studios in Barnes, London, although it can be heard in The Big Hurt by Toni Fisher which rose to #3 in the Billboard chart in 1959. One of the first instances of the sound being used on a commercial pop recording was the Small Faces' 1967 single Itchycoo Park, recorded at Olympic and engineered by Chkiantz's colleague Glyn Johns. John Lennon of the Beatles used the term 'flanging' to refer to automatic double tracking, a technique developed at Abbey Road Studios by recording engineer Ken Townshend, in answer to producer George Martin's joking assertion that the ADT effect employed a "double-bifurcated sploshing flange". This usage of the term is coincidental. Standard flanging was used on the Beatles song "Blue Jay Way", written and sung by George Harrison.

So what actually is flanging?

Flanging is a time-domain based audio effect that occurs when two identical signals are mixed together, but with one signal time-delayed by a small and gradually changing amount, usually smaller than 20 ms (milliseconds). This produces a swept 'comb filter' effect: peaks and notches are produced in the resultant frequency spectrum, related to each other in a linear harmonic series. Varying the time delay causes these to sweep up and down the frequency spectrum.

Part of the output signal is usually fed back to the input ('re-circulating delay line'), producing a resonance effect which further enhances the intensity of the peaks and troughs. The phase of the fed-back signal is sometimes inverted, producing another variation on the flanging sound. Flanging has a very characteristic sound that many people refer to as a "whooshing" sound, or a sound similar to the sound of a jet plane flying overhead.

Is it similar to phasing?

Flanging is generally considered a particular type of phasing. In case of phasing, a signal is passed through one or more all-pass filters which have non-linear frequency phase response. This results in phase differences in the output signal that depend on the input signal frequency. When used with multi-frequency signals like music, various frequencies in the original signal are delayed by different amounts, causing peaks and troughs in the output signal which are not in a linear harmonic series.

By contrast, flanging relies on an overall uniform time delay to the entire signal, which is equivalent to phasing as described above but with a filter that has a linear phase response across the frequency spectrum. The result is an output signal with peaks and troughs which are in a linear

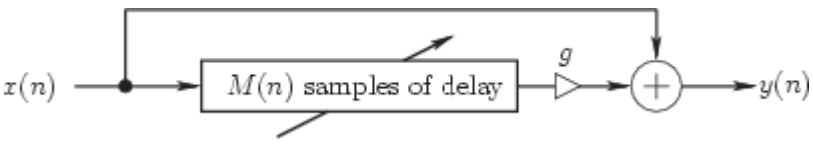


Figure 1: The basic flanger effect

harmonic series. To the ear, flanging and phasing sound similar, yet they are recognizable as distinct colorations.

More on its working

Flanging is created by mixing a signal with a slightly delayed copy of itself, where the length of the delay is constantly changing. Most modern day flangers lets you shape the sound by allowing you to control how much of the delayed signal is added to the original, which is usually referred to as a 'depth' control.

Flanging is modelled quite accurately as a feed-forward comb filter, in which the delay *M* is varied over time. Figure 1 depicts such a model. The input-output relation for a basic flanger can be written as

$$y(n) = x(n) + gx[n - M(n)]$$

where *x(n)* is the input signal amplitude at time *n = 0,1,2 …, y(n)* is the output at time *n, g* is the 'depth' of the flanging effect, and *M (n)* is the length of the delay-line at time *n*. Since *M (n)* must vary smoothly over time, it is clearly necessary to use an interpolated delay line to provide non-integer values of *M* in a smooth fashion.

As shown in Fig. 2, the frequency response has a 'comb' shaped structure.

For *g>0*, there are *M* peaks in the frequency resp

$$\omega_k^{(p)} = k\frac{2\pi}{M}, \quad k = 0, 1, 2, \ldots, M-1.$$

For *g*=1, the peaks are maximally pronounced, with *M* notches occurring between them at frequencies:

$$\omega_k^{(n)} = \omega_k^{(p)} + \pi/M$$

As the delay length *M* is varied over time, these 'comb teeth' squeeze in and out like the pleats of an accordion. The notches are spaced at intervals of *fx* /*M* Hz, where *fx* denotes the sampling rate. In particular, the notch spacing is inversely proportional to delay-line length.

…flanging was actually "discovered" by accident. Legend says it originated while the Beatles were producing an album.

The time variation of the delay-line length *M (n)* results in a 'sweeping' of uniformly-spaced notches in the spectrum. The flanging effect is thus created by moving notches in the spectrum. Notch motion is essential for the flanging effect. Static notches provide some coloration to the sound, but an isolated notch may be inaudible. Since the steady-state sound field inside an undamped acoustic tube has a similar set of uniformly spaced notches (except at the ends), a static row of notches tends to sound like being inside an acoustic tube.

These notches in the frequency response are created by destructive interference. Picture a perfect tone - a sine wave. If you delay that signal and then add it to the original, the sum of the two signals may look quite different. At one extreme, where the delay is such that the signals are perfectly out of phase, as one signal increases, the other decreases the same amount, so the entire signal will disappear at the output. Likewise, the two signals could still remain in phase after the delay, doubling the magnitude of that frequency (constructive interference). For any given amount of delay, some frequencies will be eliminated while others are passed through. In the flanger, you can control how deep these notches go by using the depth control. When the depth is at zero, the frequency response is flat, but as you increase the depth, the notches begin to appear and extend downward, reaching zero when the depth is one. Even if the notches do not extend quite all the way to zero, they will still have an audible effect.

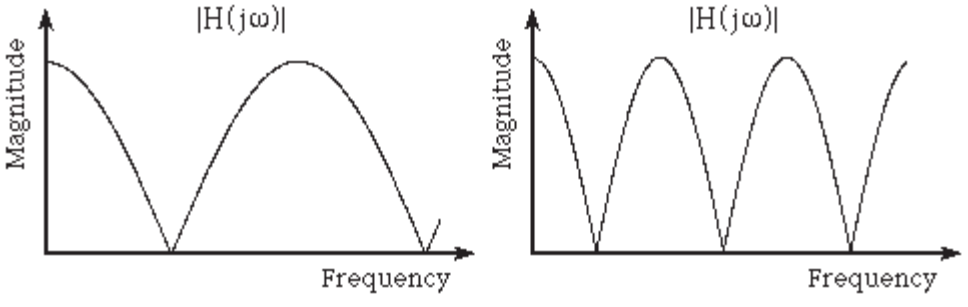


Figure 2: The frequency response of a simple flanger with two different delay times.The plot on the left would be for a flanger with a smaller delay than that on the right.

The characteristic sound of a flanger results when these notches sweep up and down the frequency axis over time. Picture the notches compressing and expanding like a spring between the two plots in Figure 2. The

sweeping action of the notches is achieved by continuously changing the amount of delay used. This changing of the delay in the flanger creates some pitch modulation i.e., the perceived pitch 'warbles'.

As the delay increases, the notches slide further down into the lower frequencies. The manner in which the delay changes is determined by the LFO (Low Frequency Oscillator) waveform

The flanged sound does not sound like an echo because of very short delays. In a flanger, the typical delay times are from 1 to 10 milliseconds (the human ear will perceive an echo if the delay is more than 50-70 milliseconds or so). Instead of creating an echo, the delay has a filtering effect on the signal, and this effect creates a series of notches in the frequency response, as shown in Figure 2. Points at which the frequency response goes to zero means that sounds of that frequency are eliminated, while other frequencies are passed with some amplitude change. This frequency response is sometimes called a comb filter, as its notches resemble the teeth on a comb.

Use of flangers

While using a flanger, for non-percussive instruments that generate a pitch and the harmonic overtones, the notches that the flanger produces could in theory coincide exactly with the fundamental and the overtones, eliminating the sound of the instrument altogether. In practice, an instrument won't disappear entirely, but there can be severe amplitude modulation. For this reason, some favor using flangers on percussive and noise-like signals. In fact, flanging is often used on an entire mix, rather than a specific instrument within a mix.

Conclusion

In this dynamic world, though there may have been many developments and enhancements in the gadgets used, the basic theory of flanging remains the same. From the glorious days of the Beatles to this day, many artists in the music industry have made use of flanging and have come up with superb titles which have been highly praised by the audience worldwide. "Blue Jay Way" performed by the Beatles, The bridge of "Life in the Fast Lane" performed by The Eagles, "Paranoid" by Black Sabbath, the main riff of "Unchained", by Van Halen are few songs among many where flanging can be heard explicitly. Be it with some covert underground metal band or some flamboyant world-touring mainstream band we can still expect some breath taking music to be created from this technique or rather say art: the art of flanging. ■

AUTOMATED ENGRAVER

(Senior Level Project)

Prabhat Rai, Sandesh Joshi,
Suraj Karki, Surendra Sedhai
2058 Electronics

Engraving is the process of etching or carving a design or lettering into a hard surface for decoration or printing purpose. Engraving can be done on a variety of surfaces. On glass surfaces high powered sand blasting technique is used whereas for marble or slates the design is etched manually using a hammer and a chisel.

Engraving is done for various purposes, as decoration and printing. In Nepal, glass engraving is done manually using the sand blasting method. First the design to be etched is made on paper. Then using masking technique i.e. keeping the space to be etched open and covering rest of the work piece by some protective cover, we etch the glass. For etching, very fine sand is blown on the glass surface with a powerful sand blower. This blown sand eats away the glass in the unmasked region. Finally we get the desired engraving.

Now this technique is very primitive. The design that we etch with this technique is not uniform, it is hard to maintain uniformity. For better results professional engraving machines are used which are very expensive and has to be imported from aboard. We could find only a handful of engraving machines during our project survey. Our engraver provides the same service at a much lower cost. Much to the operator's relief the process of making the mask manually, blowing the sand, maintaining uniformity etc. will be completely eliminated. With our automated engraver all you have to do is feed the design to the engraver. Once the design is fed to the engraver it will automatically engrave the design on the work piece. The engraver will work on the basis of point plotting. The resolution is 0.5mm. Stepper motors are used to control the plotting and positioning. DC motors with feedback circuit can be used for controlling but the circuitry is very complex and expensive. For automation Atmel AT89C51 microcontroller is interfaced with the required transducers and control devices. Microcontroller is suitable for repetitive task and is efficient to work as CPU, RAM, ROM, input/output ports, timer, interrupts, and serial ports are all on a single chip. It is better to use microcontroller instead of using the computer in order to reduce the workload on CPU of computer, which can be used for carrying out other operations. Further, nowadays, these microcontrollers can be reprogrammed if problem arises in the existing operation or if the application has to be modified.

The functional block diagram of the automated engraver is shown figure 1.

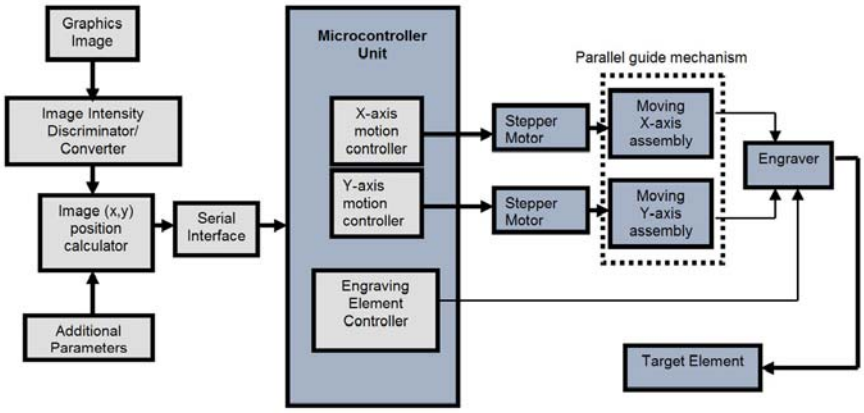


Figure 1. Overall block diagram of the automated engraver

The hardware section consists of mainly four units. They are:

(i) X-axis positioning unit

X-axis positioning unit consists of a spur-gear arrangement which moves a solid metal shaft. The two ends of this shaft are wound with a few turns of inextensible string. Each end of the string is fixed to the ends of the base board. Therefore when the shaft rotates, it winds over the string and pulls itself and the whole assembly towards the required direction.

Due to inherent friction coupling between string and shaft there is always some chance of slip which we try to minimize by changing the string material. We also have added rubber for improved friction so that the slip could be further reduced. The motion of stepper motor is in short discrete steps. Therefore inertia of the whole mechanism plays a significant role in the settling time and the overshoot of the system.

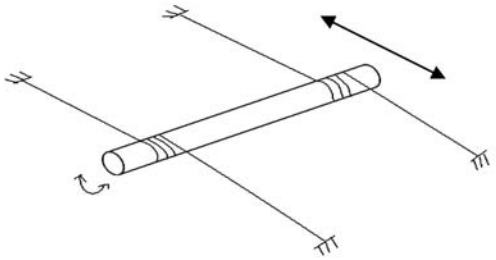


Figure 2. X-axis positioning unit

(ii) Y-axis positioning unit

This Y-axis positioning unit rides over the X-axis positioning assembly on two parallel shafts and carries the striker unit. The motion is obtained directly from the stepper motor without any reduction. A timing belt is attached to the carriage, the Y-axis positioning unit, and is passed over a matching pulley of the stepper motor. The timing belt helps to avoid any slip during the motion in the Y-axis. Since there is no reduction in gears, the resolution in Y-axis is inversely proportional to the pulley diameter. If we require higher resolution then we will use smaller pulleys and if we require low resolution then we can opt for larger pulleys. The belt is suitably stretched using a spring, such that there is no sag/bend in any portion of the belt.

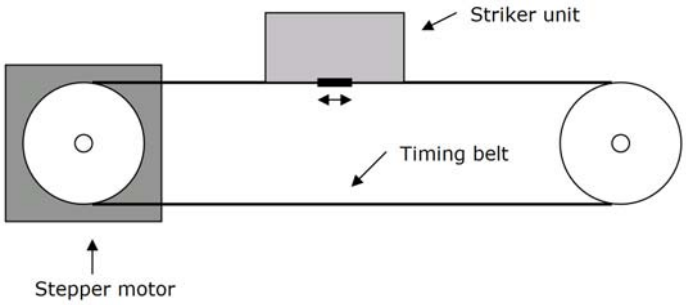


Figure 3.Y-axis positioning unit

(iii) Engraver unit

The engraver unit is mounted on the carriage which rides on the Y-motion belt and consists of an electromagnet and a hammering mechanism. The electromagnet is energized by a half rectified current. The hammering mechanism is made of a block of soft iron which is linked to a small shaft. This shaft is fitted with a sharp pointed tip at the lower end. This point is the engraving tool of the machine. When the electromagnet is energized, it attracts the soft iron which in turn hits the shaft. This causes a small dot to be engraved on the object. The spring is inside the shaft which will pull back the hammering unit once the electromagnet is demagnetized during the relaxation period. For smooth operation of the engraver we first have to calibrate and adjust the length of the tool tip and the clearance. This has to be kept in mind when we have work pieces with various thicknesses. If enough clearance is not allocated,

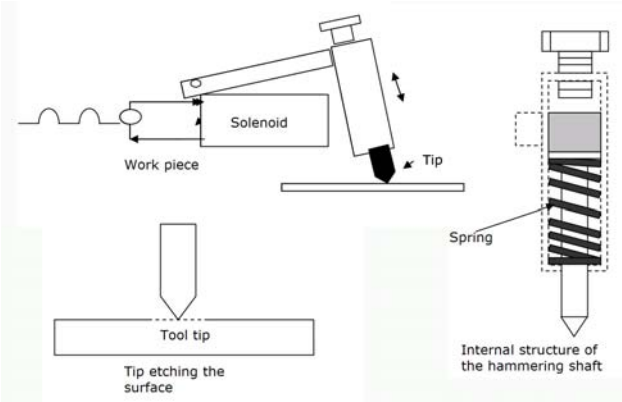


Figure 4. Engraver Unit

it may damage the work piece ruining the design. For setting up the hammering unit we have provided two screws. The screw on the side of the shaft holds the hammering tip while the screw on the bottom of the shaft is for adjusting the length of the screw.

(iv) Parallel guide mechanism

Parallelism is a strict requirement to the making of geometric figures. This is required in both X and Y directions. For parallelism in Y-axis, there are two parallel rods over which the striker unit slides. For X-axis parallelism there is a thread mechanism similar to that used in architect's drawing boards. This assembly provides a simple and effective means to keep the segments aligned to initial axes. Two strings are passed over pulleys fixed at the bottom of the carriage. The two strings are fixed at the two ends of the base.

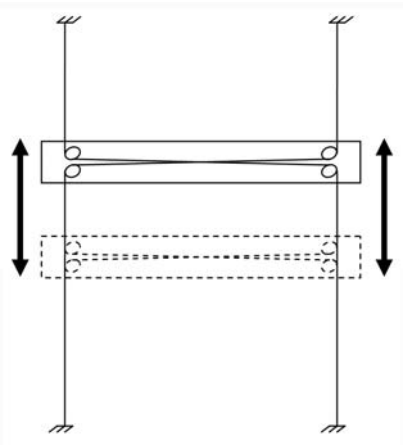


Figure 5. Parallel guide mechanism

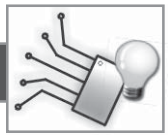
Improvements and Modifications

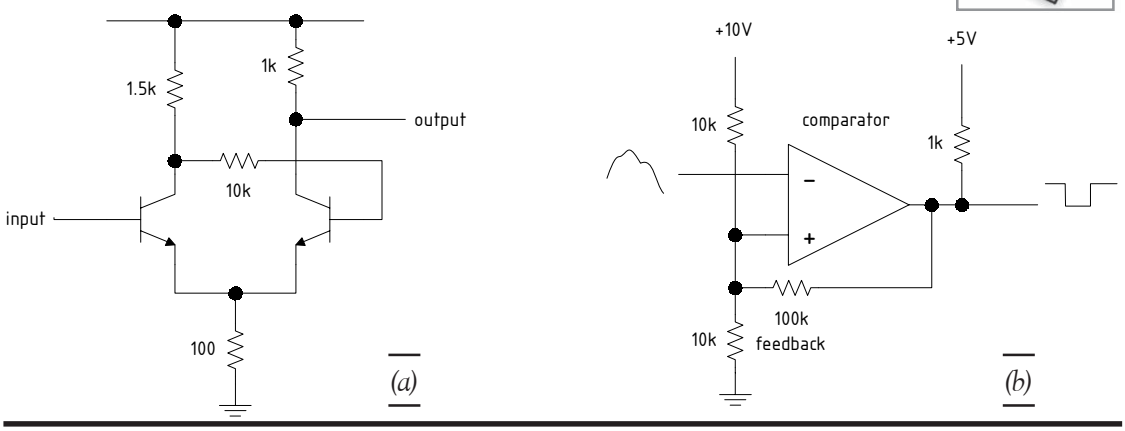
Many improvements can be done in this existing project. The whole hammering assembly is quite inadequate to cope with frequent use. After every job the assembly has to be tightened to make it immovable as even a pixel of deviation could have adverse effect on the result. But due to the lack of funds, a better module could not be developed though many different possibilities were thought of.

The same parameters may not be suitable for different work piece. For this the tip has to be adjusted manually. So, this can be automated using sensors in conjunction with the tool tip. Once the engraver senses that a work piece has been laid, the tip can automatically detect the thickness of the piece and can adjust the clearance itself. The hammering module we have used is electromagnet and spring combination. This module works fine but solenoid has slow switching time. Due to the slow switching time we face problems with the hammering.

The hammering module could be replaced with a milling module. In milling the etching is done by a rotating head which cuts in the direction perpendicular to the axis of its rotation. But the tool tip for milling is very expensive and the mechanism very complicated though the output is far better as we will be able to engrave not only black and white but also grayscale images on the work piece. ■

circuit ideas





Schmitt Trigger; using discrete components (a) , using a comparator (b)

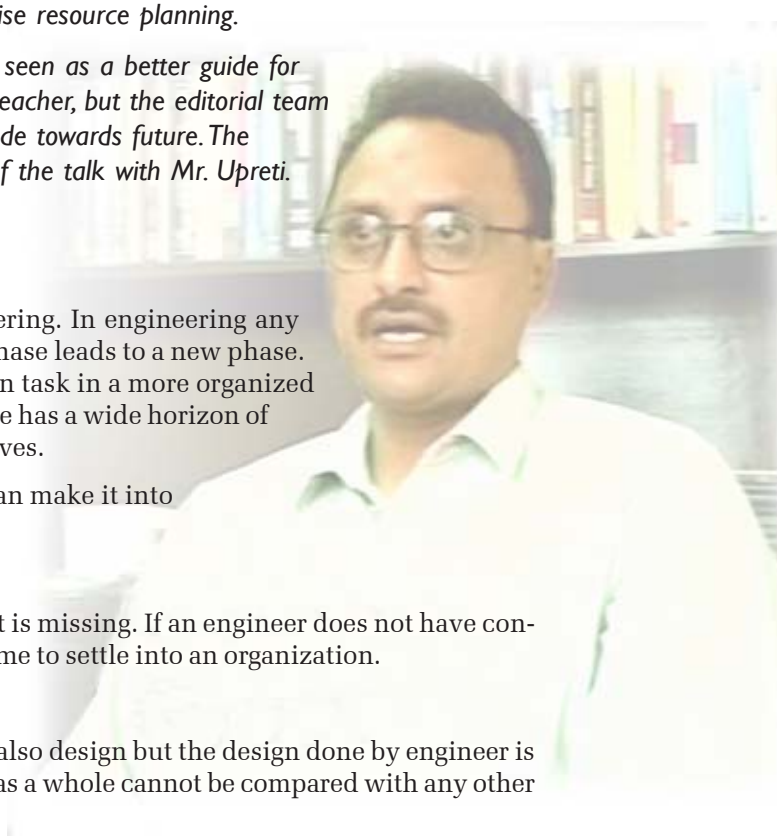
An Interview with Tika Upreti

Know Mr Upreti

Mr. Tika Upreti, a faculty member of DOECE is one of the most dynamic person in the department. Mr. Upreti received his BE in Electrical Engineering from NED University, Pakistan, in 1990 and completed his Masters of Science degree in Computer science from the University of Calgary, Canada, in 1996.

Besides being an inspiring teacher, he has gained significant experience in software development, business process outsourcing, networking and enterprise resource planning.

With experience from professional field, Mr. Upreti is seen as a better guide for students at DOECE. Many see Mr. Upreti as a strict teacher, but the editorial team found him to be pragmatic person with positive attitude towards future. The editorial team of Zerone 2005 presents an excerpt of the talk with Mr. Upreti.



What is Engineering?

The systematic implementation of any work is engineering. In engineering any work is done in a phase wiser manner. Success in one phase leads to a new phase. An engineer has a certain perspective of doing any given task in a more organized way. With his knowledge on mathematics and science, he has a wide horizon of thinking and can analyze things from different perspectives.

A person with good base in mathematics and physics can make it into engineering field.

Where does B.E stand among other IT degrees?

B.E more hardware/software oriented. Management part is missing. If an engineer does not have concepts of management, in his career, he will need more time to settle into an organization.

Is it a must to do BE to be an engineer?

Yes. Engineers are groomed in a set pattern. Others may also design but the design done by engineer is different. It is more cost effective and solid. Engineering as a whole cannot be compared with any other field.

How do you differentiate between electronics and computer engineer?

Both electronics and computer engineering are growing sectors and I specifically don't differentiate between these two fields. If knowledge on database and object oriented programming is fulfilled, an electronics engineer can work as computer engineer. Similarly for a computer engineer to work as electronics engineer the primary focus should be hardware.

What is the future of Computer/Electronics engineering in Nepal?

As far as I see, it is very bright. Once the peace and political stability is restored, the economic state of Nepal will rise up. Due to the continuing violence in the country, many people are going abroad to work. But once there is peace, I believe they will return to the country and work here. Once peace is established, you will sell lots of outsourcing firms set up in Nepal.

There is no need to get depressed with the situation right now. Looking at it positively, it is the time to hibernate. I mean, we should utilize this time to gain more knowledge so that as and when peace is restored, we can readily start the work.

> *"Once the peace and political stability is restored, the economic state of Nepal will rise up"*

We are in between two huge economies. China is a hardware giant and India will become a software giant. Economic growth is like conveyor belt in a factory. In order to revive our economy, all Nepal has to do is just place itself on the conveyor belt, rest will be taken care of.

We cannot compete with China in hardware. But our English is better than theirs, and we can compete with China in this term, for outsourcing jobs. Similarly, in case of India we can compete in terms of cheaper labor cost.

What changes would you like to see in the engineering courses?

The changes in the engineering courses take place very slow here. The course should be market driven. The course should be designed by the professionals and not by the professors. Experts from various fields should be consulted while designing the course.

Career?

I did my undergraduate studies from Pakistan in 1990. Back then I always wanted to be a VCR mechanic. I used to travel 1 hour in bus and walk 20 minute further to go to a mechanic for training. Once I reached my 3rd year of study, I did lots of hardware project, and then I realized that, repairing job was not for me. Now I wanted to design those systems myself. Back then computer had slowly started to pick up the market, and I realized that I must do something with computers. My university did have computers, but unlike today, we had to go through three doors before we could see one. I never learnt computers in my college days. But I bought my own system in 1987 and started learning about it myself.

After I returned to Nepal, I opened a computer repair workshop at Putalisadak. At the same time, I started teaching at IOE in Diploma level. Back then computer systems were very difficult to maintain. We had to perform low level hard disk format and transfer operating system on to it. I had an oscilloscope and even did chip level repairs of XT systems. Beyond that, the systems started becoming very complex. The concept of repairing was waning out.

It was difficult to sustain the workshop in the beginning. It was difficult to gain people's trust. But with God's grace and my hard work I succeeded.

Your perspectives on digital divide?

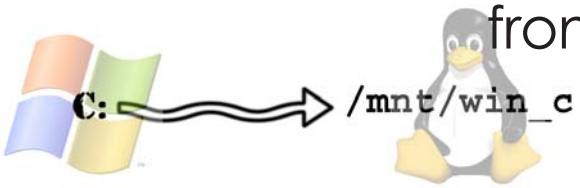
It is just the case of computer's haves and have nots. I do not think digital divide can last long. In early days, any new technology had to be brought in by someone. But nowadays if you see, the technology itself comes to you. Just recently, there have been talks about providing laptops in $100. With such concepts coming up, digital divide will soon narrow down.

Final words of encouragement...

Every one should have a goal in their mind to achieve something in their life. The goal should be fixed. For instance, you cannot take a good picture if the object is moving; only a stable object gives good picture. Exactly, only a fixed goal will lead you to success. You should try to specialize in one particular field. While choosing your field of specialization, you should be either 'interest driven' or 'market driven'. I personally prefer 'interest driven' aims. Talking about the scope, it all depends on oneself. If you have the knowledge, the scope will be created automatically. Regarding the engineering course, the current one is stagnant. The course should be 'market driven', I mean the course should be able to serve the needs to the market.

> *"the scope, it all depends on oneself. If you have the knowledge, the scope will be created automatically"*

Accessing your **Windows** Partitions from Linux



Sudeep Shakya
2059 Electronics

If you are new to Linux then you may be wondering where your C:, D: or E: drives are in Linux. Well there is no such thing as drive letters in Linux. To access your windows partitions you will first have to mount them. Mounting a partition on to a folder is like assigning a drive letter. After you have mounted your windows partitions you can go about your business as usual.

This article assumes you have Linux in a Dual-boot scenario with Windows. You should also know how to open a Console and knowledge of a few common commands will be helpful. For the entire length of the article it would be easier for new users to login as root or if you are a bit experienced you can use the "su" [switch user] command.

Step 1: Checking Filesystems supported by your running kernel

Linux supports the FAT [FAT, FAT16 & FAT32] filesystem but NTFS may or may not be supported by you distribution. To find out if ntfs is supported check if "ntfs" is listed in "/proc/filesystem" file. You can either graphically browse to the "/proc" directory and open the "filesystem" file or you can use the following command in a console.

```
# cat /proc/filesystem
```

If ntfs is listed in that file or if you are not concerned about ntfs partitions you can just skip step2. The rest will have to download and install the ntfs filesystem driver.

Step 2: Downloading and installing the correct ntfs drivers

- Find out the version number of your kernel
  ```
  # uname -r
  ```
  Note down this number. For Fedora Core 3 its "2.6.9-1.667"
- Go to http://linux-ntfs.org and download the ntfs driver for your distribution and kernel version. Below are few examples.
- kernel-ntfs-2.4.20-9.i586.rpm for Red Hat 9
- kernel-module-ntfs-2.6.9-1.667-2.1.20-0.rr.3.3.i686.rpm for FC3
- kernel-module-ntfs-2.6.11-1.1369_FC4-2.1.22-0.rr.6.0.i686.rpm for FC4
- Install the driver.
  ```
  # rpm -ivh kernel-driver-name.rpm
  ```

There should be no error messages. If there are then you probably downloaded the wrong driver or you didn't give the name of the kernel driver correctly.

Step 3: Finding Partition numbers and filesystems

For a list of all the drives and their partitions enter the following command.

```
# /sbin/fdisk -l
```

This should display the partition no. and their filesystems. Mine is given here.

/dev/hda1 is the device no. of your first partition [generally C: in windows] and W95 FAT32 or NTFS is the filesystem. You can just ignore the partition with the filesystem Extended or W95 Extd. The next partition /dev/hda5 is generally D: drive in windows and the rest follow. Note down the device no of partitions with filesystems fat32 or ntfs. Also there is no need to mount Linux partitions as they will already have been mounted.

Step 4: Creating Mount Points

Mount points are similar to drive letters. In Linux partitions are mounted on empty folders. These folders are called mount points. Create mount points for each of your windows partitions. For /dev/hda1 [C: drive in windows] you could create a folder called "win_c" or anything you like.

Do this for all your other partitions. Though you can create mount points just about anywhere it's customary to create mount points in the "/mnt" folder.

```
Device     Boot  Start  End    Blocks     Id  System
/dev/hda1  *     1      383    3076416    b   W95 FAT32
/dev/hda2        384    9729   75071745   5   Extended
/dev/hda5        384    994    4907826    7   HPFS/NTFS
/dev/hda6        995    2526   12305758+  7   HPFS/NTFS
/dev/hda7        2527   5443   23430771   7   HPFS/NTFS
/dev/hda8        5444   5456   104391     83  Linux
/dev/hda9        5457   5849   3156741    83  Linux
/dev/hda10       5850   6308   3686886    83  Linux
/dev/hda11       6309   6376   546178+    82  Linux swap
/dev/hda12       6377   8289   15366141   7   HPFS/NTFS
/dev/hda13       8290   9729   11566768+  7   HPFS/NTFS
```

Output of /sbin/fdisk –l

Step 5: Mounting your windows partitions

Partitions can be mounted manually or automatically.

Manual Method

To manually mount a partition use the following command format.

```
# mount device_name mount_point –t
   filesystem_type –o options
```

It should look like

```
# mount /dev/hda1 /mnt/win_c –t vfat
   –o defaults
```

You will have to do this each time you boot Linux or you can use the automatic method.

Automatic Method

Open the file "/etc/fstab" and append a few lines to tell your kernel to automatically mount windows partitions at boot time. Do remember to backup the file before editing it. The fstab file is arranged in the following way.

```
device_name   mount_point   fs_type
   mount_options
/dev/hda1     /mnt/win_c    auto
   defaults 0 0
```

e.g.

Placing auto in place of filesystem type forces the kernel to detect the filesystem type ant most of the times it works. If it doesn't work use "vfat" for fat and "ntfs" for ntfs. Similarly add entries for your other partitions.

Note for NTFS

By default NTFS partitions are mounted with root as owner and no permissions to general user. Also NTFS support is read-only. So for NTFS partitions edit your fstab entries to look like the following.

```
/dev/hda6 /mnt/win_d   auto
   defaults,ro,uid=user_name 0 0
```

Your complete fstab should appear similar to the following.

```
# This file is edited by fstab-sync - see 'man fstab-sync' for details
LABEL=/       /           ext3    defaults 1 1
LABEL=boot    /boot       ext3    defaults 1 2
none          /dev/pts    devpts  gid=5,mode=620 0 0
none          /dev/shm    tmpfs   defaults 0 0
LABEL=home    /home       ext3    defaults 1 2
none          /proc       proc    defaults 0 0
none          /sys        sysfs   defaults 0 0
/dev/hda11    swap        swap    defaults 0 0
/dev/hdc      /media/cdrom        auto
   pamconsole,fscontext=system_u:object_r:removable_t,exec,noauto,managed 0 0
/dev/fd0      /media/floppy       auto
   pamconsole,fscontext=system_u:object_r:removable_t,exec,noauto,managed 0 0

#Comment: Below are the windows partitions and their mount points. Only Two windows
   partitions are mounted
/dev/hda1     /mnt/win_c auto     defaults 0 0
/dev/hda6     /mnt/win_d auto     defaluts,ro,uid=sud 0 0
```

sample fstab file

Ajax for developers: Build dynamic applications

Ajax paves the way for better Web applications

Roshan Newa
2059 Computer

"The page-reload cycle presents one of the biggest usability obstacles in Web application development and is a serious challenge for Java™ developers"

- Philip McCarthy, 20 Sep 2005

Ajax is a ground-breaking approach to creating dynamic Web application experiences. Ajax (Asynchronous JavaScript and XML) is a programming technique that lets you combine Java technologies, XML, and JavaScript for Java-based Web applications that break the page-reload paradigm.

Ajax, or Asynchronous JavaScript and XML, is an approach to Web application development that uses client-side scripting to exchange data with the Web server. As a result, Web pages are dynamically updated without a full page refresh interrupting the interaction flow. With Ajax, you can create richer, more dynamic Web application user interfaces that approach the immediacy and usability of native desktop applications.

Ajax isn't a technology, it's more of a pattern -- a way to identify and describe a useful design technique. Ajax is new in the sense that many developers are just beginning to be aware of it, but all of the components that implement an Ajax application have existed for several years. The current buzz is because of the emergence in 2004 and 2005 of some great dynamic Web UIs based on Ajax technology, most notably Google's GMail and Maps applications and the photo-sharing site Flickr. These UIs were sufficiently groundbreaking to be dubbed "Web 2.0" by some developers, with the resulting interest in Ajax applications skyrocketing.

The Ajax Roundtrip

An Ajax interaction begins with a JavaScript object called XMLHttpRequest.

As the name suggests, it allows a client-side script to perform HTTP requests, and it will parse an XML server response. The first step in this Ajax roundtrip is to create an XMLHttpRequest instance. The HTTP method to use for the request (GET or POST) and the destination URL are then set on the XMLHttpRequest object.

Ajax is based on asynchronous mode. When you send that HTTP request, you don't want the browser to hang around waiting for the server to respond. Instead, you want it to continue reacting to the user's interaction with the page and

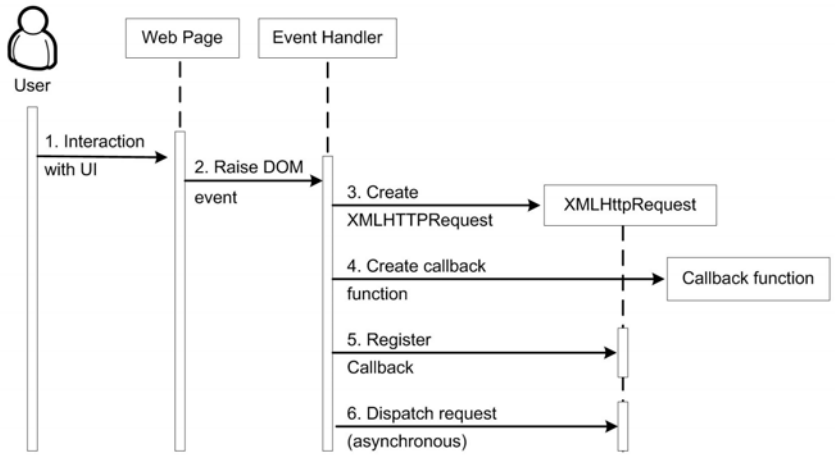


Figure 1. Dispatching an XMLHttpRequest

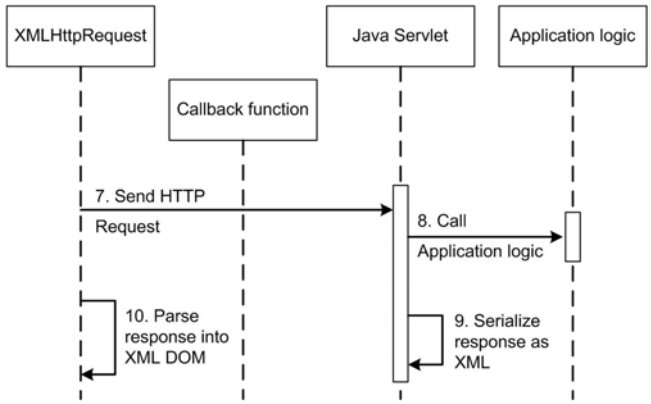


Figure 2. Servlet request handling

deal with the server's response when it eventually arrives. To accomplish this, you can register a callback function with the XMLHttpRequest and then dispatch the XMLHttpRequest asynchronously. Control then returns to the browser, but the callback function will be called when the server's response arrives.

On the Java Web server, the request arrives just like any other HttpServletRequest. After parsing the request parameters, the servlet invokes the necessary application logic, serializes its response into XML, and writes it to the HttpServletResponse.

Back on the client side, the callback function registered on the XMLHttpRequest is now invoked to process the XML document returned by the server. Finally, the user interface is updated in response to the data from the server, using JavaScript to manipulate the page's HTML DOM.

The Ajax roundtrip consists of three operations:

i. Dispatching an XMLHttpRequest

ii. Servlet request handling

iii. Response handling with JavaScript

Dispatching an XMLHttpRequest

The question now is where to start the Ajax sequence: creating and dispatching an XMLHttpRequest from the browser. Unfortunately, the method to create an XMLHttpRequest differs from browser to browser. There are custom JavaScript functions can be used to smooth out these browser-dependent wrinkles, detecting the correct approach for the current browser and returning an XMLHttpRequest ready to use. It's best to think of this as boilerplate code: simply copy it into your JavaScript library and use it when you need an XMLHttpRequest.

Servlet request handling

Handling an XMLHttpRequest with a servlet is largely the same as handling a regular HTTP request from a browser. The form-encoded data sent in the POST request's body can be obtained with HttpServletRequests parameter list. Ajax requests take part in the same HttpSession as regular Web requests from the application. The object that takes part in the ServletRequest is then serialized to XML, and that XML is written to the ServletResponse. It's important to set the response's content type to application/xml, otherwise the XMLHttpRequest will not parse the response content into an XML DOM

Response handling with JavaScript

Ajax uses the event handler thread in JavaScript to process the Response. There is a readyState property of XMLHttpRequest which is a numeric value that gives the status of the request's lifecycle. It changes from 0 for "uninitialized" through to 4 for "complete." Each time the readyState changes, the readystatechange event fires and the handler function attached via the property is called. It is the job of handler function to return a function that checks whether the XMLHttpRequest has

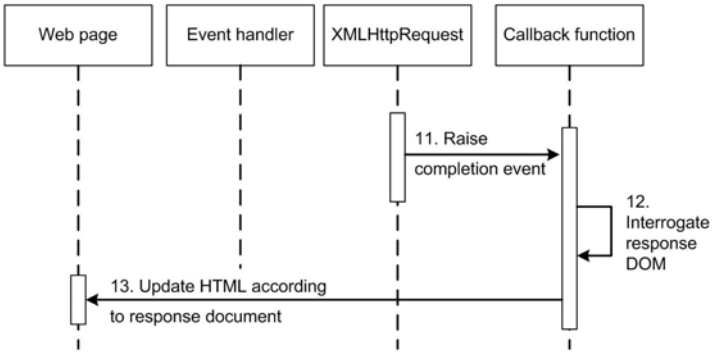


Figure 3. Response handling with JavaScript

completed and passes the XML response onto the handler function specified by the caller.

Challenges of using Ajax

As with any technology, there are plenty of ways to make mistakes with Ajax. Some of the problems here currently lack easy solutions but will improve as Ajax matures. As the developer community gains experience developing Ajax applications, best practices and guidelines will be documented.

Availability of XMLHttpRequest

One of the biggest issues facing Ajax developers is how to respond when XMLHttpRequest isn't available. While the majority of modern browsers support XMLHttpRequest, there will always be a minority of users whose browsers do not, or whose browser security settings prevent XMLHttpRequest from being used. If you're developing a Web app to be deployed on a corporate intranet, you probably have the luxury of specifying which browsers are supported and assuming XMLHttpRequest is always available. If you're deploying on the public Web, however, you must be aware that by presuming XMLHttpRequest is available, you are potentially preventing users of older browsers, for people with disabilities, or lightweight browsers on handheld devices from using your application.

Usability concerns

Some of the usability issues surrounding Ajax applications are more general. For instance, it can be important to let users know that their input has been registered, because the usual feedback mechanisms of the hourglass cursor and spinning browser "throbber" do not apply to XMLHttpRequests.

Another issue is that users may fail to notice that parts of the page they're viewing have been updated. You can alleviate this problem by using a variety of visual techniques to subtly draw the user's eye to updated areas of the page. Other issues caused by updating the page with Ajax include "breaking" the browser's back button, and the URL in the address bar not reflecting the entire state of the page, preventing bookmarking.

Server load

Implementing an Ajax UI in place of a regular forms-based one may dramatically increase the number of requests made to the server. For instance, a regular Google Web search causes one hit on the server, occurring when the user submits the search form. However, Google Suggest, which attempts to autocomplete your search terms, sends several requests to the server as the user types. When developing an Ajax application, be aware of how many requests you'll be sending to the server and the resulting server load this will cause. Buffering requests on the client and caching server responses in the client, where applicable. The best practice is to design your Ajax Web applications so that as much logic as possible can be performed on the client, without needing to contact the server.

Dealing with asynchrony

It's very important to understand that there is no guarantee that XMLHttpRequests will complete in the order they were dispatched. Indeed, you should assume that they will not and design your application with this in mind.

Ajax is new in the sense that many developers are just beginning to be aware of it, but all of the components that implement an Ajax application have existed for several years.

In conclusion

Creating a successful Ajax application requires a in-depth approach from UI design through JavaScript design to server-side architecture. There's good news if you're feeling daunted by the complexity of writing a large Ajax application using the techniques demonstrated here. Just as frameworks like Struts, Spring, and Hibernate have evolved to abstract Web application development away from the low-level details of the Servlet API and JDBC, so toolkits are appearing to ease Ajax development. Some of these focus solely on the client side, providing easy ways to add visual effects to your pages or streamlining the use of XMLHttpRequest. Others go further, providing means to automatically generate Ajax interfaces from server-side code. The Ajax community is fast moving, and there's a great deal of valuable information out there. ■

References

1. http://www.java.sun.com
2. http://bpcatalog.dev.java.net/nonav/ajax/
3. http://www.ajaxpatterns.org
4. http://www.ibm.com/developerworks/java/library/

Look and Feel

Ashay Thakur
2060 Electronics

If Solitaire appeals more to you than Doom 3, Quake 4, Call of Duty 2 and the like, please skip over to the next article. This one could induce sleep. Of course if you are one of those freaks who spend big bucks on oodles of RAM and a top-of-the-line graphics card, you are welcome to stay.

Just visualize your favorite 3D game. Now think of the developer who scripted it. The textures and shadows and lights and water effects and what not. Games today come with mesmerizing scenery. Add to this the physical properties that the objects in the game need to exhibit. There is a maddening amount of mathematics, coding and rendering that the graphics chip and CPU have to do. No wonder these games are resource hogs- one needs a 3 GHz+ machine coupled with a graphics card beyond the Rs. 35000 bracket to run the new titles.

Now visualize if developers were asked to code each and every command, each and every difficulty level, and for each video resolution and effect! Think of the mass suicides in Software houses.

Thankfully though, a software coder's life is not as miserable. It is made easier with the help of APIs (Application Programming Interfaces), which standardize commands that are sent to the hardware so that a developer just needs to know what command his software is supposed to send, regardless of the hardware or software in use. Hence an API allows both developers and hardware manufacturers to standardize the commands, code and drivers. We, in this article are not going to elucidate the concept of APIs as a whole but only those that are used for 3D graphics- whether it be for Maya and 3ds Max or Doom and F.E.A.R.

A 3D graphics API has input software, the OS libraries and hardware device drivers. So, it takes commands from a graphics program and translates the input requirements and output data to the hardware. Major players in this market are OpenGL and DirectX (Direct3D, more like).

OpenGL started off the docs as Iris GL, a graphics API developed by Silicon Graphics Incorporated (SGI), for the UNIX operating system. In 1992, SGI renamed it to OpenGL and made it available for public use. OpenGL stands for 'Open Standard Graphics Library'. It is popularly considered to be open source but it is not. Anyone can license the API but cannot modify it- a stark contrast to its billing as open source.

There is a maddening amount of mathematics, coding and rendering that the graphics chip and CPU have to do

The best feature of OpenGL is how it seamlessly integrates with almost any operating system, be it Unix Linux or Windows. There is an official open source implementation of OpenGL called Sample Implementation.

OpenGL began as and remains to this day, a low-level graphics library specification. It deals with primitive geometry such as lines, points, simple polygons, images and bitmaps to create larger 2D and 3D rendered images. Using OpenGL, a programmer can always give simple commands to render these shapes. Originally this API was designed to be used with C and C++, but currently supports Java, FORTRAN, ADA and practically each language that you can recall.

Given that OpenGL is independent of the operating system's windowing system, it may be used on any platform. A windowing system handles window management, colour mapping and event handling. All this results in simple and short codes that are easy to debug or change.

Talk of DirectX and most will put their hands up in acknowledgement. However Direct3D has many dumbfounded. DirectX is an API suite that has audio, input and graphics all rolled into one. Direct3D is the Graphics API within DirectX. Back in 1995, all APIs and consequently games were written with DOS in mind. Windows made endeavours to shift the native focus of games to Windows; easier said than done, though. Windows has many abstraction layers, and coding became a nightmare due to this. In 1995 though,

Microsoft bought off Reality lab from RenderMorphics. This 3D rendering software was later called Direct3D. However the beginning was as rocky as it was for IE against Netscape.

OpenGL was well established by the time Direct3D came to the market with its version 1.0, which was a huge failure. Coding was much better and OpenGL was less buggier. So Microsoft could have been forgiven for having shelved the project but they persisted and by the time they got to version 7.0, it had been accepted by a considerable section of developers. With DirectX 8 and above, though, a set of code classes were included as common files, which made coding simpler and shorter.

Most important is the way that DirectX has standardized the graphics card manufacturers and developers. Basically, you can now play the latest games on a DirectX compliant card- the resolutions will be lower on an older card, but at least you can play.

Another and a more intriguing fact is that DirectX has Unified Shader Structure. Hence the graphics chipset can use the same transistor for pixel or vertex shading. This gives DirectX a definite edge over OpenGL.

However there are obvious faults on the DirectX side as well. First, it Microsoft based. So, it will listen to suggestions that much slower.

Although Windows is by far the most popular OS in the world, not all programmers like their games to be limited to one OS. id Software, for instance, uses OpenGL and makes games for all OSes.

However, improvements in DirectX API and enhanced integration with Windows make it a safe bet and designers do not consider investing time and money for games on other platforms a viable option. These people target the masses by coding for Windows.

Most open source developers use OpenGL and game developers use DirectX. So long as there are more than one OSes both can co-exist. For Sony's PlayStation series OpenGL is being used. However with the popularity of Microsoft Xbox, Developers seem to stick with DirectX giving it an enviable presence in the console market.

The future, in terms of PC gaming, could pose some problems to OpenGL though. Especially important in this regard is the soon to be released Microsoft Vista, which uses 3D graphics engine even to display its regular Desktop. Obviously, Vista will use DirectX for this purpose. But the engine is now called Windows Graphics Foundation (WGF). The terrible news for OpenGL is that Vista might run its own version of OpenGL (as 1.4) in a layer above WGF.

Though this is mere speculation, if it happens to be true, game developers will have to code in this new API. The major concern today is whether or not OpenGL will survive this onslaught. Another interesting thing is that Microsoft wants to improve the way games are installed and played. WGF might enable us to install and play games within minutes. Also, Vista can unload anything that is not required to play a game, be it services or the desktop environment. There is also support for Pixel and Vertex Shader 4.0 so as to future-proof the graphics API. It could make even simple programs to be done with raster images instead of vector images. (Make Programming Difficult!).

Hence the API wars are coming to an end in the near future if one might prophesize. Whatever be the outcome of the API wars, we will be at the receiving end of great games and brilliant visuals. ■

Recovering partition table, *manually*

Jwalanta Shrestha
2060 Computer

Losing partition can be a real nightmare. Trust me, even losing a processor won't be that nerve chilling. And to make things worse, partition damage can happen to *your* hard disk too. Partitions can disappear due to various reasons. Sometimes, viruses are the cause, sometimes they may get corrupted due to software crash or while attempting an OS installation. And sometimes, partitions just get lost bizarrely on the next restart of computer!

Don't hold your breath though. Losing a partition doesn't necessarily mean all of the data are gone. If partition is lost due to a software crash or any such short term event, it can be assumed that all the data is still there intact. So, if you have a little knowledge of partition table structure and dare to mess around with your disk in byte level, partition recovery is within your reach.

Tools

There are a few tools needed for a manual partition table recovery. Obviously you'll need a disk editor. For DOS, Norton Disk Editor (included in Norton Utilities) is good enough. For Windows, WinHex (X-Ways Software) is recommended. You may also need a boot diskette and a calculator (for offset calculation and binary-decimal conversion).

Before moving on to partition recovery stuffs, let's first get to know some of the basics.

The Master Boot Record

Master Boot Record (MBR) is located at the first sector ie sector 0 (note that, 1 sector = 512 bytes) of the hard disk which consists of two things - boot code and partition table. The structure is given in Table 1.

Offset	Nature	Size
+00h	Executable code	may vary
+1BEh	1st partition table entry	16 bytes
+1CEh	2nd partition table entry	16 bytes
+1DEh	3rd partition table entry	16 bytes
+1EEh	4th partition table entry	16 bytes
+1FEh	Executable marker 55AAh	2 bytes

Table 1. Structure of Master Boot Record (MBR)

Table below is partition table entry structure.

Offset	Nature	Size
+00h	Partition State	
	00h = non active	
	80h = Boot Partition	1 byte
+01h	Begin of partition : Head	1 byte
+02h	Begin of partition : Cylinder - Sector	2 bytes
+04h	Type of partition	1 byte
+05h	End of partition : Head	1 byte
+06h	End of partition : Cylinder - Sector	2 bytes
+08h	Number of sectors between the MBR	
	and the 1st sector of the partition	4 bytes
+0Ch	Number of sectors in the partition	4 bytes

Table 2: Structure of Partition table

The 2 bytes of cylinder – sector is encoded as,

- Bit 15-8: Cylinder bits 7-0
- Bit 7 & 6: Cylinder bits 9 & 8
- Bit 5-0: Sector bit 5-0

'Type of partition' refers to a 1 byte number, which is different for different file systems. For DOS, it is 06h, 0Ch is FAT32, 07h means NTFS, etc. (Refer http://www.win.tue.nl/~aeb/partitions/partition_types-2.html for the list). It can also have a value 05h/0Fh which refers to 'Extended partition'. Extended partition means that the partition table points to another Extended Partition Pointer (EPP). This EPP can further contain partition table entries or point to other EPPs recursively.

From above information, it can be concluded that the MBR itself can contain at most 4 partitions (aka primary partitions/volumes) or may point to an extended partition pointer, which may further contains other partitions. This is diagrammatically shown in figure 1.

Now let's take an example. Here's what was found on a disk's first sector's 1BEh to 1CDh (first partition):

80 01 01 00 07 EF FF FF 3F 00 00 00 F1 B1 2B 01

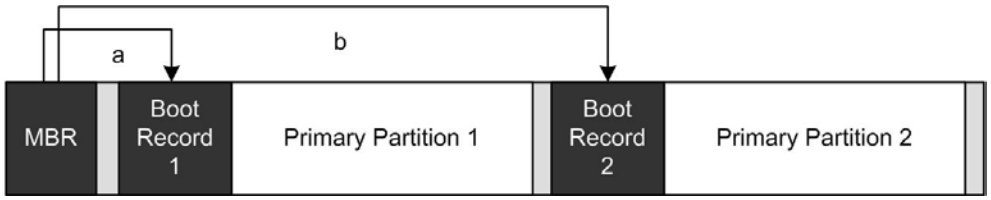
Translation gives table 3,

Partition damage is nothing but corruption of any/some of these 16-bytes partition entries.

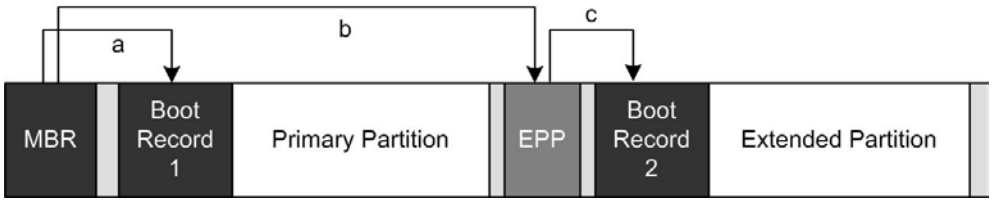
Partition damage scenario and solutions

Boot record backup

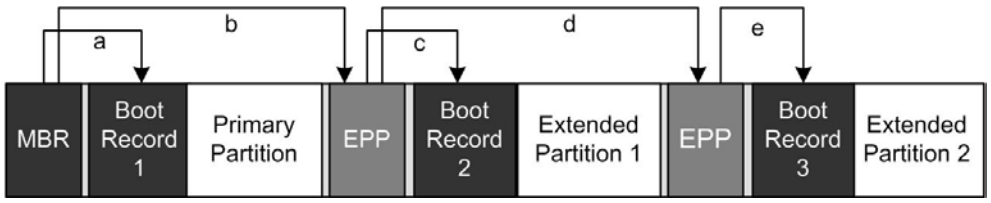
Modern filesystems store a backup of the boot sector somewhere on the volume. FAT32 typically places it into the 6th sector of the volume. On NTFS, the backup copy is stored in the last sector of the volume. So before moving into other recovery procedures, check these sectors. The partition data could still be there, undamaged.



(a) Only primary partitions present



(b) Extended Partition using EPP



(c) Two extended partitions with chained EPP

Figure 1: Partition schemes

State	Begin Head	Begin Sector & Cyl	Type	End Head	End Sector & Cyl	Relative Sector	No. of Sectors
80	01	0001	07	EF	FFFF	0000003F	012BB1F1
Boot Partition	1	Sect=1 Cyl=0	NTFS	239	Sect=63 Cyl=1023	63	19640817

Table 3:Translation

The magical 55AAh

Be it the MBR's, or the boot record's or EPP's sector, it has two bytes 55h and AAh at the end of the sector. While scanning a harddisk, if a 55AAh is found at the end of any sector, that sector could be a boot record or a partition table entry sector. So while searching for a lost partition, this is the number to find.

Slack spaces between partitions

The partitions are placed close to each other. "Slack" (unused) space between them is typically about 64 or 128 sectors (32KB and 64KB respectively). While searching for a lost partition, this slack space can also help to determine the partitions, given the fact that the data in this area usually null. However for partition recovery purposes, it is in most cases safe to disregard this slack space and treat the allocation as contiguous. But take a special note that FAT series filesystems (FAT16 or FAT32) place their metadata very close to the start of the volume. So in this case, recovery process is much more sensitive to lower (start) boundary of the volume than to the upper (high) boundary.

MBR damage

This is the worst case scenario. Since MBR consists the main partition table and has link to other extended partition, recovering a partition when the MBR is gone can be difficult.

Fortunately, if the disk had only one partition, the partition table can be rewritten allocating the whole disk space to a single partition. However, there's one thing to note, boot record's sector doesn't start immediately after the MBR, so the first thing to do would be to find the sectors after the MBR which has 55AAh at the end. But having 55AAh at the end doesn't necessarily mean it's a boot record. To confirm whether this is boot record or not, search for strings like 'FAT32', 'NTFS', etc in this sector. Volume name can also be found there. In most cases, the first partition starts at the sector 63, so this is the first place to look for a valid boot record. Try sector 69 (6th sector of the volume) thats where the backup boot record resides.

If the disk had multiple partitions, most of the time, it is done using extended partition pointers. So, for multiple partitions, the MBR has one entry for the first partition and an entry for the EPP. This EPP contains one entry for second partition and an entry for another EPP, and so on. In this case too, the 55AAh strategy would be appropriate.

Losing a partition doesn't necessarily mean all of the data are gone.

MBR Ok, other partitions lost

In this case, the disk is bootable but the partitions other than the first are missing (consider link (b) damaged in figure 1c). This means that the pointer to EPP in the MBR is damaged. Since the first partition is okay, the end sector of this partition can be calculated from the first partition entry in the MBR. Now doing 55AAh search after that sector could get you to the EPP. If this EPP is fine, calculating the values specified in table 2 for this sector and writing it to the MBR would bring the lost partitions back.

Some partitions-in-between lost

This may be due to damage in partition table entry, while the EPP pointers are ok. For example, link (c) damaged in figure 1c. The recovery procedure is same as the previous cases.

Partitions are ok, size higher

This is due to overlapping partitions, which results in partition sizes higher than it should be. Although partitions are visible in this case, this is a dangerous situation as writing to one parti-

tion could damage another. The remedy is to calculate the partition sizes (as we know the starting sector of the current erroneous partition and the next partition) and editing the values in the partition table.

There can be other partition problems too, but the reason is common – damaged partition entries. So while recovering a partition, the first step would be determining the structure of the partitions and searching the disk area for clues. Once it is determined where exactly the partitions are located, the above mentioned techniques can easily lead to the recovery.

Final note

We saw that disk partitions are merely few set of numbers defined in the MBR and EPP. When the partition table is damaged, all we need to do to get the partitions back is to rewrite these numbers back to their proper places. So it would be wise enough to copy these values to some safe place (maybe paper). There are several software to do this job too. Prevention is better than cure. ■

References

1. http://www.z-a-recovery.com/, *Zero Assumption Recovery*
2. http://www.datarescue.com/life/index.htm, *DataRescue*
3. http://ata-atapi.com/, *Hale's HIW*
4. http://en.wikipedia.org, *Wikipedia*

The BSOD

In a surprise announcement today, Microsoft President Steve Ballmer revealed that the Redmond-based company will allow computer resellers and end-users to customize the appearance of the Blue Screen of Death (BSOD), the screen that displays when the Windows operating system crashes.

The move comes as the result of numerous focus groups and customer surveys done by Microsoft. Thousands of Microsoft customers were asked, "What do you spend the most time doing on your computer?"

"Staring at a Blue Screen of Death." At 54 percent, it was the top answer.

"We immediately recognized this as a great opportunity for ourselves, our channel partners, and especially our customers," explained the excited Ballmer to a room full of reporters.

Immense video displays were used to show images of the new customizable BSOD screen side-by-side with the older static version. Users can select from a collection of "BSOD Themes," allowing them to instead have a Mauve Screen of Death or even a Paisley Screen of Death. Graphics and multimedia content can now be incorporated into the screen, making the BSOD the perfect conduit for delivering product information and entertainment to Windows users.

The BSOD is by far the most recognized feature of the Windows operating system, and as a result, Microsoft has historically insisted on total control over its look and feel. This recent departure from that policy reflects Microsoft's recognition of the Windows desktop itself as the "ultimate information portal." By default, the new BSOD will be configured to show a random selection of Microsoft product information whenever the system crashes. Microsoft channel partners can negotiate with Microsoft for the right to customize the BSOD on systems they ship.

Major computer resellers such as Compaq, Gateway, and Dell are already lining up for premier placement on the new and improved BSOD.

Ballmer concluded by getting a dig in against the Open Source community. "This just goes to show that Microsoft continues to innovate at a much faster pace than open source. I have yet to see any evidence that Linux even has a BSOD, let alone a customizable one."

Save your Desktop

Om Chandra Rimal
2059 Computer

It has been a tradition of saving documents. But, have you ever thought of saving the *Document Savior* itself. The Document Savior here means the sequence of actions done to save files in Microsoft Office tools, Paintbrush, etc. Some of the common Document Saviors end up with CTRL+S and just that. So, saving the sequence of actions to save a document does not come up with any profit. What if you want to know about installing a software which has a series of actions you have never experienced of? Or may be you encounter a really bizzare GUI that keeps you thinking what the hell to do with it. I think here *Desktop Saving* really helps you to learn the procedure much faster than the way you do by the help document. The Desktop Saving is nothing but capturing the screenshots in series and then creating motion picture of them known as a screen capture.

There are various ways of capturing screenshots. One of the most common way is pressing the "PrintScreen" key in the windows operating systems. This helps getting only one picture of the desktop to the clipboard at the moment the key was pressed. But, the Desktop Saving is series of such pictures. We cannot just keep pressing PrintScreen to record the video any way.

"*None of the big operating systems have built in mechanisms to record videos of the screen. A multitude of utilities have come up to fill this void*", says Wikipedia.

An Example of the Desktop Saver

Here, for those who have been working with JMF API, I have brought an idea of making your own `MediaLocator` (Note, MediaLocator is a class provided by JMF API, which locates the source of the media), which can stream the screenshots (of selected rectangular portion of the desktop screen) and present the streamed video as `DataSource` (class in JMF API, which is a formatted source of media) for media referred to by it. You can use this `MediaLocator` as you have done for files and URLs:

```
file:c:\movies\test.mpg,
http://swww.movies.com/test.mpg, etc.
```

The MediaLocator you would use here is

```
screen://l, t, r, b/fps,
```

which is a URL of a kind with 'l', 't', 'r' and 'b' meaning the left, top, right and bottom extremes of the rectangular area to be captured, whereas 'fps' means frame per second to be captured. As for example, if you type

```
screen:// 0, 0, 100, 100/ 10,
```

you get the top left rectangular corner of the desktop with the size 100 by 100 and with 10 frames per second of the video stream. So, the only purpose of the `screen://` protocol (so called) is to provide a media to be processed, pre-fetched, realized, played and even stream through RTP.

The processing, pre-fetching and realizing are some "needed to be done" events for media for proper handling. The playing is some kind of amusement. But, playing what you can see on your desktop means wasting your computing resource. What about streaming your desktop to your friend's ? Is it clicking what you can benefit ? Yes, of course, you can teach your friend how to use mouse around the desktop as you have been doing. The use of Desktop Saving and revealing it in future is another benefit.

The code below gives you the `screen://` protocol, which you need to compile and place the .class file to the `JAVAHOME\jre\lib\ext` folder. There are two files – `LiveStream.java` and `DataSource.java`, so it may need to make a .jar compilation. However, put the .jar file in the folder mentioned above.

It has been tested in the system installed with JMF API 2.1.1e, JAVA 1.5 SE and the Windows operating system (98, 2000 and XP).

The Source Code

```
//   LiveStream.java
package com.sun.media.protocol.screen;
import java.awt.*;
import java.awt.image.BufferedImage;
import javax.media.*;
import javax.media.format.*;
import javax.media.protocol.*;
import java.io.IOException;
import java.util.StringTokenizer;
public class LiveStream implements PushBufferStream, Runnable {
    protected ContentDescriptor cd = new ContentDescriptor(ContentDescriptor.RAW);
    protected int maxDataLength;
    protected int [] data;
    protected Dimension size;
    protected RGBFormat rgbFormat;
    protected boolean started;
    protected Thread thread;
    protected float frameRate = 1f;
    protected BufferTransferHandler transferHandler;
    protected Control [] controls = new Control[0];
    protected int x, y, width, height;
    protected Robot robot = null;
    public LiveStream(MediaLocator locator) {
     try {
         parseLocator(locator);
     } catch (Exception e) {
         System.err.println(e);
     }
     size = new Dimension(width, height);
     try {
         robot = new Robot();
     } catch (AWTException awe) {
         throw new RuntimeException("");
     }
     maxDataLength = size.width * size.height * 3;
     rgbFormat = new RGBFormat(size,maxDataLength,Format.intArray,frameRate,32,

     0xFF0000,0xFF00,0xFF,1,size.width,VideoFormat.FALSE,Format.NOT_SPECIFIED);
     // generate the data
     data = new int[maxDataLength];
     thread = new Thread(this, "Screen Grabber");
    }
    protected void parseLocator(MediaLocator locator) {
     String rem = locator.getRemainder();
     while (rem.startsWith("/") && rem.length() > 1)
         rem = rem.substring(1);           // Strip off starting slashes
     StringTokenizer st = new StringTokenizer(rem, "/");
     if (st.hasMoreTokens()) {
         // Parse the position
         String position = st.nextToken();
         StringTokenizer nums = new StringTokenizer(position, ",");
         String stX = nums.nextToken();
         String stY = nums.nextToken();
         String stW = nums.nextToken();
         String stH = nums.nextToken();
         x = Integer.parseInt(stX);
         y = Integer.parseInt(stY);
         width = Integer.parseInt(stW);
         height = Integer.parseInt(stH);
     }
     if (st.hasMoreTokens()) {   // Parse the frame rate
         String stFPS = st.nextToken();
         frameRate = (Double.valueOf(stFPS)).floatValue();
     } }
```

```
    public ContentDescriptor getContentDescriptor() {
     return cd;
    }
    public long getContentLength() {
     return LENGTH_UNKNOWN;
    }
    public boolean endOfStream() {
     return false;
    }
    int seqNo = 0;
    public Format getFormat() {
     return rgbFormat;
    }
    public void read(Buffer buffer) throws IOException {
     synchronized (this) {
         Object outdata = buffer.getData();
         if (outdata == null || !(outdata.getClass() == Format.intArray) ||
          ((int[])outdata).length < maxDataLength) {
          outdata = new int[maxDataLength];
          buffer.setData(outdata);
         }
         buffer.setFormat( rgbFormat );
         buffer.setTimeStamp( (long) (seqNo * (1000 / frameRate) * 1000000) );
          BufferedImage bi=robot.createScreenCapture(new
  Rectangle(x,y,width,height));
         bi.getRGB(0, 0, width, height, (int[])outdata, 0, width);
         buffer.setSequenceNumber( seqNo );
         buffer.setLength(maxDataLength);
         buffer.setFlags(Buffer.FLAG_KEY_FRAME);
         buffer.setHeader( null );
         seqNo++;
     }
    }
    public void setTransferHandler(BufferTransferHandler transferHandler) {
     synchronized (this) {
         this.transferHandler = transferHandler;
         notifyAll();
     }
    }
    void start(boolean started) {
     synchronized ( this ) {
         this.started = started;
         if (started && !thread.isAlive()) {
          thread = new Thread(this);
          thread.start();
         }
         notifyAll();
     }
    }
    public void run() {
     while (started) {
         synchronized (this) {
          while (transferHandler == null && started) {
              try {
               wait(1000);
              }    catch (InterruptedException ie) { }
          }
     }
     if (started && transferHandler != null) {
          transferHandler.transferData(this);
          try {
              Thread.currentThread().sleep( 10 );
          }    catch (InterruptedException ise) { }
     }
     }
}
```

```
    public Object [] getControls() {
     return controls;
    }
    public Object getControl(String controlType) {
       try {
          Class  cls = Class.forName(controlType);
          Object cs[] = getControls();
          for (int i = 0; i < cs.length; i++) {
             if (cls.isInstance(cs[i]))
                return cs[i];
          }
          return null;
       } catch (Exception e) {   // no such controlType or such control
         return null;
       }
    }
}

//DataSource.java
package com.sun.media.protocol.screen;
import javax.media.Time;
import javax.media.MediaLocator;
import javax.media.protocol.*;
import java.io.IOException;
public class DataSource extends PushBufferDataSource {
    protected Object [] controls = new Object[0];
    protected boolean started = false;
    protected String contentType = "raw";
    protected boolean connected = false;
    protected Time duration = DURATION_UNBOUNDED;
    protected LiveStream [] streams = null;
    protected LiveStream stream = null;
    public DataSource() {
    }
    public String getContentType() {
     if (!connected){
            System.err.println("Error: DataSource not connected");
            return null;
        }
     return contentType;
    }
    public void connect() throws IOException {
      if (connected)
            return;
      connected = true;
    }
    public void disconnect() {
     try {
            if (started)
                stop();
        } catch (IOException e) { }
     connected = false;
    }
    public void start() throws IOException {
        if (!connected)
         throw new java.lang.Error("DataSource must be connected before it can be
   started");
        if (started)
            return;
     started = true;
     stream.start(true);
    }
    public void stop() throws IOException {
     if ((!connected) || (!started))
         return;
```

```
        started = false;
        stream.start(false);
       }
       public Object [] getControls() {
        return controls;
       }
       public Object getControl(String controlType) {
        return null;
       }
       public Time getDuration() {
        return duration;
       }
       public PushBufferStream [] getStreams() {
        if (streams == null) {
            streams = new LiveStream[1];
            stream = streams[0] = new LiveStream(getLocator());
        }
        return streams;
        }
}
```

Scratching Java Security Architecture

Kiran Shakya
2059 Computer

Security is a vast subject which defies even the most promising programmer. For the beginners like us, it is quiet hard to understand even the basics. Lot of books dedicate their expensive pages on theory. Much venerated books like "Inside Java 2 Security Platform – Li Gong" although provide enough material; they all focus on theory, API's and their documentation.

This article explains few simple examples which strive to explain the basic security architecture of Java. I hope the readers are familiar with Java language.

Start by extending the default *SecurityManager* class provided by Java as in the following example.

```
1.  import java.io.*;
2.  import javax.swing.*;
3.  import javax.swing.event.*;
4.
5.  public class  MySecurityManager extends SecurityManager
6.  {
7.       //a password will be asked each time the user opens a file to read
8.       private String password;
9.       public MySecurityManager()
10.      {
11.           super();
12.           this.password="test";
13.      }
14.      public MySecurityManager(String pwd)
15.      {
16.           super();
17.           this.password=pwd;
18.      }
19.
20.      //This method is called whenever a program opens a file
21.      public void checkRead(String filename)
22.      {
23.           String pwd;
24.           System.out.println("Enter secret password for reading file:");
25.           try
26.           {
27.                pwd=new BufferedReader(new
   InputStreamReader(System.in)).readLine();
28.                if(pwd.equals(password))
29.                     System.out.println("Access Granted");
30.                else
31.                     throw new SecurityException("You do not have permission to
   read the file!");
32.           }
33.           catch(IOException e)
34.           {
```

```
35.               throw new SecurityException("Access Denied");
36.          }
37.          //UnComment the line below if you want to control the file reading
  permission on the basis of just a  password and policy file
38.          //super.checkRead(filename);
39.      }
40.      //You can override other methods such as checkWrite() and //
  checkConnect(host,port)
41.  ……..
```

In order to test it we can use following simple program:

```
1.   Import java.io.*;

2.   Public class TestSecurityManager
3.   {
4.        public static void main(String[] args) throws IOException
5.        {
6.             int count=0;
7.             if(args.length!=1)
8.                  System.out.println("Usage: java TestSecurityManager
  <filename>");
9.             Else
10.            {
11.                 try
12.                 {
13.                      //Our password is 'java'
14.                      System.setSecurityManager(new MySecurityManager("java"));
15.                 }
16.                 catch(SecurityException e)
17.                 {
18.                      System.err.println("Security Exception");
19.                 }
20.                 try
21.                 {
22.                      FileInputStream fis=new FileInputStream(args[0]);
23.                      While(fis.read()!=-1)
24.                           count++;
25.                      fis.close();
26.                      System.out.println("Total Line numbers: "+count);
27.                 }
28.                 catch(Exception e)
29.                 {
30.                      e.printStackTrace();
31.                 }
32.            }}}
```

In order to give our security manager full right to open and write file, its best to compile it and move the compiled class to `classes` directory in `${java.home}` (usually `c:\jdk\jre` for v1.3 or `c:\program files\java\jre` for v1.5). This directory does not exist by default. But once we create it becomes part of the boot class path, so it is completely trusted. Run the program it will ask you for password. Enter 'java' and it will count the number of line in the file. So what we have done? Whenever an application has a security manager installed, it is not allowed to do all sort of activities. For each activity, it needs to consult with the security manager which decides whether the application has enough credentials or rights for that activity or not. If we want more control over the execution then we add specific permission for the application in `java.security` file found in `${java.home}/lib/security` directory.

```
Grant codebase "file:/D:/security/"
{
permission java.lang.RuntimePermission "createSecuritymanager";
permission java.lang.RuntimePermission "setSecurityManager";
permission java.io.FilePermission "C://Documents and Settings//kiran//
   *","read,write";
}
```

The policy file specifies that the program is allowed to read and write all files in `c:\Documents and Settings\kiran` directory. Here codebase refers to the directory where out application resides. First uncomment the line number 38 in security manager, compile it and move it to classes directory. Then we can run the above example as:

```
java –
   Djava.security.policy=MyPolicy.policy
   TestSecurityManager <filename>
```

The output depends on the location of file you give on command line and that we gave in policy file. In order to successfully open the file, both password and policy should be correctly matched. Try changing both policy file and <filename>, it will make your concept clear. This technique of running an application in secure environment is called **Sandboxing.** An Applet always runs in a sandbox, that's why we can't read or write files from the client.

We can also record all the permission granted or denied by security manager in a log file. For this just use ordinary java i/o function to record the permissions in a text file form *MySecuritymanager* class.

```
Class MySecuritymanager extends SecurityManager
{
     //Same as above
          …….
     private DataOutputStream log;
     Constructor()
     {
          ……..
          log=new DataOutputStream(new FileOutputStream("Log.log"));
          log.writeBytes("Write Request log started.\n")l
          ……..
          ……..
     }
     public void checkRead(String file)
     {
          ……
          log.write("File read Access granted or denied");
          ……..
     }
}
```

Run the above program again and check the log file.

So now we have basic understanding of Java security model, lets us turn our attention to SSL (Secure Socket Layer)

SSL

SSL is a standard protocol for implementing cryptography and enabling secure transmission on the web. To access SSL web sites, we use https in place of http in URL location. The primary goal of SSL is to provide privacy and reliability between two communicating parties.

SSL has 2 security aims:

1. To authenticate the server and the client using public key signatures and digital certificates.
2. To provide an encrypted connection for the client and server to exchange messages securely.

The steps involved in a simple SSL transaction before the communication of data begins are described in the following list:

1. The client sends the server *Client Hello* message containing a request for connection along with client capabilities and version, the cipher suites and the data compression methods it supports.
2. The server responds with *Server Hello* message which includes cipher suite, compression method, session ID for connection.
3. The server sends its certificate chain if it is to be authenticated and client verifies it.
4. The client sends *ClientKeyExchange* message. This is the random key encrypted by server's public key which the client get form the server's certificate.
5. When client and server agree on common symmetric key for encrypting the communication, the client sends a *ChangeCipherSpec* message indicating the confirmation that it is ready. This message is followed by *Finish* message.
6. The server then sends its own C*hangeCipherSpec* message indicating its readiness. This message is followed by *Finish* message.

Now the communication starts in secure mode.

Note that a cipher suite determines:

- The kind of key exchange algorithm used
- The encryption algorithm
- The digest algorithm
- Whether the cipher strength is freely exportable.

An example of Cipher Suite: SSL_RSA_EXPORT_WITH_RC4_40_MD5 implies:

1. RSA (512 bit) key exchange mechanism
2. RC4-40 bit encryption algorithm
3. MD5 hash function
4. Exportable

SSL in Java

We use `javax.net` and `javax.net.ssl` package for SSL programming as demonstrated in following simple code:

```
// create a class and insert the following code where we need SSL
SSLServerSocket s;
System.out.println("Server starting...");
SSLServerSocketFactory SSLsvrfact=(SSLServerSocketFactory)
SSLServerSocketFactory.getDefault();
s=(SSLServerSocket)SSLsvrfact.createServerSocket(8289);
s.setNeedClientAuth(false);
System.out.println("Server started on port 8289");
System.out.println("Waiting for the client...");
String[] cipher_suites=s.getEnabledCipherSuites();
System.out.println("Enabled cipher Suites are:");
for(int i=0;i<cipher_suites.length;++i)
     System.out.println(" "+cipher_suites[i]);

String[] encs=new String[cipher_suites.length+1];
for(int len=0;len<cipher_suites.length;++len)
     encs[len]=cipher_suites[len];

s.setEnabledCipherSuites(encs);
SSLSocket c = (SSLSocket)s.accept();
//Rest are typically same with ordinary socket programming
```

```
//Similarly in client side:
System.out.println("Requesting connection from " + arg + " on port 9335...");
SSLSocketFactory SSLfact=(SSLSocketFactory)
     SSLSocketFactory.getDefault();
SSLSocket s=(SSLSocket)SSLfact.createSocket(arg,8289);
String[] cipher_suites=s.getEnabledCipherSuites();
System.out.println("Enabled cipher Suites are:");
for(int i=0;i<cipher_suites.length;++i)
     System.out.println(" "+cipher_suites[i]);
String[] encs=new String[cipher_suites.length+1];
for(int len=0;len<cipher_suites.length;++len)
     encs[len]=cipher_suites[len];
s.setEnabledCipherSuites(encs);
String[] ncipher_suites=s.getEnabledCipherSuites();
System.out.println("Enabled Cipher Suites are:");
for(int i=0;i<ncipher_suites.length;++i)
     System.out.println(" "+ncipher_suites[i]);
System.out.println("Connected to Server");
System.out.println("The one negotiated is "+s.getSession().getCipherSuite());
InputStream in = s.getInputStream();
DataOutputStream out = new DataOutputStream(s.getOutputStream());
// Rest are more or less same with socket programming
```

If we use above code for creating our socket, then packet analyzing software won't be able to read the packet contents as the data will be in encrypted form. Consider this case with ordinary socket programming where the data are sent in packet without any encryption.

(There may be typing mistake in the above programs, so I can't guarantee the validity of above programs)

A Final Note

Security is difficult subject to write about. It touches all aspects of computing technology, including hardware, software, operating system, software libraries, communication and network infrastructure and so on. Be careful when you think about securing your application. It's very difficult and challenging.

Nevertheless, Enjoy programming ■

> *"There is no reason anyone would want a computer in their home."*
>
> *--Ken Olson, founder of DEC, 1977"*

The X in ML

Roshan Newa
2059 Computer

"There is a lot of hype about the XML. Despite critics backing off against XML for the more popular HTML, XML is to exist and exist for good with HTML or anything other ML at least for present moment."

Web, web and still more web is all that exists in the World Wide Web today. Whether you love it or hate it, the Internet and the WWW are here to stay. This is the destiny of the small vision of Tim Berners-Lee laid down in his original proposal for the CERN (Centre for European Nuclear Research) in 1989 where he says:

"...a *universal linked information system*, in which *generality* and *portability* are more important than fancy graphics and complex extra facilities."

No wonder HTML has been, and still is, a fantastic success. It's no exaggeration that had there been no HTML there would not have been the web. As the use of HTML grows more and more its limitations are exploited more and more by the web as its *weaknesses*!

Well…

1. HTML lacks syntactical checking

Means you cannot validate HTML code. The web browsers have been designed to accept almost anything like HTML documents. If you don't care that ends with then that is another matter. Strangely enough, the only markup that is really compulsory in HTML is the <title>....</title> tag. I guess this is no more true anymore.

2. HTML lacks structure

Well this isn't HTML's fault at all. The problem lies in the way HTML code is used.

3. HTML is not content aware

Yeah this is why searching the web is complicated by the fact that HTML doesn't give the way to describe the information content and the semantics of the document.

4. HTML is not international

Surprisingly nobody is claiming it and yet there were a few proposals to internationalize it and give it a way of identifying the language used inside the tags.

5. HTML is not suitable for data exchange

Why, HTML tags do little to identify the information that a document contains. They only define how the data should appear on the browser.

6. HTML is not object oriented

Sure modern programmers want to transit to all but the object oriented techniques, and there is little for them in HTML.

7. HTML lacks a robust linking mechanism

Now you know why there is little you could do when you encounter a broken link: because the links are hard coded in the pages and no webmaster is excited to update thousands of pages just to correct a broken link.

8. HTML is not reusable

HTML code can be extremely difficult to reuse because they are so specifically tailored to their place in the web of associated pages.

9. HTML is not extensible

Well…., HTML was never actually meant to be extensible at the first place. Maybe we are demanding too much from it.

All it means that we are now geared up to head start with XML. So what is exactly XML? In his book "Learning XML", Eric Ray defines XML as:

"XML (Extensible Markup Language) is a flexible way to create ***self-describing data*** *and to* ***share both the format and the data*** *on the World Wide Web, intranets, and elsewhere".*

XML is a text-based markup language. XML is a data storage toolkit, a configurable vehicle for any kind of information, an evolving and open stand-

ard embraced by everyone from bankers to webmasters. In just a few years, it has captured the imagination of technology pundits and industry mavens alike. A short list of XML's features says it all.

1. Data Orientation

XML can store and organize just about any kind of information in a form that is tailored to your needs.

2. Internationalization

As an open standard, XML is not tied to the fortunes of any single company, nor married to any particular software. With Unicode as its standard character set, XML supports a staggering number of writing systems (scripts) and symbols, from Scandinavian runic characters to Chinese Han ideographs. On contrary HTML and SGML are based on ASCII.

3. Modularity

While HTML had implied DTD (Document Type Definition), XML enjoys the freedom of leaving DTD altogether or using sophisticated resolution methods, combine multiple fragments of XML instances or DTDs into one single compound instance.

4. Extensibility

XML allows you to link to material without requiring the link target to be physically present in the object. This opens up exciting possibilities for linking together things like material to which you do not have write access, CD-ROMs, library catalogs, the results of database queries, or even non-document media such as sound fragments or parts of videos. Furthermore it allows you to store the link separately, so that it is easy to maintain the link.

5. Document Validation

XML offers many ways to check the quality of a document, with rules for syntax, internal link checking, comparison to document models, and datatyping.

6. Simplicity

With its clear, simple syntax and unambiguous structure, XML is easy to read and parse by humans and programs alike.

7. Distribution

XML is easily combined with stylesheets to create formatted documents in any style you want. The purity of the information structure does not get in the way of format conversions.

XML is itself not a language, but a specification of creating markup languages. A simple XML document contains following parts:

1. XML prolog

An XML file always starts with the XML prolog. The minimal prolog contains a declaration that identifies document as an XML document:

```
<?xml version="1.0"?>
```

The declaration may also contain additional information:

```
<?xml version="1.0" encoding="ISO-
   8859-1" standalone="yes"?>
```

version: Identifies the version of the XML markup language used in the data. This attribute is not optional.

encoding: Identifies the character set used to encode the data. ISO-8859-1 is Latin-1, the Western European and English language character set. (The default is 8-bit Unicode: UTF-8.)

standalone: Tells whether or not this document references an external entity or an external data type specification. If there are no external references, then "yes" is appropriate.

2. Tags and attributes

In XML you identify data using tags (identifiers enclosed in angle brackets: <…>). Collectively these tags are known as markups. But unlike HTML, XML tags identify the data rather than how to display the data. An XML tag acts like a field name with the field value enclosed between the start and end of tag. A simple tag can be:

```
<Message> Hello </Message>
```

Here the tag name is 'Message' and the value of the tag is 'Hello'. Like in HTML, the end of tag is specified by '/', but whereas in HTML they are optional, in XML it results in parsing error. The XML file can contain any XML tags that make sense for the given application. The tags can contain other tags.

Tags can also contain attributes – additional information included as part of the tag itself, within the tag's angle brackets. For example:

```
<Message
   from=me@myAddress.com>Hello</
   Message>
```

The attribute name is followed by an equal sign and the attribute value. Multiple attributes are separated by spaces.

3. Empty Tags

XML document is always constrained to be well formed. In XML, every tag has a closing tag. So, in XML, the </to> tag is not optional. The <to> element is ever terminated by any tag other than </to>. An empty tag is a special kind of tag which has no content. They can be written as:

```
<empty tag></empty tag>
```

Or

```
<empty tag/>
```

XML is a flexible way to create self-describing data and to share both the format and the data

4. Processing Instructions

An XML file can also contain processing instructions that give commands or instructions to the application that is processing the XML data. Processing instructions has following format:

```
<?target instructions?>
```

Here, target is the name of the application that is expected to do the processing and instructions is the string of characters that embodies the information or commands for the application to process. A complete data to send a mail can be written in XML form as:

```
<?xml encoding="utf-8" version="1.0"
   ?>
<message to="zerone@ioe.edu.np"
   from="me@myAddress.com"
subject="XML Is Really Cool">
<!– This is a comment –>
<text>
```

How many ways is XML cool? Let me count the ways...

```
</text>
</message>
```

Document Models

This is as simple as it gets with the XML. The real power behind XML lies in the fact that XML can be used to define custom tags and markups. A document model is used to define your own elements and attributes. These models are stored in separate documents themselves and are referenced in the main XML document. Document models may be hassle to maintain for just one or two documents, but if your needs are for more documents and your quality control needs are very high, then they are the best choice you could have. There are two ways to create document models:

i) Document Type Definition
ii) XML schema

Out of these two models, it is upto the application user to decide which one to use. It has always been the complain that DTDs are old fashioned and aren't expressive. Others find it strange that documents follow one syntax and DTDs another. The content models and attribute list declarations are difficult to read and understand, and it's frustrating that patterns for data in elements and attributes can't be specified. Unlike DTD syntax, XML Schema syntax is well-formed XML, making it possible to use your favorite XML tools to edit it. It also provides much more control over datatypes and patterns, making it a more attractive language for enforcing strict data entry requirements. Nonetheless DTDs still have their strengths: compact size, familiar syntax, simplicity. Together, the two provide alternate methods to achieve similar goals.

Application Development with XML

An XML application is typically built around an XML parser. It has an interface to its users, and an interface to some sort of back-end data store.

An XML parser is a program that is used to read and analyze an XML document. We categorize parsers in several ways:

- Validating versus non-validating parsers
- Parsers that support the Document Object Model (DOM)
- Parsers that support the Simple API for XML (SAX)
- Parsers written in a particular language (Java, C++, Perl, etc.)

Each one of them have their own reasons for using or not using. To use parser we:

1. Create a parser object
2. Pass the XML document to the parser
3. Process the results

Due to the massive use of XML in application development, most of today's Integrated Development Environment such as Microsoft's Visual Studio, Netbeans Netbeans IDE and Sun's Studio have in-built support for XML.

Where does XML stands in today's context?

XML is revolutionizing the way applications are written and developed. The ability to exchange information with unseeming ease irrespective or platform, language or medium has made XML the number one choice of application builders. More and more number of applications are being developed using XML. Because of its self-documenting property, many applications (particulary graphics programs) use XML documents to store the file data and a temporary replacement of database. And still larger number of protocols are being developed based on XML. Only a shortlist of them are mentioned here:

a. Web services

Web Services are objects and methods that can be invoked from any client over HTTP. Web Services are built on the Simple Object Access Protocol (SOAP) that use XML for exchanging data through HTTP. Web services are the core of today's web applications.

b. Math Markup Language

Even now, if you want to display basic mathematical equations in a Web page,

you can choose between making a screen capture and importing it into your document as a graphics file, or wasting hours on the extremely frustrating exercise of trying to find the right symbol in the right font. MathML intends to bridge between Mathematics and HTML and provide support for both presentation markup and semantic markup for mathematics.

c. Asynchronous JavaScript And XML (AJAX)

Ajax is an approach to Web application development that uses client-side scripting to exchange data with the Web server. As a result, Web pages are dynamically updated without a full page refresh interrupting the interaction flow.

d. Vector Markup Language

VML is XML based implementation of Structured graphics. VML takes the basics of structured graphics but puts them inside XML elements that allows us to describe ASCII vector format for describing drawings on web browsers.

e. Cascading Action Sheets

It is an Netscape's proposal to attaching behaviours to XML elements.

f. HandHeld Device Markup Language (HDML)

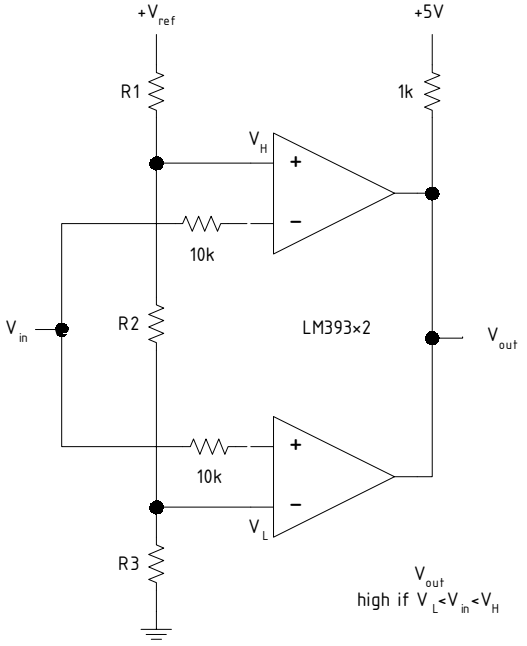
This is a sort of mini-HTML that would allow hand-held devices (PDAs like the PalmPilot, mobile telephones, and palm computers) to browse the Web and communicate over the Internet.

Many believe that XML is the future of markup languages. The very power that it presents at its disposal has revolutionized the way web programs are written. So much so that ASP, JSP and DHTML have all provided support for XML tagging. And as HTML is being absorbed by XML, the web is finally getting simpler at last, I guess. ■

References

1. *The XML Handbook*, Prentice Hall India Publications
2. *Learning XML*, O'Reilly Publications
3. *Beginners XML*, Wrox Publications
4. www.xml.org
5. www.w3c.org

circuit ideas



Window Discriminator

(Source: AoE)

One great shot

Santosh Pradhan
2059 Electronics

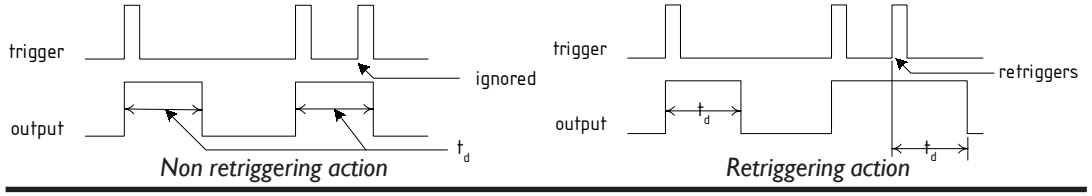
One-shot is the Monostable multivibrator, with which most of us are familiar with. To begin with lets go a little off track and define what a multivibrator is. A multivibrator is a circuit whose output may assume one of two discrete values or voltage levels; just a fancy name for two state digital circuit. The multivibrator family comprises of a few classes of circuits. Monostable multivibrator is one of them others are the bistable and the astable multivibrator. Bistables = the flip-flop are the ones which are stable in both states - high or low, meaning the output may remain in any one of the states for an indefinite period until triggered to switch states externally. Astables = square wave generators are the ones with no stable states, meaning the output will change from high state to low state and vice versa after a fixed interval of time without any external intervention. Monostables are in between these two, have one stable state.

An example of bistable device would be a common household switch used to turn on/off lights, it has two stable states light ON and OFF. A monostable device would be a door bell switch, it has only one stable state doorbell OFF. Extending this analogy further, an astable device can be either of the above switches with a 2 year old playing with it, periodically turning it on and off, here we see that the oscillations sustain indefinitely or at least till the switch breaks.

Getting back to the point, one-shots tend to remain in its stable state unless they are forcibly triggered to switch to its unstable state. One-shots are triggered externally either by applying a high/low level or by a rising/falling edge at the triggering input. As a response the output switches to the unstable state but returns to the stable state after a fixed interval of time. The result is a pulse of predefined width and polarity.

Retriggerability

Retriggerability of one-shot determines whether or not the circuit will acknowledge the occurrence of trigger at the input *while* it is executing its unstable cycle. A retriggerable one-shot will monitor its input during its unstable state and restart the cycle (in effect lengthening the pulse) if trigger is encountered. As for the non-retriggerable one-shot, it will simply ignore any trigger events while its in the unstable state.

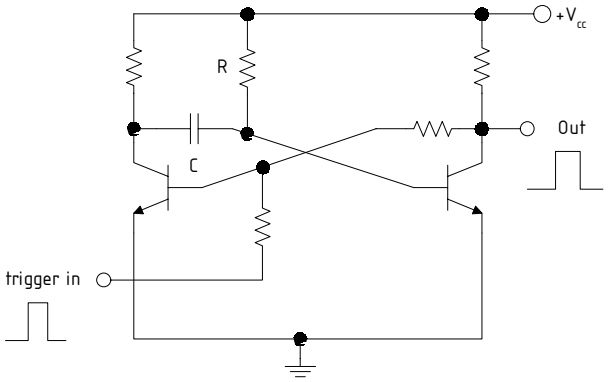


One-shots triggered at rising edge of trigger input

Build one

Using discrete components

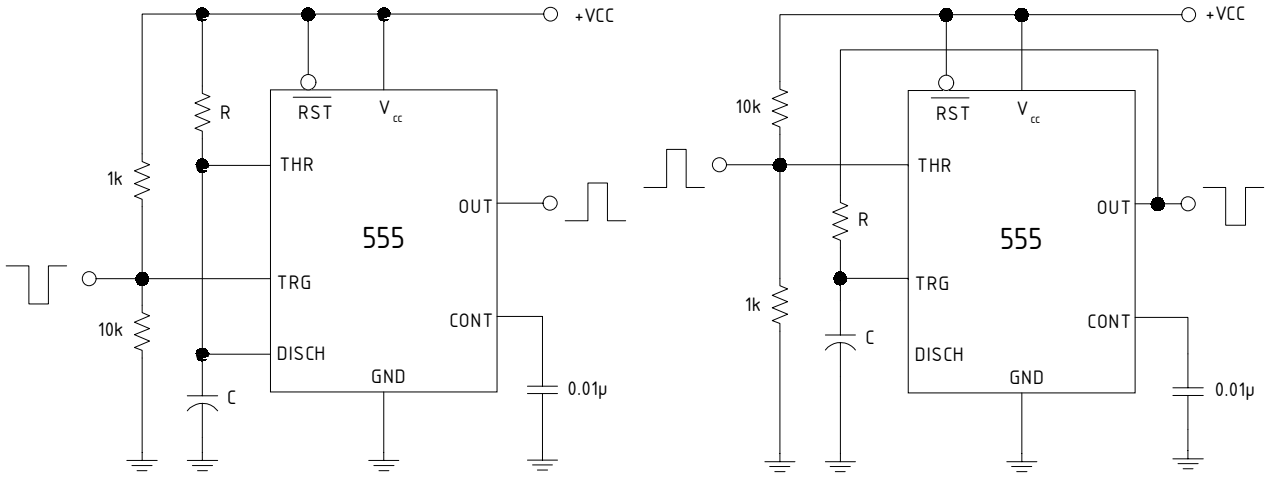
A very simple and crude one-shot. The pulse width is proportional to the product RC. Exact values of other resistors does not matter as long as they are chosen to allow suitable current through them. This is circuit is not desirable as one-shot as it has rather long rising and falling edges in the pulse it produces.



a level triggered one-shot, pulse width roughly equal to time it takes to charge C to 1V through R at given V_{cc}.

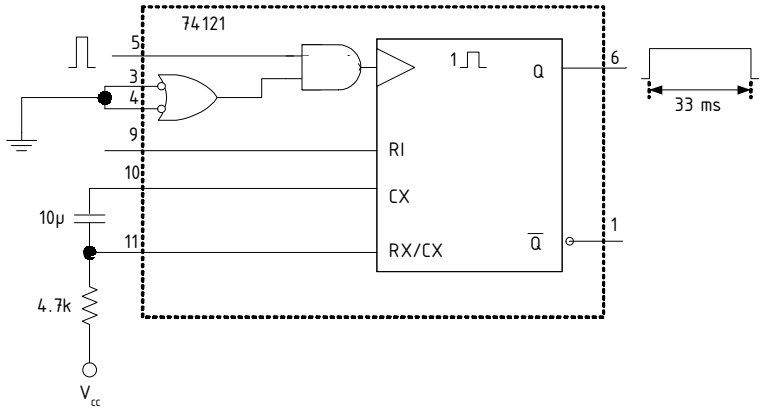
Using 555 timer

A classic one-shot circuit. There are two variations presented here, Form I takes an inverted pulse as input and gives a positive output pulse and Form II acts in opposite manner. The voltage divider at the input is there (thanks to Mali sir) to clamp it to well above 1/3 V_{cc} in Form I and well below 2/3 V_{cc} in Form II, otherwise they are prone to false triggering by just about anything. These are level triggered, non-retriggerable one-shots. As these are level triggered devices, the input pulse should be removed well before (at least 10μs before, according to National LM555 datasheet) the output pulse is over.



Form I *Form II*

Pulse width roughly given by t = 1.1RC in both forms



using 74xx121 as a non-retriggerable one-shot with pulse width of 33 ms. The pulse width for this particular chip is given roughly by t=0.7RC. '121 has 3 trigger inputs at pins 3, 4 and 5. only pin 5 is used here, one-shot is triggered at the rising edge. if pin 3(or 4) were used it would be triggered at the falling edge. See datasheet for details

Using monostable ICs

74xx121 is a non retriggerable one-shot, 74xx122 is a retriggerable with reset, 74xx123 is almost a dual '122 and 4538 dual CMOS retriggerable one-shot to name a few. All of them use external RC combinations to set the pulse width. Some have internal components that determine the pulse width of none are connected externally. All of them have multiple edge sensitive trigger inputs gated together (much like gated enable pins in '138 decoder) so that more than one (and more than one type of) input can trigger it. Some have reset pin that allow inhibit triggering or cause premature termination of pulse. These however are quite expensive, one such IC costs around 70-90 rupees.

Use one

Contact debouncer

Mechanical switches are often used to pull up/down a point in circuits. When these switches are toggled they tend to oscillate/bounce several times before they settle down. This is a contact bounce and produces erratic voltages at that point, which if connected directly to some logic can produce unexpected results. To eliminate this use a non-retriggerable one-shot to interface the switch with other circuits.

Timed switch

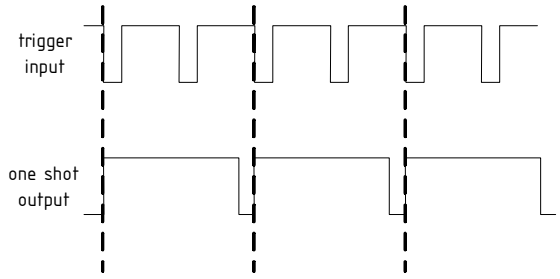
Consider a scenario where you need to do something for a certain interval after some event has occurred — say turn on a LED for 2 sec. after you press a push button — Drive the LED through one-shot triggered by the button, set 2 sec pulse width.

Pulse generator/regenerator

It is quite obvious from the function of one-shots that they can be used to generate fixed width rectangular pulses. Also it can be used to regenerate clean pulses out of noisy or distorted ones.

Frequency divider

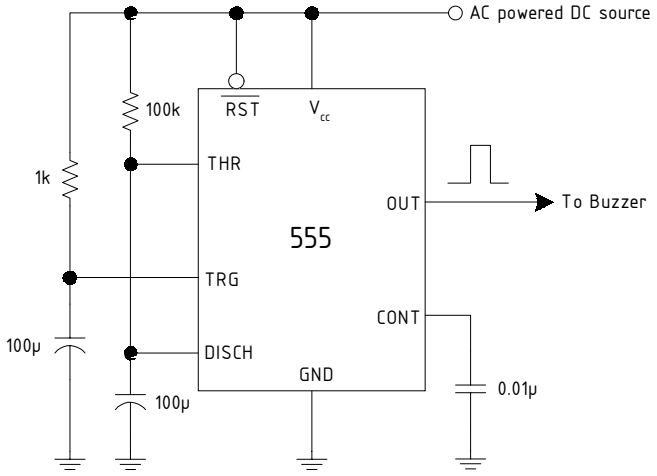
One-shots can be used as a clock frequency divider, it can act a divide by n circuit for any integer n. The trick is to use a non-retriggerable one-shot and set the pulse width slightly higher than (n-½)×clock periods. The duty cycle of resulting pulse train will not be 50%, however, required frequency is obtained.



to build a divide by 2 circuit set pulse width to around I ½ input periods. using Form I, 555 monostable

The 'Load shedding is over' alarm

This slightly modified 555 one-shot is triggered at power on because the capacitor at input, initially uncharged, pulls the trigger pin down to ground. This activates the buzzer at its output. Just to sound a buzzer is perhaps quite stupid thing to do here as similar action could have been achieved simply by putting the DC source across the buzzer. A better use might be to turn on a computer at power on, to do this use to the output pulse to drive a transistor and momentarily short computer's power switch. This only works for ATX based PCs and remember to lower the pulse width to 1 sec or less as most ATX supplies switch off again when the power button is pressed for more than 4 sec. One more thing, this is not a very reliable circuit as it is crucial that the cap at the trigger be uncharged at power on, but in this context it is bound to be so as the circuit would be powered down for hours.



one-shot that gets triggered at power on

Concluding remarks

This short list of simple uses of one-shots is by no means complete. One can think of many more sophisticated uses. But one-shots isn't the answer to all timing problems. When precision is required these simple monostables may not suffice. Also use of one-shots often leads to asynchronism. So use them but wisely. ■

References

1. Floyd, Thomas L., *Digital Fundamentals*
2. Horowitz & Hill, *The Art of Electronics*
3. www.allaboutcircuits.com
4. www.discovercircuits.com
5. National Semiconductor, *LM555/LM555C data-sheet*, May 1997