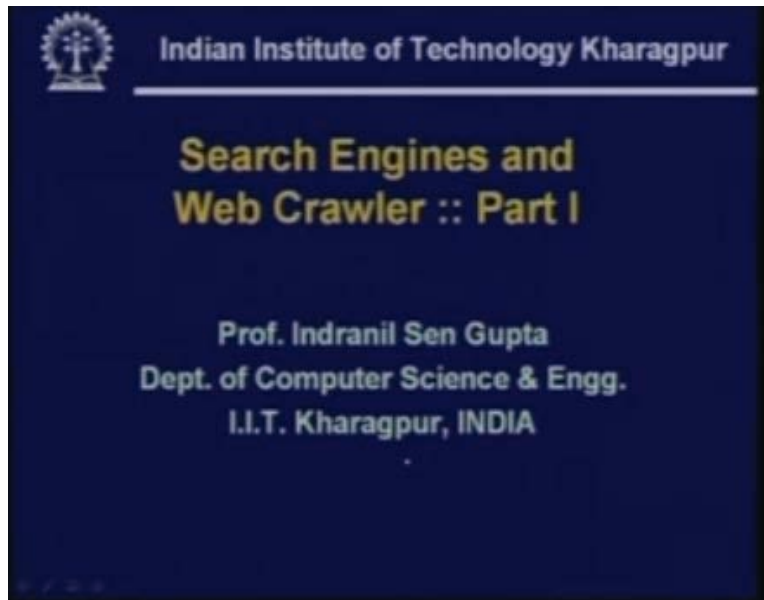


**Internet Technology**  
**Prof. Indranil Sengupta**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**  
**Lecture No #38**  
**Search Engines and Web Crawler :: Part 1**

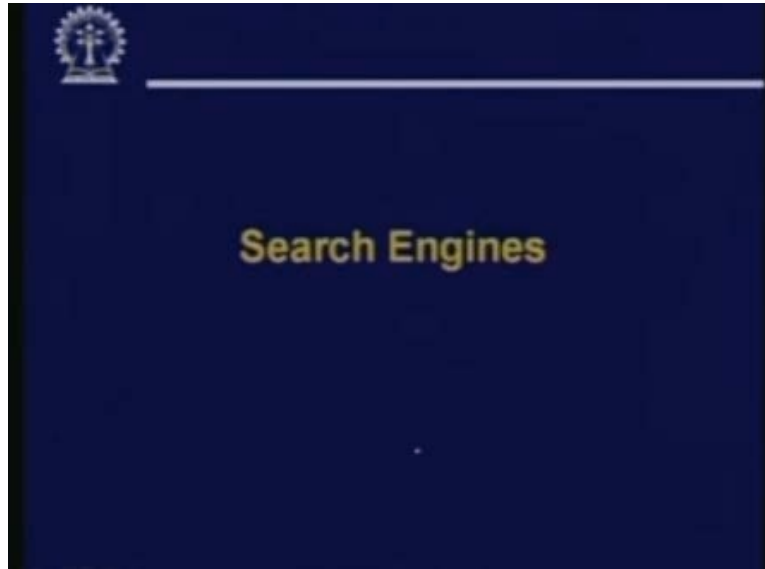
Today we shall be starting discussion on search engines and web crawler.

(Refer Slide Time: 01:06)



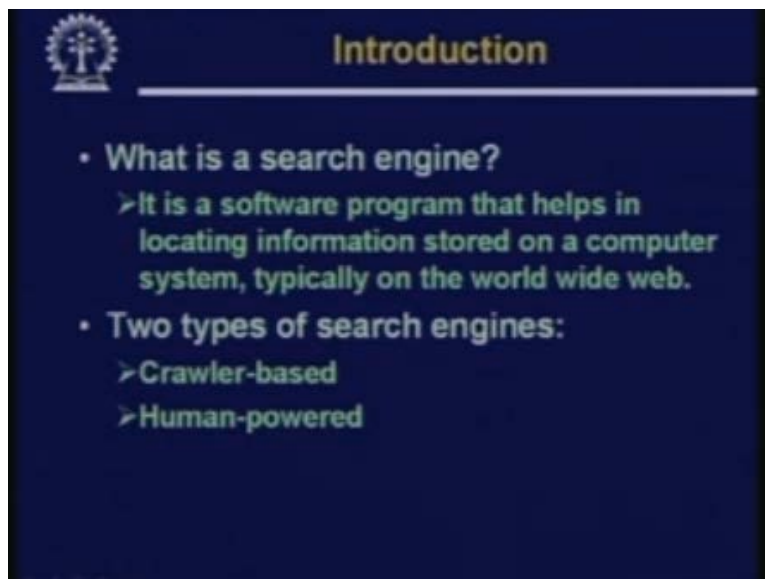
Now you all know that search engine forms some sort of a live wire in the present internet scenario. So without search engine many of us would definitely not be able to use the internet in a meaningful way. They help us in getting track, getting the information we want to access keeping track of the information we are interested in. So there are a number of underlying technologies that drive such search engines and crawling is one such underlying technique which we will be discussing during the course of our discussion. So our topic of discussion is search engines and web crawler.

(Refer Slide Time: 01:52)



We start by talking about search engines first.

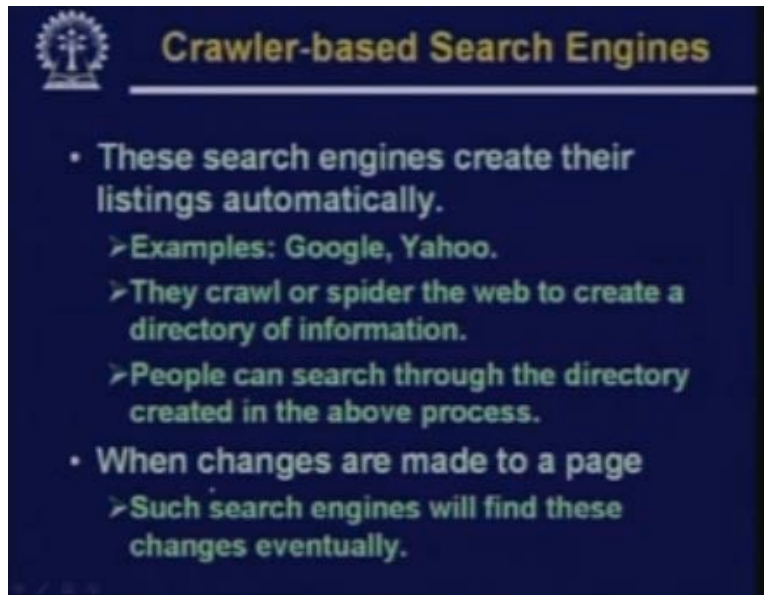
(Refer Slide Time: 01:58)



Let us first try to understand what is a search engine? Now naturally our search engine is a software program. Many of us are familiar with the search engines that are available in the internet like Google, like Yahoo. So whenever we want to find something we type in some phrase or a keyword out there and the search engine helps in finding out the information for us. Now search engines can be very roughly defined as follows. It is a software program that helps in locating information stored on a computer system typically on the World Wide Web. Now here when we are talking about the information

stored on a computer system we mostly mean that some information is stored on some web servers which the search engine will try to keep track of its way. Whenever a user wants to search for kind of that information those particular pages documents or files would be returned as search result. Now in terms of how they work search engines can be categorized into two types. One is called crawler based and the other is human powered. Let us try to understand what are the main differences between the two?

(Refer Slide Time: 03:35)

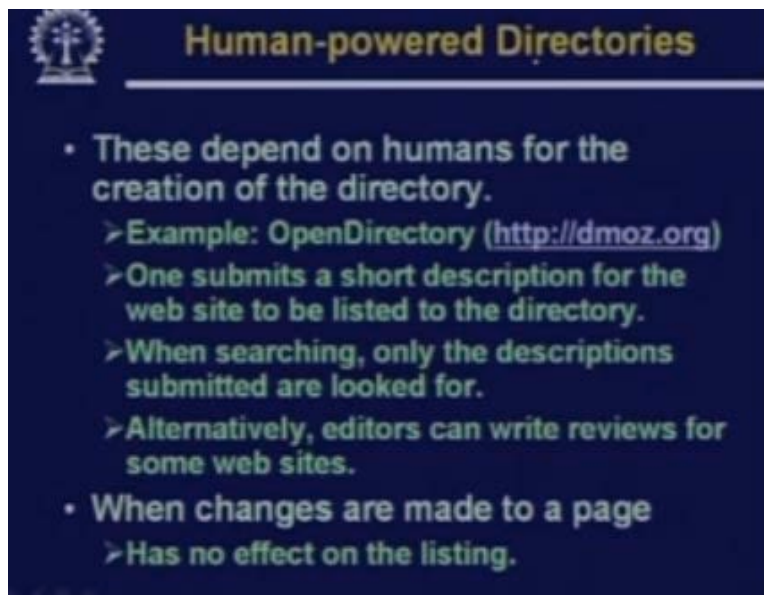


The crawler based search engines, there is some degree of automation involved here. These search engines create their listings automatically. There is no human intervention involved in the process. Some of the search engines like Google and Yahoo work in this way. Now when I say that the listings are obtained automatically, well I mean basically when the search engine or the system on top of which the search engine is running wants to do some sort of an indexing of the information that is available. Now it will automatically try to find out what information is available where it is not like the situation where a user tells the system that well if you look for this x y and z you can find it here.

Now here the human is helping the system to build up the so called index or the directory the dictionary whatever you call. So these kinds of search engines the crawler based they crawl or spider. This is the work which is used. They crawl the web going from one web server to the other systematically to create a directory of information. Now here this directory is created in an automated way. Subsequently when people submit some search query their query would be directed to this directory which was already created in the above process and whatever matches or found out there, they will return. And the second issue is that what happens when pages undergo some changes over a period of time some updation. Since the process is automated the entire thing will be repeated periodically.

And so such search engines will ultimately find the changes. So here what we have seen is that the search engine or the system on top of which the search engine is running, it will automatically go from one web to the author looking for information. And based on whatever information it can find from there it will try to build up some kind of a dictionary. Later when a user submits a query that dictionary will be consulted to return the results or the matches and if some pages are updated then the next time when the crawler again visits that web. The newer version will be considered and dictionary will also be updated.

(Refer Slide Time: 06:31)



**Human-powered Directories**

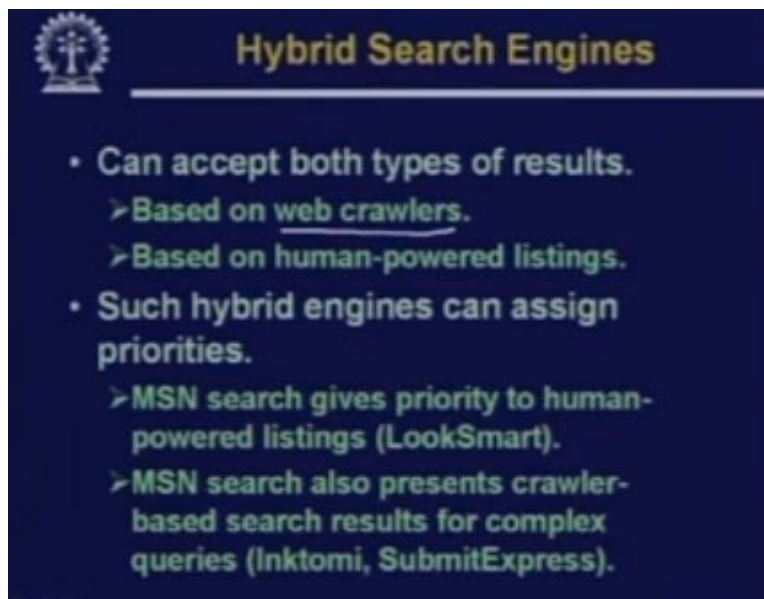
- These depend on humans for the creation of the directory.
  - > Example: OpenDirectory (<http://dmoz.org>)
  - > One submits a short description for the web site to be listed to the directory.
  - > When searching, only the descriptions submitted are looked for.
  - > Alternatively, editors can write reviews for some web sites.
- When changes are made to a page
  - > Has no effect on the listing.

Now in contrast you have human powered directories. This is the other alternative. Now as the name implies these kinds of directories depend heavily on human for its creation. Now one very common example of such human created or human powered directory is the open directory whose website is dmoz.org. You can visit that website and have a look at it. Now here the idea is like this. Suppose a person creates a web page. Suppose I myself today create a web page. I know that my web page contains information about these five keywords. So what I will do I will submitting to the directory service that well this is the page I have created. And these are my 5 keywords so that the dictionary gets created based on the information the human or the person who has created has applied.

So basically the process involves two steps one submits a short description as I mentioned for the website to be listed. The short description typically contains keywords and any other information you want to submit about it and subsequently when the dictionary is searched in response to some user queries only the descriptions which have been submitted will be searched or looked for. Now I have said that this process is typically carried out by the person who has created the web page. Alternatively you can have a system where there will be some editors who can edit or review web pages or web sites and the result of the review will be submitted to those related service.

So after review some indices will be created based on the information content of a particular website or page and the editor will be taking the additional responsibility of submitting the information to the directory service so that the web page is indexed. But however here when some changes take place to a page normally, there is no effect on the listing unless the user explicitly informs the directory service that a change has taken place. So unlike crawling this will not be incorporated automatically, so both the approaches have some good features. One can surf automatically and the second one since the creator is submitting the keywords the keywords are likely to be more meaningful.

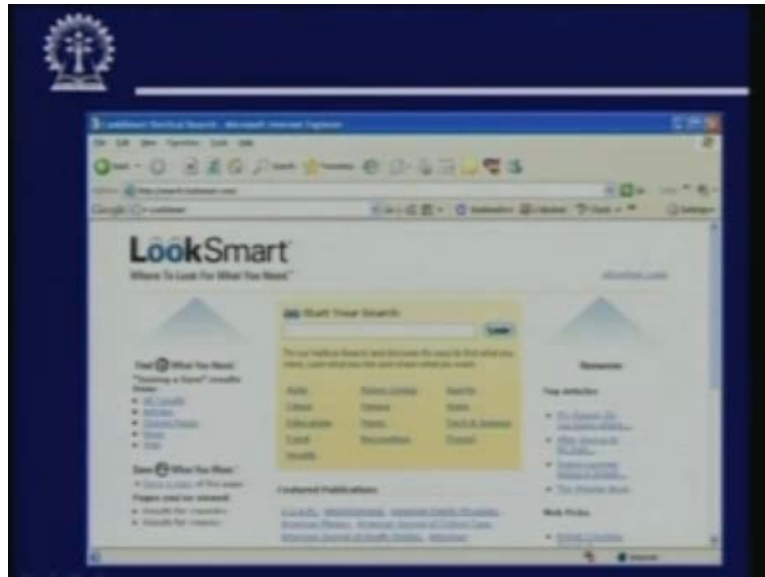
(Refer Slide Time: 09:31)



Combining the best of both worlds we can have a hybrid system. We can have something called hybrid search engines. Now a hybrid search engine can look through dictionaries which are created by web crawlers and also using human powered listings. So both these are allowed here but however most of such hybrid engines they assign some priority means out of the web crawler these indices are human powered indices which one should be given higher priority. For example MSN which is a product of Microsoft, this is an example of a hybrid search engine.

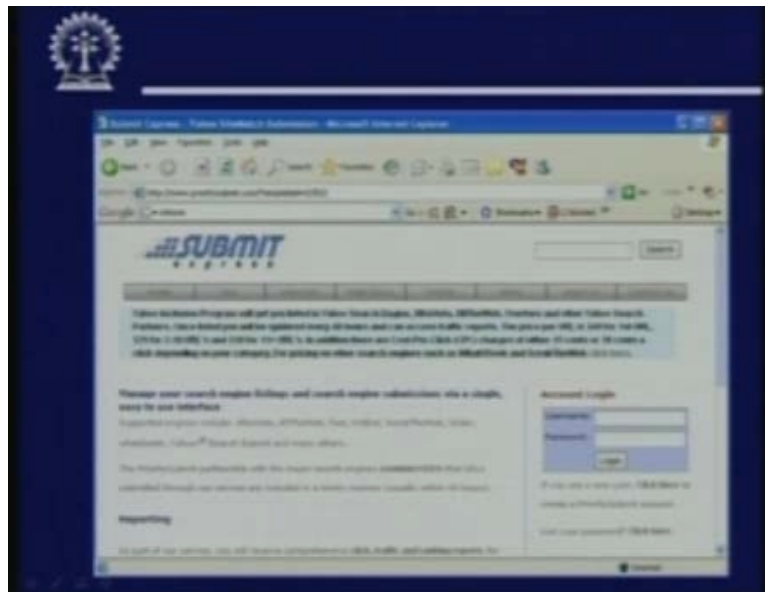
MSN search gives priority to human powered listings and the technology of the tool that is used by MSN is called LookSmart. Now as I said human powered directories will convey more accurate information. So this priority assignment is quite natural. But however MSN search also has the feature of crawler based search results for complex queries. So first it will look for the human powered listings then it will search for the other one. Now the tools that it uses for crawler based search are Inktomi, SubmitExpress. These are again some tools which are developed based on web crawling technology.

(Refer Slide Time: 11:10)



Now let us look at some of the typical website. This is the website of that LookSmart. Well this LookSmart I had said this is a website or a tool which is used to create human powered listings. So here whenever we want to submit information, this website has some links through which we submit the relevant information.

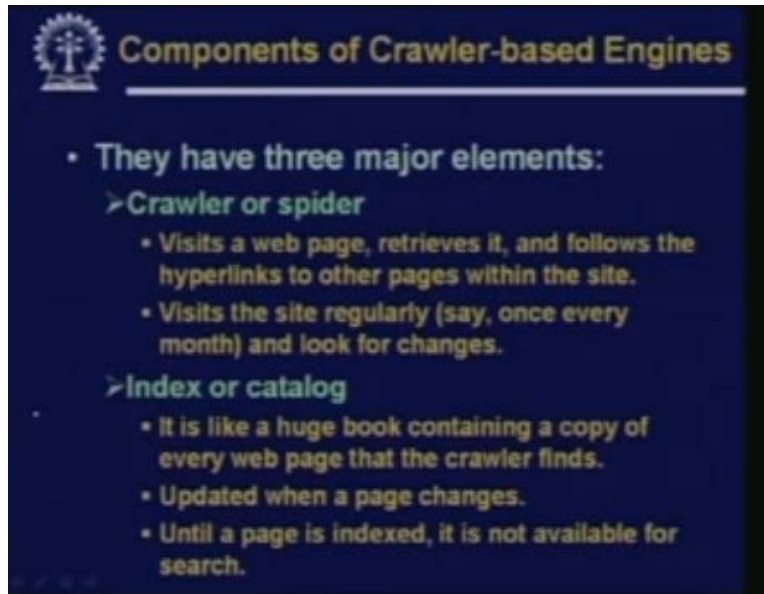
(Refer Slide Time: 11:38)



Similarly this is the submit express website. So here again submit express contains a lot of information so that you can even incorporate results in your own search engine which you have created. So I suggest you can you can visits this website and upload that what

are the facilities. These websites provide to a web developer and also to a person who is similarly writing a new search engine.

(Refer Slide Time: 12:09)



Let us look at crawler based engines because in human powered system there is not much to automate. It is the responsibility of the humans to submit the keywords the system will simply store the information in a database. But in a crawler based engine there is some sort of automation involved and the system is supposed to create the directory by itself automatically. So if you look at the different components that a crawler based engine has broadly they will have three major elements. First one is called crawler or spider. Now as the name implies a crawler will crawl from one website to other. Usually this crawling will take place based on the hyperlinks that are present on one website or a one web page. Suppose the crawler visits my page from my page there are four different links.

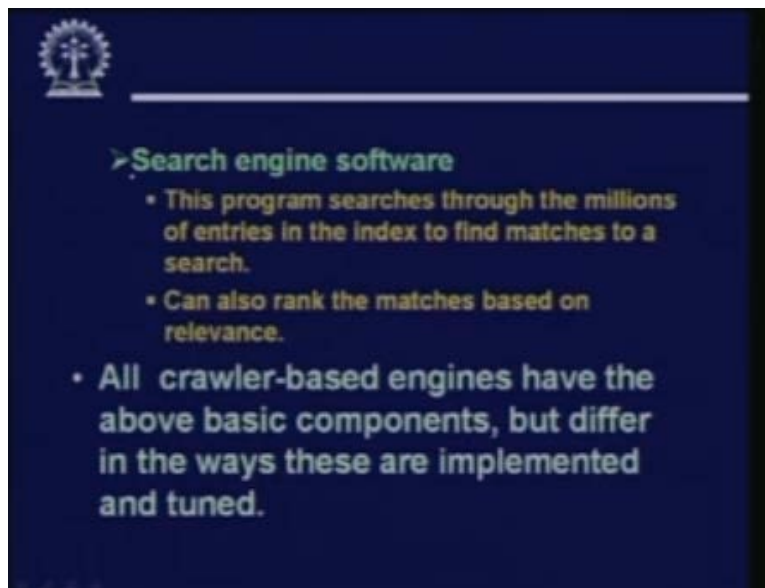
So the crawler will systemically visit those four links. Of course there is some policy or some principle that is used to decide which link to follow and which link to not to follow. For example if one of the link points to an image there is no point in traversing that link because once that image is downloaded it will not convey any additional information to the web crawler. But if it is an html page then that html page may be of interest because it can contain some new keywords, some new information that people may be interested in. So the crawler or spider basically visits a web page retrieves it and as I said and follows the hyperlinks to other pages within the site and this visiting of this page is carried out regularly. When this regularly is subjective it can be once a month or you can make it more frequent. If you want to do it that way and it will look for any changes that had taken place in the pages.

Since the last time it was visited. So the task of the caller is simply is to go from one site to the other and to retrieve the information that are stored there pointed there. So after

retrieving some additional post processing has to be done. So the second part is index or catalog. Now index is basically like a huge book which will contain a copy of all the web pages that the crawler finds. So essentially what will happen the crawler whatever pages it retrieves, it gets stored in the index or the catalogue. Now once stored, it will remain there. And it will be updated only when some page to the change occurs crawler deducts that. Well if you if the page gets deleted then the entry will also get deleted if the page is updated. Then a new version will come in.

But however retrieving a page does not necessarily mean that it will be immediately used or is available retrieval through a search. Indexing is an important step unless you index a page it is not available for search. For example in a book you know there is an index. So if you want to look for something you can look at the index, find the page number, it will straightaway go to the page number. But if someone has not given you the index it will be very difficult for you to find. So the same thing happens here. Retrieving pages are one thing and indexing them is the next thing. So unless you index them a search query cannot really utilize that information to retrieve the pages that the query is mapping to.

(Refer Slide Time: 16:36)



And third thing is that other than the crawler and index you need actual software through which the users will submit the queries. This is the so called search engine software. Search engine software basically will search through the millions of entries in the index because the index will typically contain huge number of entries and you try to find out matches to a search. Now as you know that when we give some keyword for searching usually a very large number of pages are turned. So how do we know which is good which is not so good. So the search engine software has an additional responsibility it will find out matches alright. But in additional it will also have to rank the matches in some way using the concept of relevance. Relevance means there should be some mechanism for us to tell or to find out whether a particular page is relevant to a given search or not.



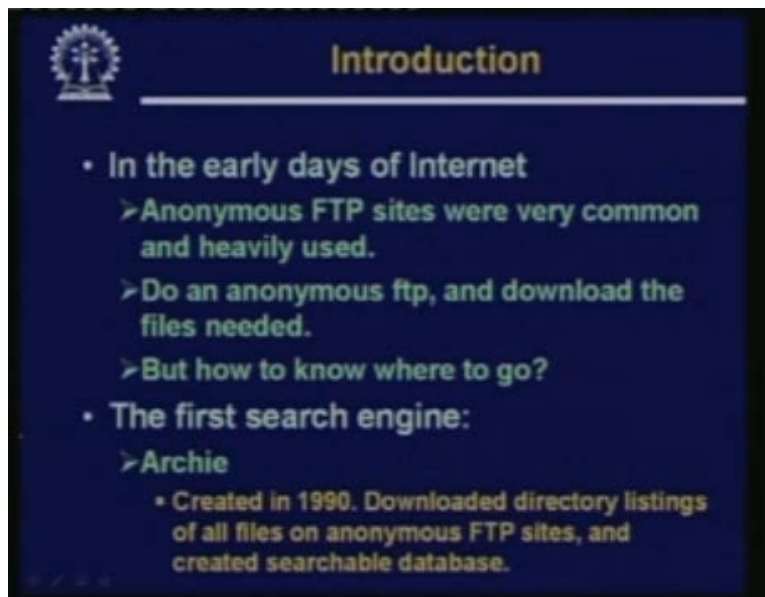


(Refer Slide Time: 19:25)



So now let us move away from the basic components of a search engine to the history of search engines. How search engine was developed from the day when internet started to evolve?

(Refer Slide Time: 19:32)



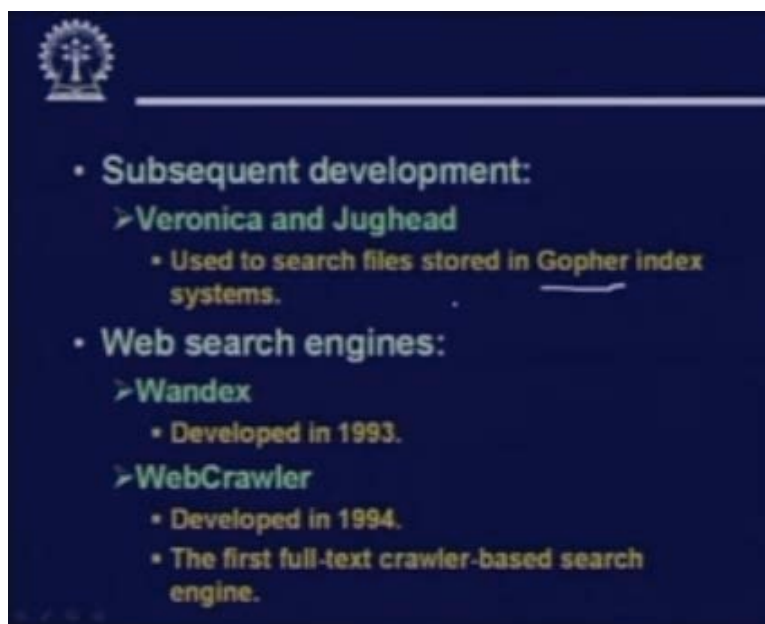
Now it will be unfair to not to talk about the anonymous FTP as a technology which was searched for in the early days of internet. Now in the early days of internet as I said earlier in one of my earlier classes that FTP and anonymous FTP were considered to be very powerful and popular tools which a large number of people use. Now one problem

with the normal FTP was that to long into a system for transferring files, one has to know a valid user name and password on that system which in general in the internet scenario is not possible always. So a more feasible method which was proposed was that if we have anonymous FTP where instead of typing in a user name. You can type anonymous in place of password; you can type your email address just for record.

And you will be having access to some shared resources which the creator of that FTP site had already put in some folders and directories on the server. So there were thousands of very rich FTP anonymous servers which got created all around the world and just like today we serve the web to dig out information. At that time people used to look for the anonymous FTP servers and to dig out useful information from there. So this was the basic technology. But the question was how to know which FTP server to connect to. So the first engine that came into you can say came into existence was one which was meant for anonymous FTP sites and the search engine was called Archie. This was created in the year 1990. So what Archie did?

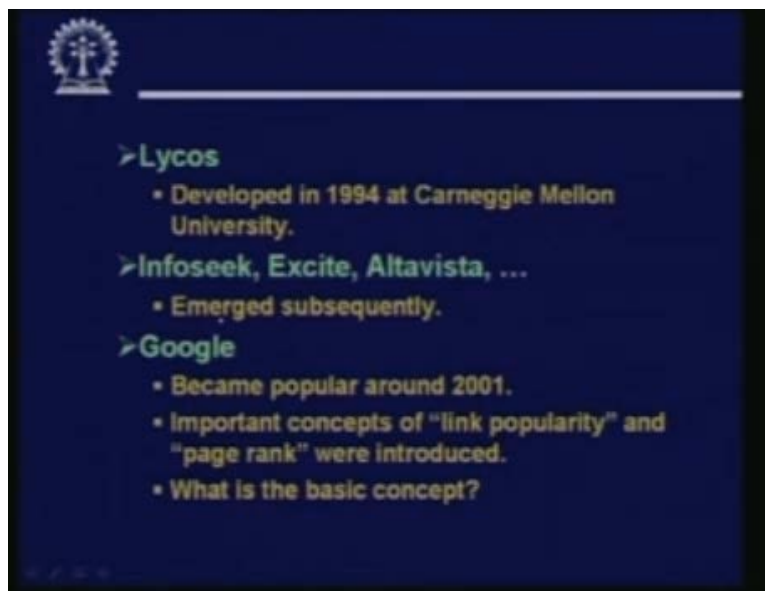
Archie proactively downloaded the directory listings of all files on all anonymous FTP sites and find and it created searchable data base in some central place so that later on when user submit queries through an Archie client the central searchable data base was consulted and the search result was returned. But here the search result was somehow different as compared to what you see today. Today we get the exact content what we are looking for on our screen but at that time what we got out of the search was that the names of the anonymous FTP sites and the possible folders in there where the information I am looking for is located. Not it was the responsibility of the user to actually go that site and download that information. Archie simply help in telling the person who was looking for the information that where the information can be found. Subsequently the person has to go there and get it manually.

(Refer Slide Time: 23:11)



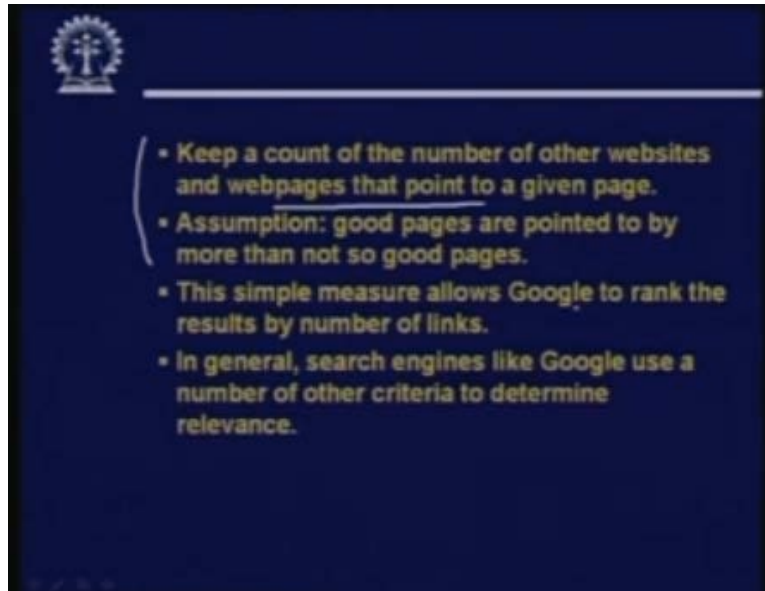
Subsequent to Archie there were some other developments Veronica, Jughead. These were some other search engines which were developed but these are again targeted to Gopher index in systems. Gopher is you can say the predecessor of the present day World Wide Web that was also some kind of contents based search. But the graphic user interface was missing. It was a text based search. You can download documents save it in a file but you cannot view it in a graphic form with all multimedia contents just like you can do it today. Veronica and Jughead helped users to search files in that way. Now subsequently the web search engines came into the being with the growing popularity of the World Wide Web and the web servers where people started to keep more and more free information for others to share and access. So some of the early day website engines where as follows Wandex was such an engine which was developed in 1993. Then I talked about web crawler. Web crawler was developed in 1994. This was in fact the first full text crawler based search engine. So that is why in many books and many tutorials, this web crawler finds special reference.

(Refer Slide Time: 24:50)



Lycos was developed in nineteen ninety four in CMU, Carnegie Mellon University as part of a research project. Then Infoseek, Excite, Altavista, these are all search engines. They were developed and emerged subsequently. Now this search engine that possibly become very popular in the shortest span of time was Google. Google became popular around 2001. Now Google introduced very interesting simple concepts which allowed a user to get more relevant search results as compared to other competing search engines. So people found that if I use Google I can get my information in much less effort as compared to other engines. So due to this reason Google started to increase in popularity. So they introduced some important concepts for the first time. Link popularity. How popular is a link? How many persons are providing a link to your page and page rank? How good or how bad this page is with respect to my search keywords? So now let us see what the basic concept. Here is the concept is like this.

(Refer Slide Time: 26:30)



Google keeps a count of the number of other pages that means websites and webpages that point to a given page. What this means is that suppose I have I have a page in one place say I have a page here. There are ten other pages which are providing hyperlinks to my page. So I can say that my page is relatively popular, ten other sites are referencing my page. But if I find my site or my page is referred to by thousand other pages I can say that my page is very popular. So this is a simple measure in using which you can say page or web you can say popularity index can be measured that how popular your page is. If it is popular it must be linked to many other pages, so during the indexing process you find this information and you can tag this information and with every page in your dictionary.

So here of course the assumption is that good pages are pointed to by many more pages as compared to not so good pages. So just the count of the number of pointers to a page and give you simple measure of web popularity or goodness of a page. So this simple measure this allows Google to rank the result by number of links that are there pointing to them. But however you should remember that commercial search engines like Google is not simple. There are a number of other complex criteria which are also added in addition to these to determine relevance for page. They rely on very strong probabilistic techniques very you can say theory intensive data mining techniques in fact this is an intense area of research today. So how to rank a page or how to grade the relevance of a page in a very good and meaningful way.

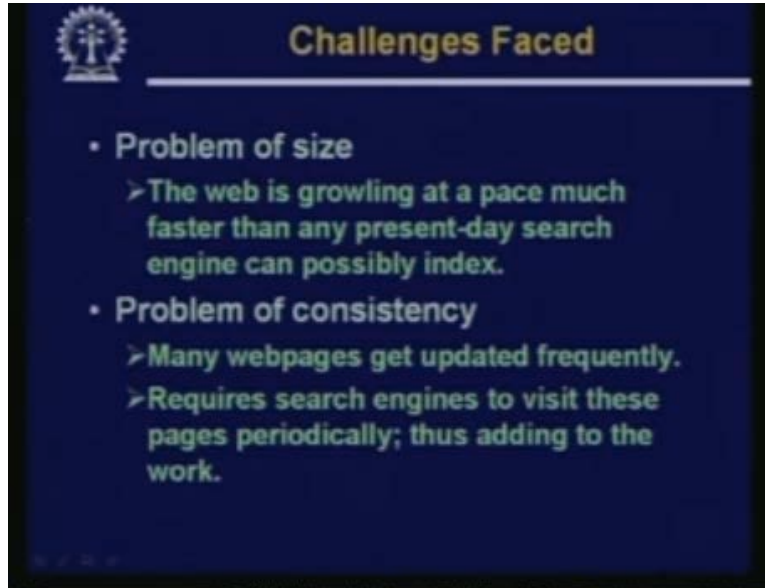
(Refer Slide Time: 28:54)



Well there is another search engine yahoo which also became quite popular. So but well yahoo you can say that yahoo became a search engine only after 2004. Before 2004 what yahoo did? It basically used the Google search engine to provide users to the search results. So the front end was yahoo as if you are using yahoo. But yahoo was calling Google in the back end. So whatever results we are getting was the same you would have got if you have called Google directly. So that was prior to 2004, but Yahoo launched its own search engine in the year 2004. Actually they bought the technologies that were present in many of the existing search engines like Inktomi, AltaVista and they integrated these technologies into the new search engines. So yahoo acquired Inktomi and Altavista. They compiled and used the technologies to build a better search engine.

And that is how the yahoo search of today is it was configured or it is working Microsoft has its own version MSN search. This is the most recent version owned by Microsoft. This is increasing in popularity again and this introduced due to the huge usage of the windows operating system. It introduces an integrated search framework that is called windows live search. This is a new search platform which uses MSN search in the background automatically. So this is something which is increasing in popularity. So all these search engines are there. They have been proposed. Many of them are used; some of them have become very popular. But all search engines which are there in the market. They are trying to compete with each other they have to address some common problems challenges.

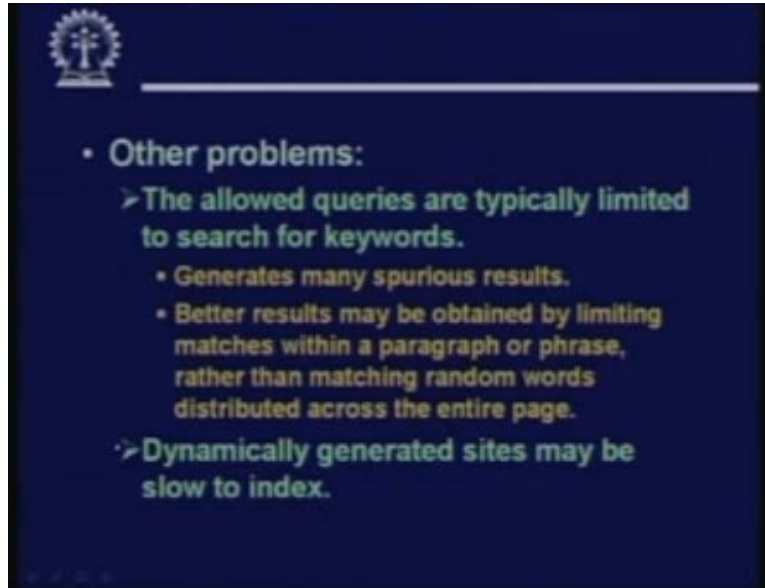
(Refer Slide Time: 31:12)



So broadly the main challenges faced are two there are others of course; but most important are two. One is the problem of size. Today there are billions of pages that exist in the World Wide Web does not mean that my search engine has to create a dictionary where size will be one billion entries it will be too large number. Even if you say that it is feasible I can provide very large amount of storage but the next question arises how long my search engine will take to create those billion indices. They will have to call through all the websites, get those billion pages and then carry out indexing. It naturally should take a very large amount of time to do that. So the main issue is the web is growing at a very fast pace which is much faster than any present day search engine can possibly index.

So we should address this problem and we should try to manage the problem in the best possible way. Possibly we will say that well this index only a subset of the pages. Let us not go, let us not try to index each and every pages that exist. Some sort of a compromise like this and the second issue is the problem of consistency. There are many webpages which get updated frequently like the news the pages which report the news they get updated every few minutes or hours. So this requires a search engine to visit these pages regularly so as to keep this frequent updation indexed. This in fact adds to the work. So in addition to the volume of the work we are saying in additionally there are some websites which require very frequent visiting for indexing. So both this problems compounds the overall problem.

(Refer Slide Time: 33:23)

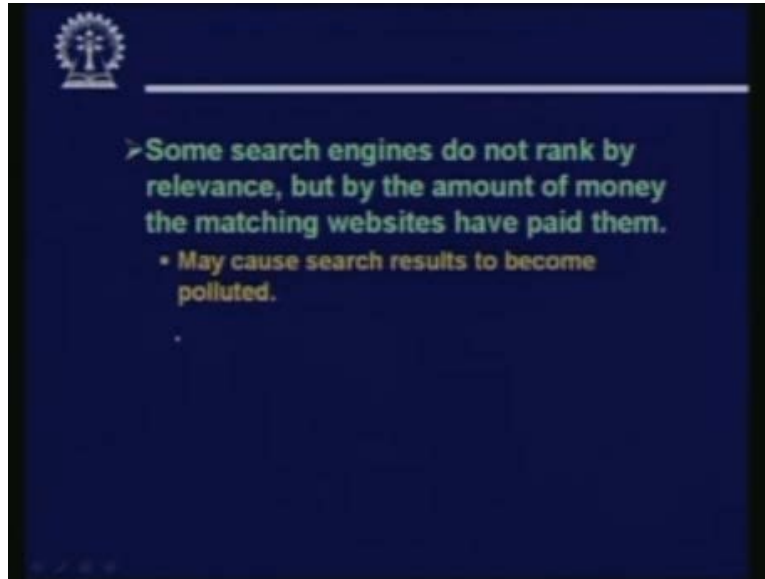


So other problems are also there. Problems like the allowed queries which are typical search engine will means allow a user to give a typically limited to search keywords. Say when I want to search for something I typically give some keywords. I want to search for say for example TCP IP I can give some keywords network TCP IP like that. So in general this keyword based search will generate many spurious results. There can be many pages from orthogonal domains which can also have such keywords in their contents. So those pages will also be returned as search results. So this is one problem. So obviously better results can be obtained if you limit the matches within a paragraph or phrase which means TCP IP these two keywords must appear within a sentence or a paragraph.

But if you find a document where TCP appears in one place IP appears in some other place, most likely this is not the document you are looking for. This is some other TCP or some other IP people are talking about. So limiting matches within a region of the document is a very interesting concept. That means you look for matches within a sentence a paragraph or even a phrase. Many of the earlier search engines they matched randomly which may be distributed across the entire page which often does not give you very good, you can say relevance terms of the match. And secondly there are some websites which are dynamically generated. They may be slow to index because dynamically generated means whose contents are not static depending on user request the pages get generated. Those kinds of dynamically generated pages are very difficult to index because they are created so fast.

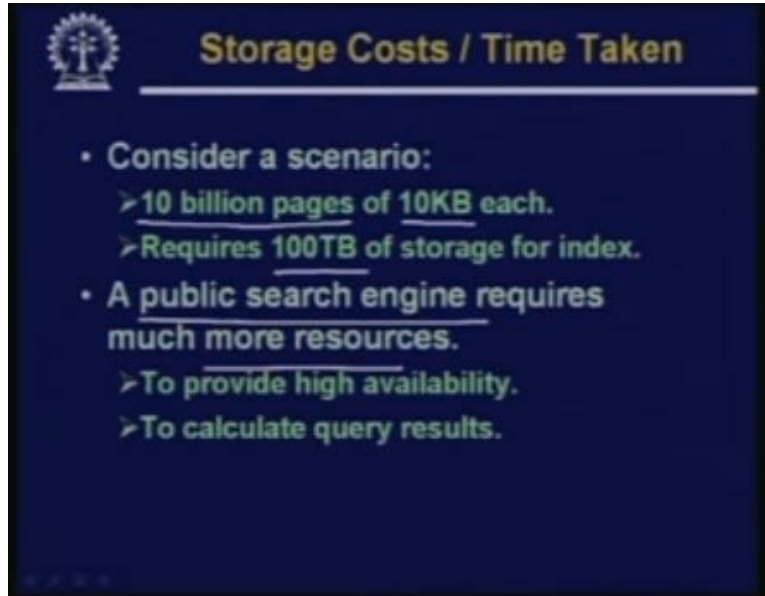


(Refer Slide Time: 35:46)



So these are some of the problems that you need to face and this is of course an unfortunate event this is also a problem you can say. There are many search engines you will find. I am sure you have encountered this experience. They do not rank by relevance but rather by the amount of money the matching websites have paid them. Which means, if I want my website to become popular I will give some large sum of money to the search engines? So this is something like a bribe I am paying to them and they are compromising on their algorithm to find the relevance. Since I have paid more money they will say that my pages are more relevant with respect to searches. So this is an adulterant which is there in the web and this in fact can cause some of the search results to become polluted. So whatever search result has been returned many of them will find they are not at all relevant. They are junk pages that are coming.

(Refer Slide Time: 36:55)

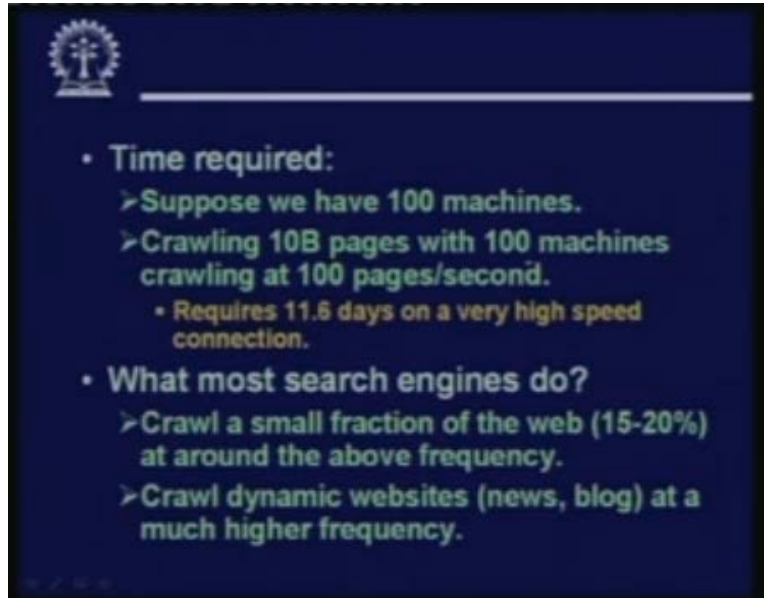


The slide features a dark blue background with a white logo in the top left corner. The title 'Storage Costs / Time Taken' is written in a yellow font at the top. Below the title, there is a list of bullet points in white and yellow text. The first bullet point is 'Consider a scenario:', followed by two sub-points: '> 10 billion pages of 10KB each.' and '> Requires 100TB of storage for index.'. The second main bullet point is 'A public search engine requires much more resources.', followed by two sub-points: '> To provide high availability.' and '> To calculate query results.'

- Consider a scenario:
  - > 10 billion pages of 10KB each.
  - > Requires 100TB of storage for index.
- A public search engine requires much more resources.
  - > To provide high availability.
  - > To calculate query results.

Mostly full of advertisements and other things. Storage costs a time taken are some other issues. Let us take a simple example. Here you consider a scenario where we have 10 billion pages of which you have to index and on the average there are 10 kilo bytes each. So if you want to index all of them this will be requiring hundred terra bytes of storage. This is an issue from the point of view of rich space. Storage place only for the index but a public search engine will require actually a much more than this. Because not only the index due to high availability, it may use some sort of redundancy or mirroring two or more copies of the data may be stored in several different locations. And on top of it you have to run queries that may require some more resources. So to calculate query results you need additional resources. So resource is a big issue.

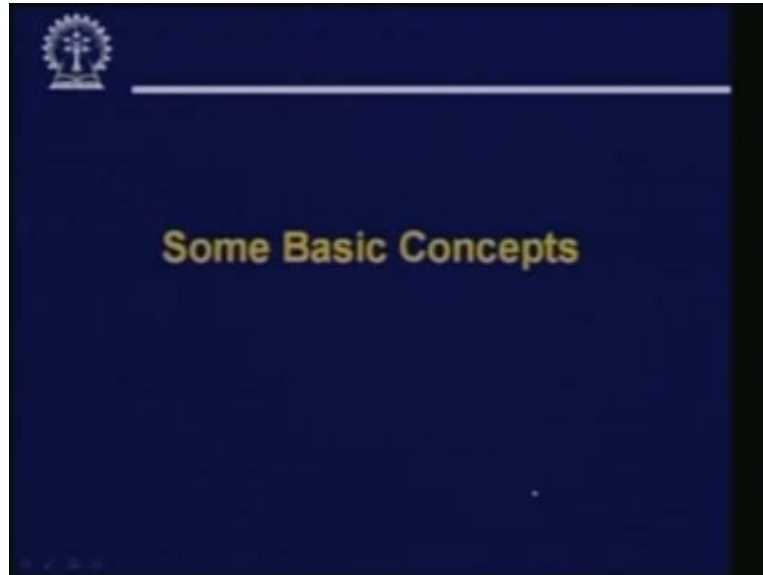
(Refer Slide Time: 38:11)



- Time required:
  - Suppose we have 100 machines.
  - Crawling 10B pages with 100 machines crawling at 100 pages/second.
    - Requires 11.6 days on a very high speed connection.
- What most search engines do?
  - Crawl a small fraction of the web (15-20%) at around the above frequency.
  - Crawl dynamic websites (news, blog) at a much higher frequency.

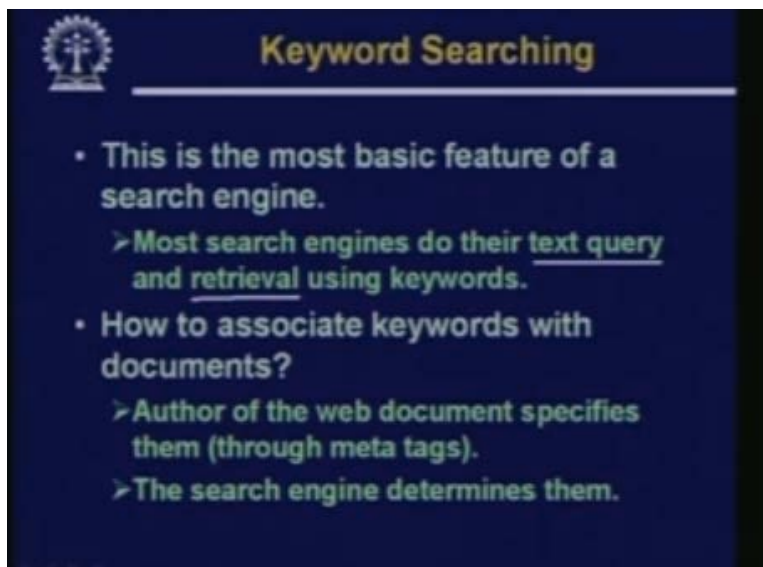
Similarly for the time required let us take an example here also. Suppose we have 100 machines, using which you are trying to build the index. Crawling 10 billion pages with 100 machines. Let us look at a realistic rate of crawling 100 pages per second. This is considered to be fast. So even at this fast rate you will be requiring 11.6 days to index all the 10 billion pages. But this is calculated with respect to the assumption that all the pages are housed or hosted on a relatively high speed connection. In fact in practice there are many pages which can take much longer for to download and hence for indexing. So the actual time taken will be much more than those 11 odd days it can take months. So what most search engines do then what they do? They crawl a small fraction of the web say about 15 to 20 percent at around the above frequency which means 11.6 days. Once every 11 days they visit those pages but the dynamic website, the news or blogs which have become so popular you crawl them at a much higher frequency because they get updated almost regularly.

(Refer Slide Time: 39:51)



So now let us look at some basic concepts regarding search engine technology.

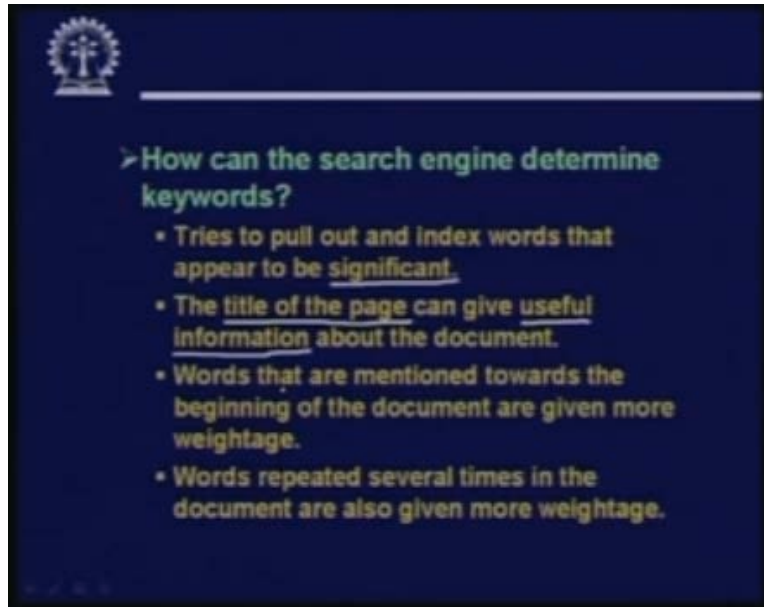
(Refer Slide Time: 39:59)



First is something called keyword searching which many of us are already familiar with. When we use these search engines, we use keywords to carry out the search. So this keyword searching is the most basic feature of a search engine. So when we want to search a specific some keywords, so their text query and retrieval by looking up the dictionary they are carried out based on keywords. The user had supplied but the problem is how do we associate keywords with the documents. These are some issues you need to understand. There are two alternatives the authors of the web documents specify them

through some special html tag called Meta. This is easier. So when the crawler visits the sites it will know that well these are the keywords. Secondly this search engine determines them automatically. Well naturally the question comes how?

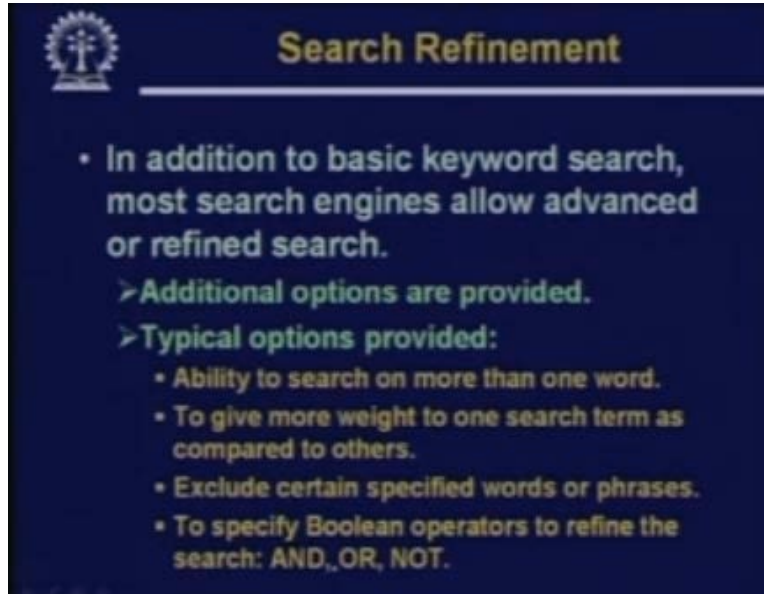
(Refer Slide Time: 41:18)



Let us see so how can the search engine determine the keywords. The search engine while crawling the pages they will try to pull out and index words from that page that appears to be significant. Now here you understand the search engine or the crawler is automated software. It automatically visits a website downloads the page and looks at the page, inspects the page. But how it will know that which word in the pages is relevant and which is not. It uses some simple criteria to judge the relevance of the page like the title of the page. Many web pages have a title which is enclosed within the title tags. They give a good idea regarding the document what the document is all about. So the title of the page can give some very useful information about the document. Words that are mentioned towards the beginning of the document are given more weightage for very natural reasons.

Suppose I have visited or means I am creating a page which talks about a particular topic. The keywords relevant to the topic are more likely to appear at the beginning. May be towards the end I have tried to discuss a few other things in connection with some peripheral matters. And the keywords that is being used there may not be directly relevant to this central theme of my page. So the words that are appearing towards the beginning should be given more weightage and also words that are repeated several times that should also be given more weightage. So these are some simple measures using which you can determine a quantitative measure or mechanism through which you can judge that which keywords should be indeed and which should not.

(Refer Slide Time: 43:28)



**Search Refinement**

- In addition to basic keyword search, most search engines allow advanced or refined search.
  - Additional options are provided.
  - Typical options provided:
    - Ability to search on more than one word.
    - To give more weight to one search term as compared to others.
    - Exclude certain specified words or phrases.
    - To specify Boolean operators to refine the search: AND, OR, NOT.

Search refinement is another issue. In addition to the basic keyword search, most keywords allow advanced or refined search. Some additional options are provided so means if you have worked with search engines you must be knowing this. So some options are the users are provided with the ability to search on more than one word. To give more weight to one search term as compared to others. Exclude certain words or phrases, you can say that these words must not appear and some combination of several keywords using Boolean operators AND, OR, NOT.

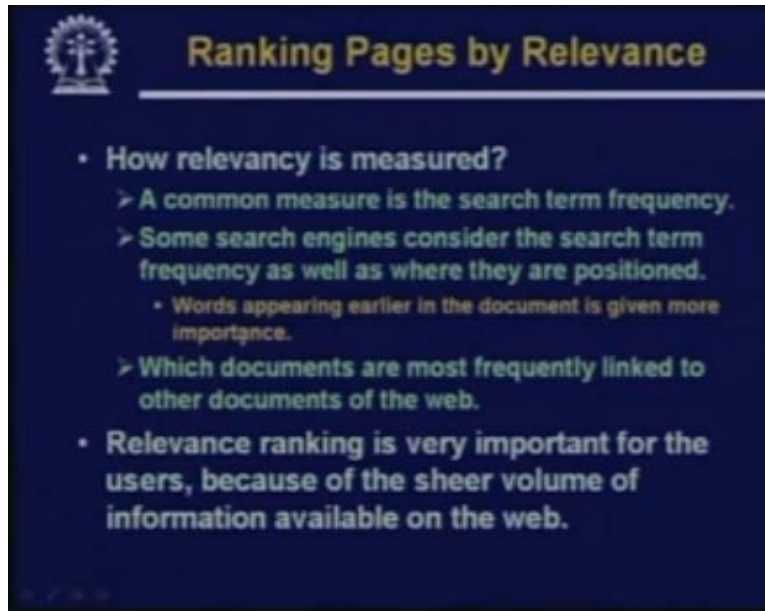
(Refer Slide Time: 44:12)



- Search for exact phrases, typically enclosed within quotation marks.
- Capitalization, which is useful when searching for some people's names.

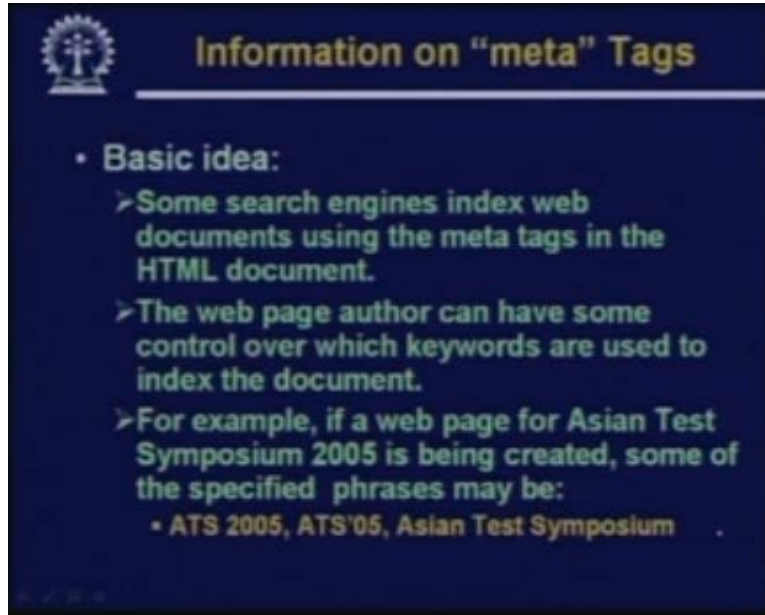
Search for exact phrases is a very popular way of searching where typically we enclose the phrase within quotation marks. Sometimes when we search for places or peoples names we use capitalization where we say that the case is sensitive. These are some typical options which are provided.

(Refer Slide Time: 44:37)



Now ranking pages by relevance as I mentioned. So the question is how the relevance can be measured. So a common measure is the search term frequency. This I have already mentioned. Some search engines consider the search term frequency as well as where they are positioned. If they appear towards the beginning of the document they are given more importance. So it should not be just a count some kind of a weighted sum is more relevant the ones appearing towards the beginning is given more weightage. And secondly the page popularity that I have mentioned earlier which documents are most frequently linked to other documents. Relevance ranking using you know, it is very important thing in the point of view of users. Because it will be very nice if the relevant pages appear towards the beginning. Due to the sheer volume of information that are typically returned when we submit a search Meta tag as I had mentioned is a mechanism using which the creator a web page can specify some keywords that are assigned or relevant to that page.

(Refer Slide Time: 45:41)



The slide features a dark blue background with a white logo in the top left corner. The title 'Information on "meta" Tags' is written in a light blue font at the top. Below the title, there is a list of bullet points in white and light blue text. The first bullet point is 'Basic idea:', followed by three sub-points starting with '>'. The third sub-point includes a list of keywords starting with '•'.

- **Basic idea:**
  - > Some search engines index web documents using the meta tags in the HTML document.
  - > The web page author can have some control over which keywords are used to index the document.
  - > For example, if a web page for Asian Test Symposium 2005 is being created, some of the specified phrases may be:
    - ATS 2005, ATS'05, Asian Test Symposium

The basic idea is simple. There is a Meta tag in HTML. So the web authors can specify some keywords using Meta tags. There by can have some control over the keywords based on which the document will be indexed. As an example suppose we have created a web page for a symposium Asian Test Symposium 2005. Some of the typical keywords that the web author may feel that will appear in many queries will be ATS 2005, ATS 05 or the full phrase Asian Test Symposium 2005. But if you do this automatically may be you cannot capture this kind of very relevant keywords. So if the author himself or herself specifies the keywords it will be much more relevant.



(Refer Slide Time: 46:57)



**How to Specify Meta Tags?**

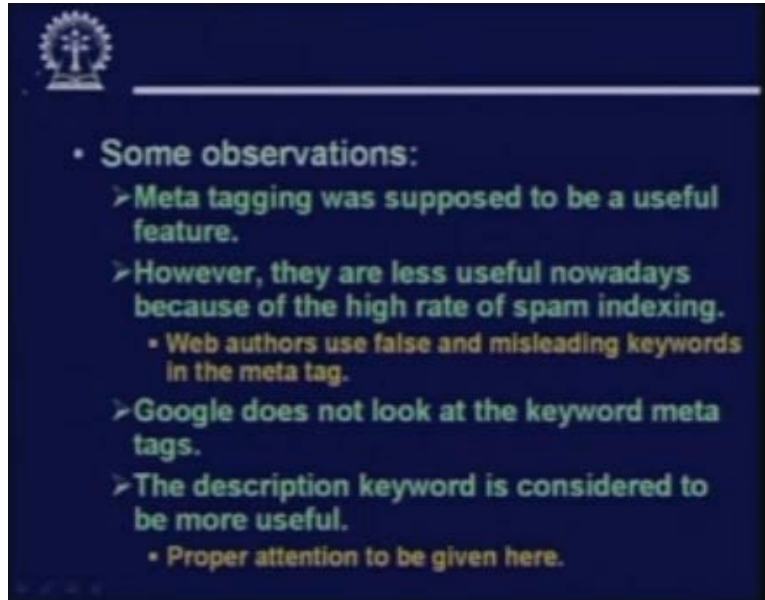
```
<HEAD>
  <TITLE>Asian Test Symposium 2005</TITLE>
  <META name="description" content="web site of
the symposium">
  <META name="keywords" content="ATS 2005,
ATS'05, Asian Test Symposium">
</HEAD>
```

- To prevent a search engine from indexing the web page, the following line can be added:

```
<META name="ROBOTS" content="NOINDEX">
```

Some examples here how is Meta is used. This is an html file. So within the head after title this Meta tag is used. The first one where the name is description the name attributes has a value description. This simply gives some information about the page. This is optional and if the value of the name is keywords. This will mean that the content attribute will contain some values separated by commas within quotes which will be keywords. So this is how you can specify keywords? There is also a way to prevent a search engine from indexing your page. Suppose you do not want your page to be indexed then we include this Meta command, Meta name equal to ROBOTS content equal to NOINDEX. So by using this name content pairs in different ways. You can use this Meta tag to provide some very relevant information.

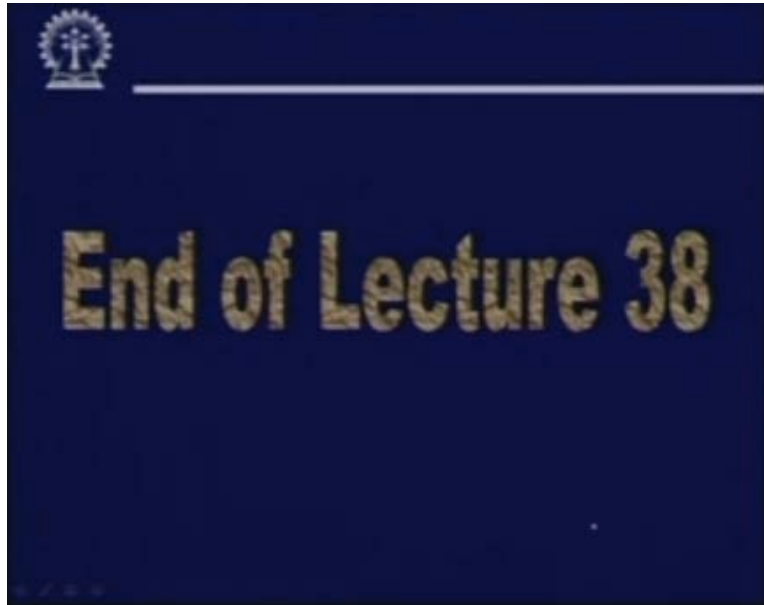
(Refer Slide Time: 48:03)



Now some observations are very relevant here. Well these Meta tags were created or they were defined as part of the HTML language because people thought that they will be very useful. If a page author submits the keywords himself or herself keywords will be much more relevant much more content, you can say content based. But however there is some a hindrance which is defeating the basic purpose. There is something called spam indexing where some of the web page authors they arbitrarily insert some web search keywords which are very popular in their meta tags. So that their pages will be indexed or referred more often.

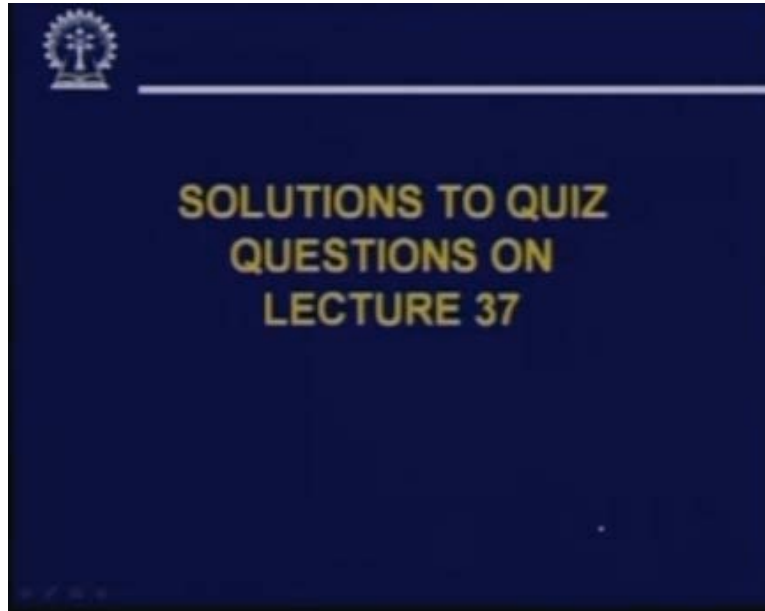
They are basically false and misleading keywords. This is the reason why Google does not look at all at the keyword Meta tags. So even if you have given some keyword Meta tags in your document Google will try to do the indexing automatically not look at the keywords. But however the description attribute is considered to be much more useful by Google because in description typically description of the page is given. So the web page author should give proper attention here. So that the description is given in a proper way which Google can index in meaningful way.

(Refer Slide Time: 49:55)



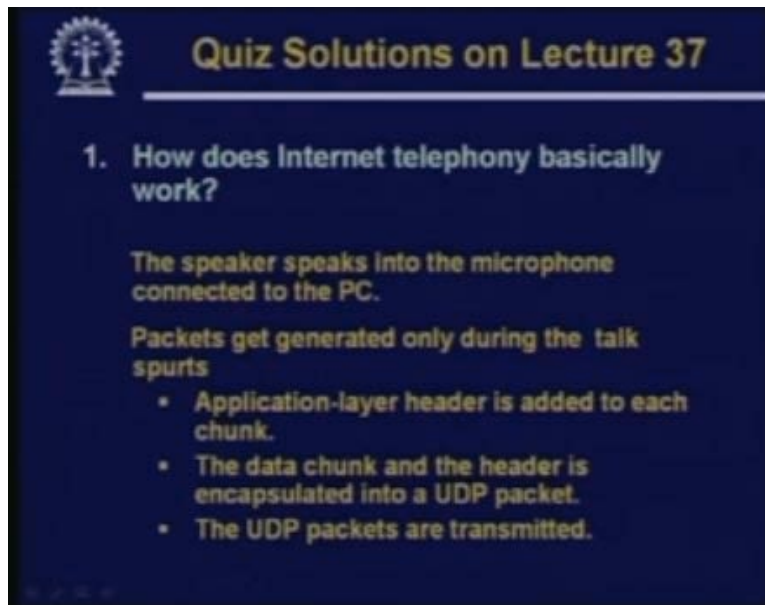
So with this we come to the end of our lecture number 38. Now in this lecture we basically looked at some of the aspects of the web search engine. We have looked at the basic components of a typical crawler based engine like the crawler the index and the actual search engine software. We had looked at some of the issues that are being faced by search engine. How the pages are ranked by relevance? How the popularity of a site is judged and so on. So in our next lecture we will be continuing with this discussion. We shall be talking about mainly how web crawler works. Web crawler is as I said one of the representative crawler based search engines that many people use for tutoring purposes. Because that was possibly the first effort and very interesting and well documented.

(Refer Slide Time: 51:00)



So now let us look at the solutions to the quiz questions that were posted in our last lecture. Lecture 37.

(Refer Slide Time: 51:12)

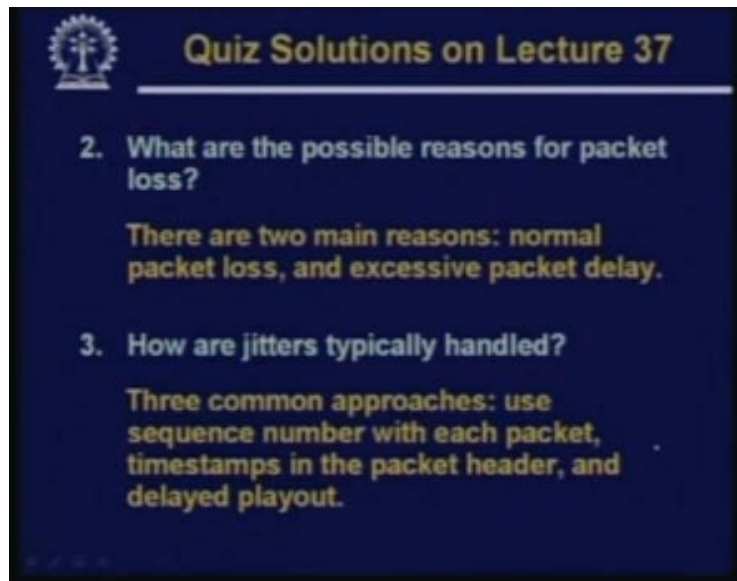


First question was how does internet telephony basically work?

Now in internet telephony the speaker as I mentioned last class speaks into the microphone connected to the PC and the voice is received and played back on the speaker. The packets get generated when the speaker speaks in the microphone. The

voice packets will get generated but only during the talk spurts the speaker is not speaking is silent the packets do not get generated. So to each junk of such voice data application -layer headers are added. The data junk and the header together are encapsulated into an UDP packet and the UDP packets as a whole are transmitted. This is how very roughly internet telephony works.

(Refer Slide Time: 52:18)



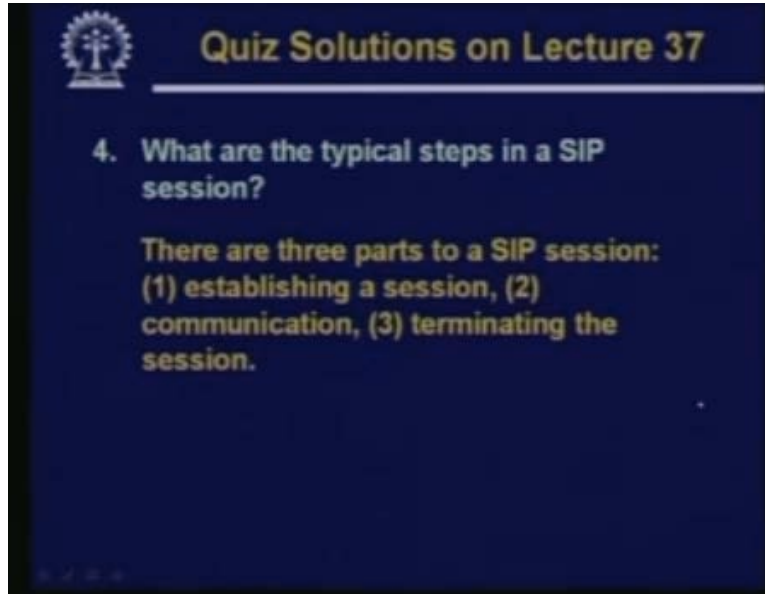
What are the possible reasons of packet loss?

There are two main reasons of packet loss. One is the normal packet loss which arises out of the bit error rate in the network. Some noisy channels noisy connections some problem with the host. So the normal unreliability of a network which drops a few of the packet that same reason may be here. This is called the normal packet loss. But sometimes due to the unpredictable packet delay of the IP protocol there can be excessively delayed packets which as I mentioned are typically dropped by the receiver for voice over IP applications.

How are jitters typically handled?

Jitters can be handled using three common approaches. Some sequence number can be assigned to each packet so that the receiver can know whether some packet is arriving out of order or not. If it is an out of order it is better to drop it. Timestamps work in a similar way. Instead of sequence number you can put timestamps in the packet header and delayed payout. You can basically delay the playback process till sufficient number of packets have arrived and then you can start it so that jitter effects will be minimum.

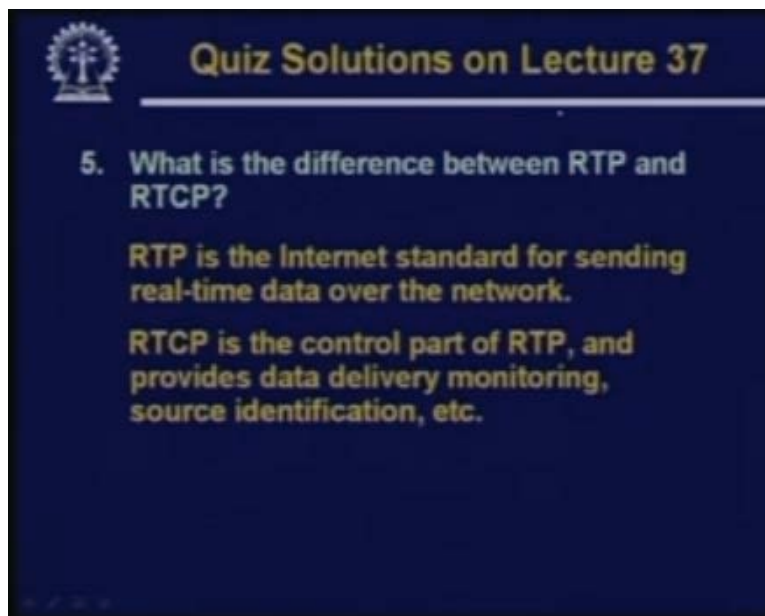
(Refer Slide Time: 54:01)



What are the typical steps in a SIP session?

Now in the SIP session we had talked about the first session is to establish the session. There is a protocol out of here. Second is the actual communication using the voice packets and thirdly to terminate the session.

(Refer Slide Time: 54:27)

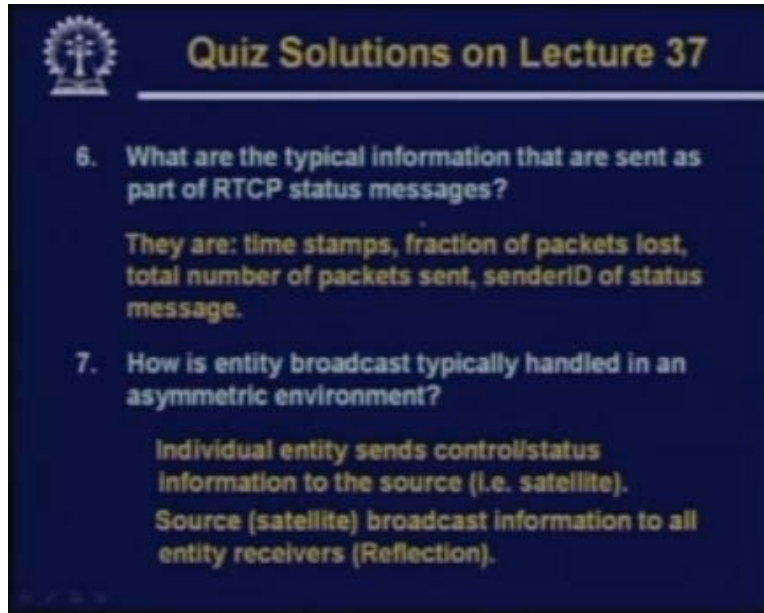


What is the difference between RTP and RTCP?

Now RTP is the internet standard for sending real time data over the network and we had mentioned if you recall RTCP is a companion protocol it is not independent. It is a

protocol which is which can be regarded as the control part of RTP. It provides basically data delivery monitoring source identification and these kinds of control features which are necessary for the RTP protocol to communicate.

(Refer Slide Time: 55:09)



The image shows a slide titled "Quiz Solutions on Lecture 37" with a logo in the top left corner. The slide contains two quiz questions and their solutions. Question 6 asks for typical information in RTCP status messages, and the solution lists time stamps, fraction of packets lost, total number of packets sent, and senderID. Question 7 asks how entity broadcast is handled in an asymmetric environment, and the solution describes individual entity sending control/status information to the source (satellite) and the source broadcasting information to all entity receivers (Reflection).

**Quiz Solutions on Lecture 37**

6. What are the typical information that are sent as part of RTCP status messages?

They are: time stamps, fraction of packets lost, total number of packets sent, senderID of status message.

7. How is entity broadcast typically handled in an asymmetric environment?

Individual entity sends control/status information to the source (i.e. satellite).  
Source (satellite) broadcast information to all entity receivers (Reflection).

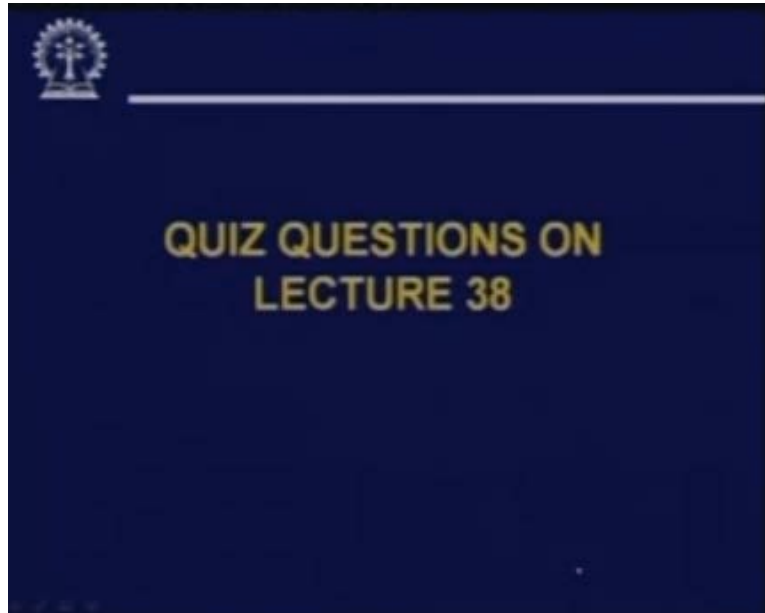
What are the typical information that are sent as part of RTCP status messages?

Well RTCP status messages convey several kinds of information. Some of the typical information's are time stamps, fraction of packets lost, total number of packets being sent and sender ID of the status message. It tells who has sent the status message. So these are the kind of information that RTCP status message also contains in addition to the normal control information.

How in entity broadcast typically handled in an asymmetric environment?

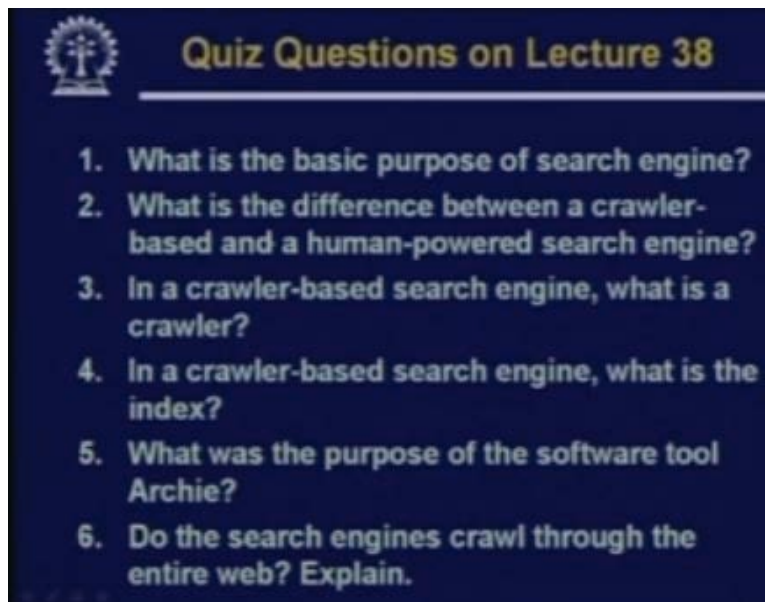
Entity broadcast asymmetric environment we had mentioned here you have taken example of the satellite communication. Here the individual entity can send control or status information to the source which is the satellite and the satellite will be broadcasting information to all the receivers. This in reflection mode where the entity who wants to broadcast is not broadcasting directly. But it is sending it to somebody else reflecting from there and it is being broadcast to all the other entities in the range.

(Refer Slide Time: 56:40)



So not let us look at the questions from today's class.

(Refer Slide Time: 56:47)



What is the basic purpose of search engine?

What is the difference between a crawler based and a human powered search engine?

This you have to illustrate with examples.

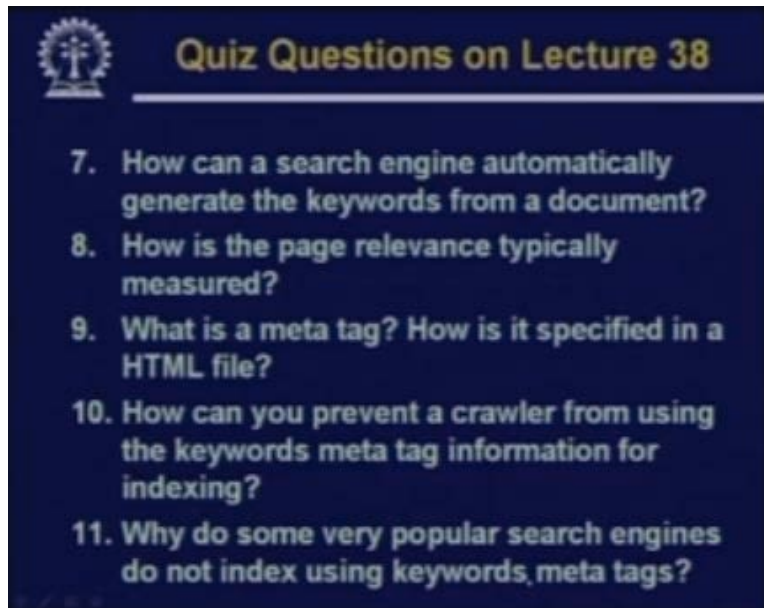
In a crawler based search engine what is a crawler? What is an index?

What was the purpose of the software tool Archie?



Do the search engines crawl through the entire web? Explain.

(Refer Slide Time: 57:25)



How can a search engine automatically generate the keywords from a document?

How is the page relevance typically measured?

What is a Meta tag?

How is it specified in a HTML file?

How can you prevent a crawler from using the keywords Meta tag information for indexing?

Why do some very popular search engines do not use indexing using keywords Meta tags?

So with this we come to the end of today's lecture. As I said, in next lecture we shall be continuing with this discussion. We have been talking into the detail implementation of the web crawler engine. Till then good bye.