Lesson 11 Multimedia Communication/Security issues and Quality of Services

By the end of the lesson the learner should be able to:

i) Explain the components of Multimedia
ii) Describe the applications of Multimedia
iii) Explain common security attributes that define security goals
iv) Describe the main Quality of Service parameters that define network perfomance

Multimedia is an interactive media that provides multiple ways to represent information to the user in a powerful manner. It provides an interaction between users and digital information. It is a medium of communication. Some of the sectors where multimedia is used extensively are education, training, reference material, business presentations, advertising and documentaries.

Definition of Multimedia

By definition Multimedia is a representation of information in an attractive and interactive manner with the use of a combination of text, audio, video, graphics and animation. In other words, Multimedia is a computerized method of presenting information combining textual data, audio, visuals (video), graphics and animations. For examples: E-Mail, Yahoo Messenger, Video Conferencing, and Multimedia Message Service (MMS).

Multimedia as name suggests is the combination of Multi and Media that is many types of media (hardware/software) used for communication of information.

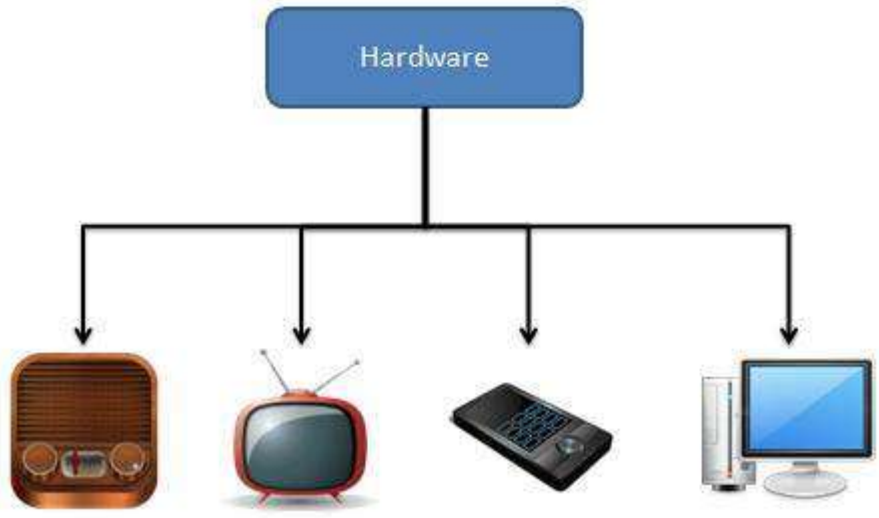

Fig 11.1 Multimedia hardware

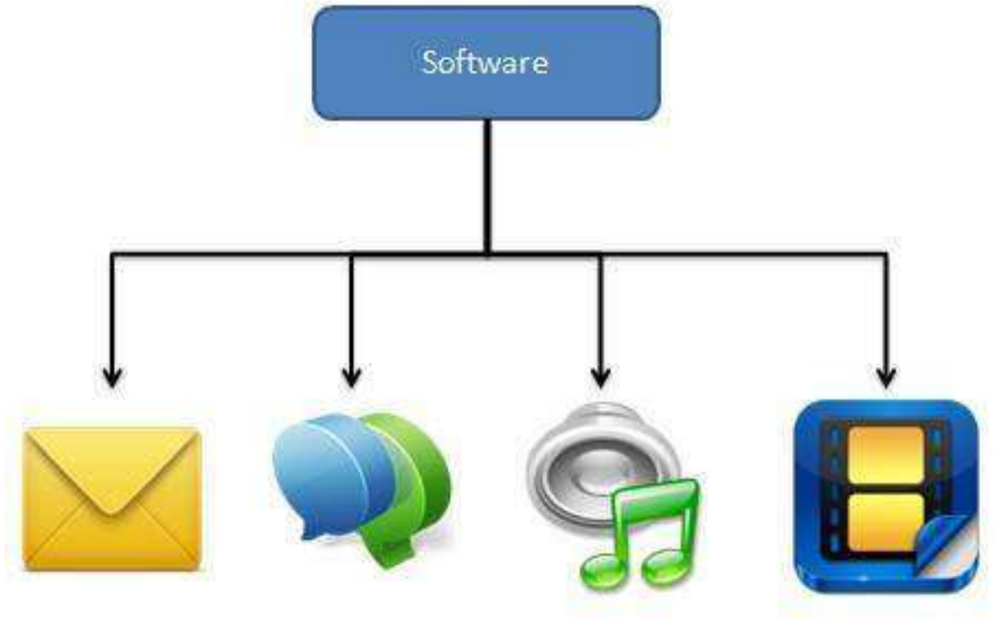

Fig 11.2 Multimedia Software

Components of Multimedia

The following are the common components of multimedia:

- **Text**- All multimedia productions contain some amount of text. The text can have various types of fonts and sizes to suit the profession presentation of the multimedia software.
- **Graphics**- Graphics make the multimedia application attractive. In many cases people do not like reading large amount of textual matter on the screen. Therefore, graphics are used more often than text to explain a concept, present background information etc. There are two types of Graphics:
    - **Bitmap images**- Bitmap images are real images that can be captured from devices such as digital cameras or scanners. Generally bitmap images are not editable. Bitmap images require a large amount of memory.
    - **Vector Graphics**- Vector graphics are drawn on the computer and only require a small amount of memory. These graphics are editable.
- **Audio**- A multimedia application may require the use of speech, music and sound effects. These are called audio or sound element of multimedia. Speech is also a perfect way for teaching. Audio are of analog and digital types. Analog audio or sound refers to the original sound signal. Computer stores the sound in digital form. Therefore, the sound used in multimedia application is digital audio.

- **Video**- The term video refers to the moving picture, accompanied by sound such as a picture in television. Video element of multimedia application gives a lot of information in small duration of time. Digital video is useful in multimedia application for showing real life objects.

  Video have highest performance demand on the computer memory and on the bandwidth if placed on the Internet. Digital video files can be stored like any other files in the computer and the quality of the video can still be maintained. The digital video files can be transferred within a computer network. The digital video clips can be edited easily.

- **Animation**- Animation is a process of making a static image look like it is moving. An animation is just a continuous series of still images that are displayed in a sequence.

- **T**he animation can be used effectively for attracting attention. Animation also makes a presentation light and attractive. Animation is very popular in multimedia application

Applications of Multimedia

The following are the common areas of applications of multimedia.

- **Multimedia in Business**- Multimedia can be used in many applications in a business. The multimedia technology along with communication technology has opened the door for information of global wok groups. Today the team members may be working anywhere and can work for various companies. Thus the work place will become global. The multimedia network should support the following facilities:
  - Voice Mail
  - Electronic Mail
  - Multimedia based FAX
  - Office Needs
  - Employee Training
  - Sales and Other types of Group Presentation/Records Management

- **Multimedia in Marketing and Advertising**- By using multimedia marketing of new products can be greatly enhanced. Multimedia boost communication on an affordable cost, opened the way for the marketing and advertising personnel. Presentation that have flying banners, video transitions, animations, and sound effects are some of the elements used in composing a multimedia based advertisement to appeal to the consumer in a way never used before and promote the sale of the products.

- **Multimedia in Education**- Many computer games with focus on education are now available. Consider an example of an educational game which plays various rhymes for kids. The child can paint the pictures, increase reduce size of various objects etc apart from just playing the rhymes.

  Several other multimedia packages are available in the market which provide a lot of detailed information and playing capabilities to kids.

- **Multimedia in Bank**- Bank is another public place where multimedia is finding more and more application in recent times.
- People go to bank to open saving/current accounts, deposit funds, withdraw money, know various financial schemes of the bank, obtain loans etc. Every bank has a lot of information which it wants to impart to in customers.
- For this purpose, it can use multimedia in many ways. Bank also displays information about its various schemes on a PC monitor placed in the rest area for customers. Today on-line and Internet banking have become very popular. Banks use multimedia extensively. Multimedia is thus helping banks to give service to their customers and also in educating them about banks attractive finance schemes.
- **Multimedia in Hospital**- Multimedia best use in hospitals is for real time monitoring of conditions of patients in critical illness or accident. The conditions are displayed continuously on a computer screen and can alert the doctor/nurse on duty if any changes are observed on the screen. Multimedia makes it possible to consult a surgeon or an expert who can watch an ongoing surgery line on his PC monitor and give online advice at any crucial juncture.

  In hospitals multimedia can also be used to diagnose an illness with CD-ROMs/ Cassettes/ DVDs full of multimedia based information about various diseases and their treatment. Some hospitals extensively use multimedia presentations in training their junior staff of doctors and nurses. Multimedia displays are now extensively used during critical surgeries.

- **Multimedia Pedagogues**- Pedagogues are useful teaching aids only if they stimulate and motivate the students. The audio-visual support to a pedagogue can actually help in doing so. A multimedia tutor can provide multiple numbers of challenges to the student to stimulate his interest in a topic. The instruction provided by pedagogue have moved beyond providing only button level control to intelligent simulations, dynamic creation of links, composition and collaboration and system testing of the user interactions.
- **Communication Technology and Multimedia Services**- The advancement of high computing abilities, communication ways and relevant standards has started the beginning of an era where you will be provided with multimedia facilities at home.

These services may include:

- Basic Television Services
- Interactive entertainment
- Digital Audio
- Video on demand
- Home shopping
- Financial Transactions
- Interactive multiplayer
- Single player games
- Digital multimedia libraries
- E-Newspapers
- e-magazines

COMPRESSION

SECURITY ISSUES IN DATA COMMUNICATION.

Security and Network Security Goals

Networked systems (simple apps, complex networks, complete IT infrastructures)

- Operate in environments involving different interconnected parties each with their own agenda (goals), which may not match with the goals of other parties of the system as whole. As such, it is essential to also consider the security requirements of systems (i.e. what should not go wrong), not only their functional requirements (i.e. what the systems should achieve).
- Is your system secure?" What does this question actually mean; does it means that nobody but you can use it; can throw it out the window; can keep you away from using it...? Denial of service.
- Security requirements are expressed in terms of security attributes that express goals that one may want to achieve to call a system `secure'. The most commonly used and widely accepted security attributes are:
    - Confidentiality, i.e. `my information stays secret',
    - Integrity, i.e. `my information stays correct', and
    - Availability, i.e. `I can get at my information' (sometimes called the C-I-A triad).

Of course these concepts can also refer to resources or system aspects other than just `information'.

- In addition to Confidentiality, Integrity and Availability (`C-I-A') other security attributes are sometimes formulated. Closely related but not usually called a security attribute is Privacy, i.e. `information about me is not misused'.
- Note the difference between Confidentiality and Privacy: where confidentiality requires data that you possess to remain secret, privacy deals with data about you that may be in the hands of others. While `who gets the data' is a key question in confidentiality, the purpose for which data is used is a key ingredient for privacy.
- Other examples of security attributes are:

  Authenticity, i.e. `is this information authentic (i.e. of undisputed origin)',
  Non-repudiation, i.e. `is this information undeniable' and
  Accountability, i.e. `is the information provider accountable (can we punish the provider if the information is incorrect)'.
  Authenticity is different from integrity in that it focuses on data coming from the `correct' source rather than on data not being changed along the way.

A signature on a contract would be an example of a way to achieve non-repudiation; you cannot later deny agreeing to the conditions in the contract.
The relation between accountability and non-repudiation is similar to that between integrity and authenticity; nonrepudiation can be an important part of achieving accountability but is by itself not sufficient.

- The security requirements together with the security policies of a system tell you what attributes should be achieved when (in which context). The requirements will typically say what security attributes should be achieved by which components and/or for what type of resources (e.g. confidential database entries should only be readable by user with the right clearance).
- Security policies detail with this e.g. by stating what type of data is confidential and what (types of) users have clearance.
- Security requirements are an integral part of the design of the system while changes of policies is typically taken into account and should not invalidate the design. Note, however, that the term security policy is widely used and the exact interpretation varies. It could be a high level textual description meant to be understood and applied by human beings, e.g. all personal identifiable information must only be read when needed to provide a service" to low level computer readable information e.g. \drwxr-xr-x"5.
- Translating high level policies into a systems design along with low level policies is an important step of creating a secure system.
- The exact meaning of a security policy can be given within a security model; a (formal) framework to express and interpret policies. For example, the Unix _le permission given above can be interpreted as a relation between Users, Groups, Objects and Permissions: An object (e.g. a directory) has an owner user and a group (an additional part of the security policy) and the owner of the object has

read, write and execute permission, while members of the group as well as other users have only read and execute permission.

Threats

- The security attributes of the system may be at risk from several types of threats. Besides the usual problem such as program errors and system failures, security also needs to address malicious entities, which are specifically trying to break the system.
- This is very challenging; every day seems to bring new security incidence where attackers are able to exploit (previously unknown) security weaknesses.
- Although this may give a skewed perspective (a system remaining secure yet another day will not make the news), it does show the importance of applying the right mechanisms for securing your system.
- To decide what the right mechanisms are to achieve the security requirements of the system, we need to know whom we want to protect against.
- Protecting information in a database from an outsider requires different solutions than protecting it from the database administrator. We thus need an attacker model.
- This attacker model captures the capabilities and possibly the intentions of an attacker. For example, in a network setting we may distinguish between attackers that can only listen in (eavesdrop) and those that can block and/or modify communication.
- Attacker models can be general, e.g. IBM's classification of attackers into three categories:
    - Clever outsiders,
    - Knowledgeable insiders and
    - Funded organizations; or formal, e.g. those used in analysis of cryptographic algorithms
- (e.g. Chosen-Plaintext-Attack (CPA) where the attacker is able to get encryptions of plain text she has chosen). Any security analysis will need both the security goals (attributes/policy) and the attacker model. Sometimes these are left implicit but they remain key ingredients; the question `is this system secure?' has no meaning without them. Not properly considering them is a common cause of security problems.

Security Engineering

- A chain is no stronger than its weakest link. This is also the case for the security of a system. Consider for example the following aspects of a system and some potential issues.
- Design There is no hope of having a secure system if the system design does not address security goals or worse has inherent features/goals that imply security problems. As an example consider the Windows Meta File (WMF) where arbitrary code execution, a clear security risk, is a design feature.
- Another example, the Internet; initially the Internet linked a group of trusted systems. Security goals that are very important now were thus not under

consideration in its design, e.g. no protection of content, any computer can claim to have an IP, no authentication of DNS, etc. of course there are currently security mechanisms (IPsec, HTTPS, etc.) that try to remedy this but `add on security' is always problematic security needs to be considered from the start.

- Software quality A perfect design does not help if the implementation is awed. Often security issues are caused by software bugs with buffer overflow vulnerabilities being one of the major issues. In buffer overflow attacks input from an untrusted source is written into a buffer without the bounds of the buffer being checked. This causes the untrusted data to be written to places it is not supposed to go; it may overwrite a return address on the stack, causing a jump to an attacker selected location at the end of the current routine.
- The problem of software bugs is not solved easily; e.g. an unsolved buffer overflow vulnerability was reported in Windows 7 and in January 2011
- Microsoft shipped _xes for 22 vulnerabilities. The `heart-bleed bug' (is a recent example of a software aw related security incident with wide media coverage. Note that software and systems evolve.
- Security Tool Selection Choose your crypto well, especially if you are a mafia boss. He wrote notes to his henchmen using a modified form of the Caesar Cipher, which was easily cracked by the police and resulted in further arrests of collaborators..." Clearly here the selected security tool was grossly insufficient to reach the security goal.
- This is an extreme example but often inappropriate security tools are used or tools are used well past their `best before/replace by' date such as the hash function MD5, which has been known to be vulnerable for a long time but is only slowly being phased out. Using `home-made' crypto solutions instead of tried and proven standard algorithms would also fit in this category. A good practice is to leave design of crypto to the experts; obscurity of a design is not a good replacement for their experience and expertise.
- System usage Even a perfectly designed and implemented security architecture (should one ever be created) is of no help if it is not used correctly. USB data sticks that offer encryption of their content are readily available and company policy may state that the encryption of such sticks should be used. However, if the user does not enable this feature this is all for nothing.
- Users have different priorities; e.g. ease of use; and many do not use security features or will even try to work around them if they interfere with what they are trying to do. There are many more aspects of a system where a weak link in the security chain may occur. The key points are that one needs to consider the system as a whole and consider security from the start.
- Cryptography is an important part of this toolbox. However recall that security tools by themselves do not make the system secure. A common claim `the data is secure because it is encrypted' is by itself meaningless and may even indicate that the security goals and the attacker model have not been considered

sufficiently. For instance, encryption offers no protection against inside attackers who have access to the key.

- A good security design determines what security tools need to be employed where and when, considering the security requirements and the effects (including trade-offs) different tools have on these requirements.
- Trade-offs \The only truly secure system is one that is powered on, cast in a block of concrete and sealed in a lead-lined room with armed guards."
- Such a system may be sure but not very useful. (Actually it may not be secure at all - Which security attribute is clearly not satisfied? - without the security goals we cannot answer this question...) There is often a clear trade-off between security and usability (why do I need to remember that password...), performance (e.g. using encryption adds computation time) and costs (e.g. replacing pin cards and readers by smart card enabled versions). There is also a trade-off between different security attributes e.g. confidentiality and availability.
- We have to be able to answer the question: Which trade-offs are worthwhile; e.g. how much security do we gain for the performance we give up?
- Why does security often not get the attention it needs? For one; if it's good you do not see it. Would you pay 50 Euro more for a television if it was more secure? Does your answer depend on `how much' more secure?

- It is also hard to quantify security. You can say that a `product is 2 times faster' and convince every consumer with some notion of why and how much better the product is, even though this statement is usually much more complex than it seems. However, what does `this product is 2 times more secure' mean?
- There are many discussions on which product is more secure, e.g. comparisons between Windows and Linux, Firefox and Windows Explorer, Mac and PC, etc.
- Claims are supported by quoting the number of bugs/vulnerabilities reported, the number of security incidents, etc. But how well do any of these really affect the overall `security' of a system. Thinking back to the earlier discussion about what is `security of a system' one can see that no single number could really adequately capture this. Still, what quantification is possible? If we try to focus our attention on a single aspect of security and a single application area, one may be able to give some numbers that make sense (just remember that, the more general the statement the less objective a score is likely to be).
- For cryptographic primitives one can look at the (computational) cost of breaking a system. This is often expressed by the entropy that it offers in a given setting, e.g. `this crypto system offers 80-bits of security' effects that the amount of computation needed to break it is similar to brute-forcing an 80 bits key, i.e. trying 280 different possibilities. This is generalized to a measure for security of systems by considering the cost (computational or otherwise) of breaking the system's security; e.g. it would take 2 years and a budget of 10 million euros to break this system (i.e. violate a specific security goal of the system).

- For web applications several security metrics have been defined by checking for common security issues and assigning a risk to each of them. For example, the CCWAPSS common criteria for web application security scoring, computes a score based on a list of eleven criteria. Each criteria has to be checked (rating the web service on a scale from 1 to 3 for each item) and assigned a risk level based on the difficulty and impact of an attack.

Security Requirement Engineering

- As already mentioned several times, to really evaluate the security of a system you have to consider it as a whole, know the security goals and the potential threats against these goals.
- To gather these we need to perform Security Requirement Engineering. Throughout the design, implementation, deployment and use of a system we should consider the requirements that the users will have from the system and how attackers will try to exploit the system. Based on this we can come up with and/or evaluate a security design, which combines several security solutions to achieve the best possible trade-offs.
- Other approaches may work just as well, what is important is that the security requirements are considered throughout in a structured and consistent way.
- Identify actors and goals. The first step in gathering the requirements is determining the stakeholders and their interests. The stakeholders are those parties with a legitimate interest in the system that we are designing.
- Their interest and goals thus have to be considered (though not necessarily completely reached - we may need to make trade-offs between the different goals of the participants).
- The stakeholders and their interests become the initial actors and goals in the requirements gathering process. If an agent has the right capabilities, it may adopt a goal, i.e. take responsibility to achieve it. If an agent does not adopt the goal it may be delegated to other agents (either existing or new or be split into new sub goals. Agents do not work in isolation; agents and their goals may depend on/interact with each other. These dependencies should be identified and could lead to new goals and/or agents. They also lead to potential vulnerabilities, e.g. when agents' goals conflict.
- So far the process matches a typical functional requirement engineering approach. In order to deal with security requirements we also need to consider attackers and possible attacks on the system.
- Identify attackers, vulnerabilities and attacks. Outsiders may try to attack our system and they need to be considered along with their goals.
- However, also the risk of attacks by insiders needs to be accounted for.
- Each agent in the system could potentially become an attacker, using its capabilities and place in the system to reach their goals at the expense of the goals of other agents. Both type of attackers are modeled as agents in the system but with malicious intent as their goal.

- Based on vulnerabilities and the malicious intent of attacker agents we identify potential attacks and assign countermeasures to protect against such attacks. The countermeasures themselves may lead to new actors/goals and/or open the possibility for new attacks which need to be considered.
- Refinement of the system continues until all goals have been assigned, dependencies taken into account, and vulnerabilities addressed.

Summary

- The goal of this section was to introduce the most basic and fundamental concepts of computer networks and network security, as well as the motivations for their existence.
- You now know the principle nuts and bolts of a computer network, the idea behind network protocols and protocol layering. A simple overview of the Internet and the Internet protocol stack have been provided, together with an outlook into the future of the Internet dominated by machine-to-machine communication, i.e. the Internet of Things. After our security discussion, you will never look at the word `secure' in the same way again: Whenever you encounter `secure' always think - what set of security requirements (which security attributes for which resources) are really meant by `secure' (what are the security policy and model) and what type of attacker is considered (what is the attacker model).

Quality of Service (QOS)

The notion of quality of service, or QoS, concerns certain characteristics of a network connection under the sole of the network service provider liability.

A QoS value applies to the whole of a network connection. It must be identical at both ends of the connection, even if it is supported by several interconnected subnetworks each offering different services.

QoS is described by parameters. Defining a QoS parameter indicates how to measure or determine its value, mentioning if necessary the events specified by the network service primitives.

Two types of QOS parameters have been defined:

- Those whose values are transmitted peer users via the Network service during the establishment phase of the network connection. During this transmission, a tripartite negotiation can take place between users and the network service provider to define a value for the QoS parameters.

- Those whose values are transmitted or negotiated between users and network service provider. For these QoS parameters, it is possible to obtain, by local means, on the value to the supplier and values to each user of the network service.

The main QOS parameters are:

- **Time of establishment of the network connection**. Is the time that elapses between a network connection request and confirmation of the connection? This QoS parameter indicates the maximum time acceptable to the user.

- **Probability of failure of the establishment of the network connection**. This probability is established from the applications which have not been met in the normal time limit for establishing the connection.

- **Flow data transfer**. The flow rate defines the number of bytes transported over a network connection in a reasonably long time (a few minutes, a few hours or days). The difficulty in determining the speed of a connection network comes from the asynchronous transport packets. To obtain a value acceptable, observe the network on a sequence of several packages and consider number of bytes of data transported taking into account the elapsed time since the application or the data transfer indication.

- **Transit time when transferring data**. The transit time corresponds to elapsed time between a data transfer request and indicating transfer of data. This transit time is difficult to calculate because of the geographical distribution ends. The satisfaction of a quality service on the transit time may moreover contradict flow control.

- **Residual error rate**. Is calculated from the number of packets that arrive erroneous, lost or duplicated on the total number of transmitted packets. It is a rate Error packet. Also denotes the probability that a packet does not arrive correctly to the receiver.

- **Transfer Probability incident**. Is obtained by the ratio of the number of incidents listed on the total number of transfer taken. To have a correct estimate of this probability, just consider the number of network disconnection relative to the number of transfer taken.

- **Probability of failure of the network connection**. Is calculated from the number of release and resetting of a network connection based on the number of transfer made.

- **Release time the network connection**. This is the maximum acceptable delay between a disconnection request and the actual release.

- Probability of failure upon release of the network connection. The number Liberation of failure required by the total number requested release.

The following three additional parameters used to characterize the quality of Service:

- **Protection of the network connection**. Determines the probability that the network connection be in working order throughout the period when it is opened by the user. There is ways to protect a connection by duplicating or having a Backup connection ready to be opened in case of failure. The value for a telephone network is 99.999%, the so-called five nines, equivalent to a few minutes of downtime per year. The protection is much lower for an IP network, with a value of the order of 99.9%, three or nine. This value arises besides problem for IP telephony, which requires stronger protection telephone connections.

- **Priority of the network connection**. Determines priority of access to a connection network, the holding priority of a network connection and priority of data connection.
- **Maximum acceptable cost**. Determines if the network connection is tolerable or not. The definition of the cost is quite complex since it depends on the use of resources for the establishment, maintenance and release of the connection network.

Flow Characteristics:

Traditionally, four types of characteristics are attributed to a flow: reliability, delay, jitter and bandwidth.

Reliability

- Reliability is an important characteristic of flow.
- Lack of reliability means losing a packet or acknowledgement which then requires retransmission.
- However, the sensitivity of application programs to reliability is not the same. For example, it is more important that electronic mail, file transfer, and Internet access have reliable transmissions than audio conferencing or telephony.

Delay

- Source to destination delay is another flow characteristic.
- Applications can tolerate delay in different degrees.
- In this case, telephony, audio conferencing, video conferencing and remote log in need minimum delay while delay in file transfer or e-mail is less important.

Jitter

- Jitter is defined as the variation in delay for packets belonging to the same flow.
- High Jitter means the difference between delays is large and low jitter means the variation is small.
- For example, if four packets depart at times 0, 1,2,3 and arrive at 20, 21,22, 23, all have same delay, 20 units of time. On the other hand, if the above four packets arrive at 21,23,21, and 28 they will have different delays of21, 22, 19 and 24.

Bandwidth

- Different applications need different bandwidths.

- In video conferencing we need to send millions of bits per second to refresh a colour screen while the total number of bits in an email may not reach even a million.

https://ecomputernotes.com/computernetworkingnotes/communication-networks/quality-of-service

http://www.idc-online.com/technical_references/pdfs/data_communications/Notes_on_Network_Security_Introduction.pdf