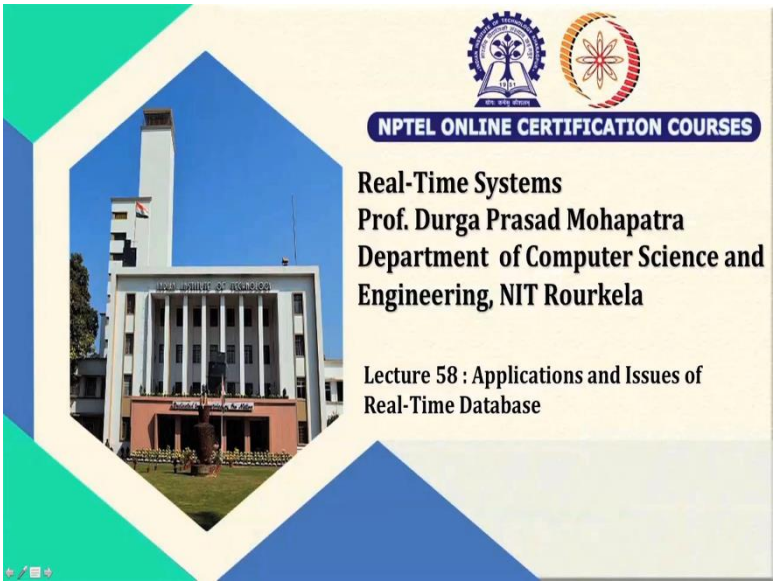
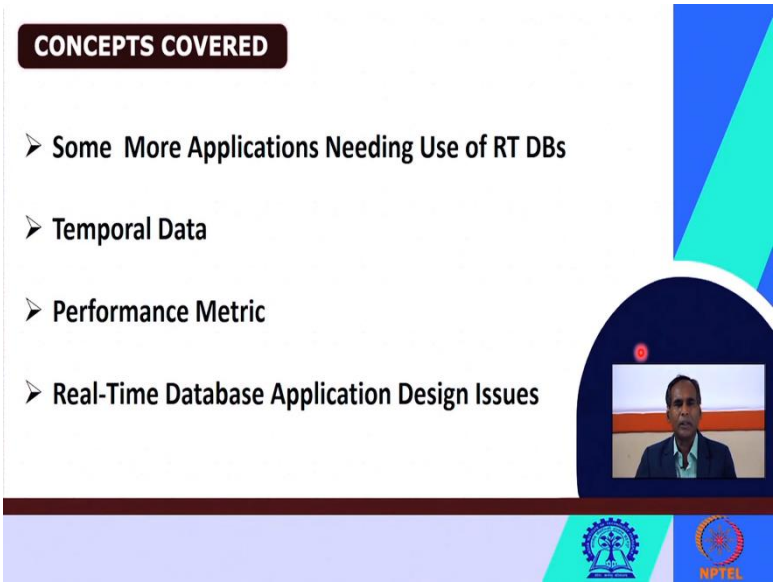
Real Time Systems
Professor Durga Prasad Mohapatra
Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Lecture 58
Application and Issues of Real-Time Database

(Refer Slide Time: 00:13)



Good afternoon to all of you. Now let us start from the point where I have left in the last class. We are discussing about the applications of the what real-time databases or the applications where, real examples where real-time databases can be used.

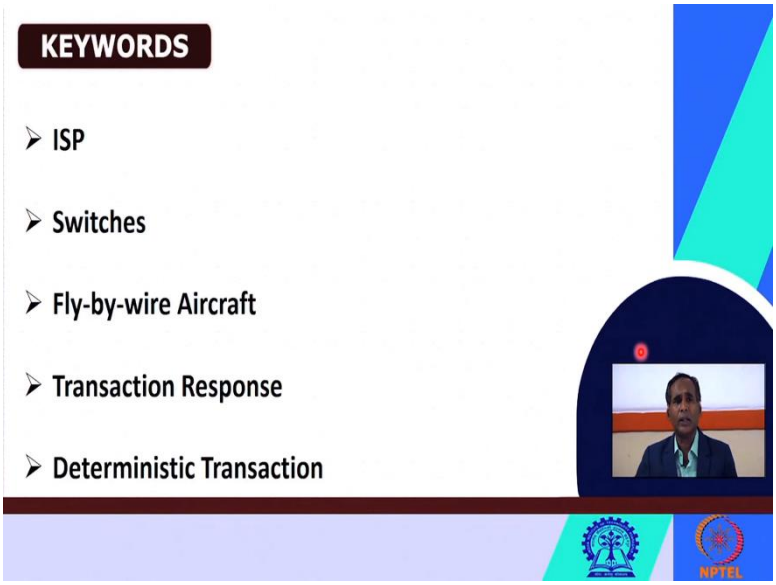
(Refer Slide Time: 00:43)



So, let us today discuss some of the more applications where real time databases can be used and then some of the issues of the real time database.

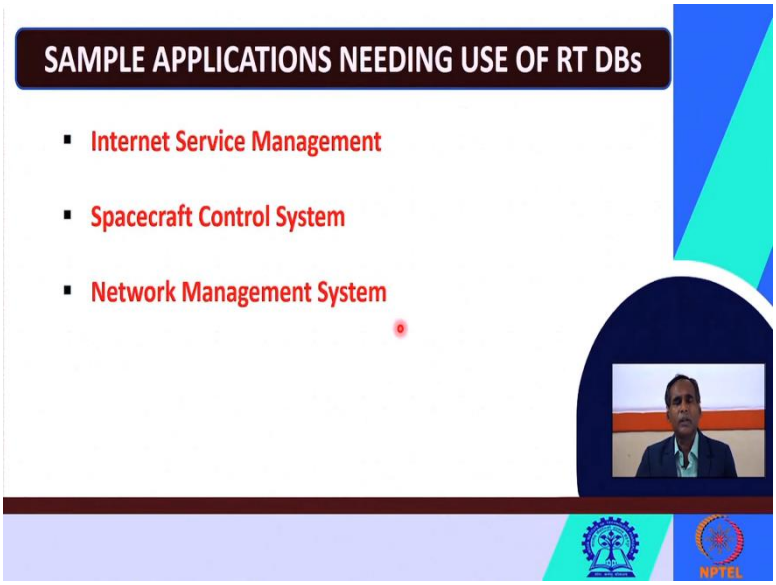
So, first we will discuss some more applications needing use of real-time databases. We will see what do you mean by temporal data, we will see what are the performance metrics for real-time databases and we will see some of the design issues for real-time database applications.

(Refer Slide Time: 00:58)



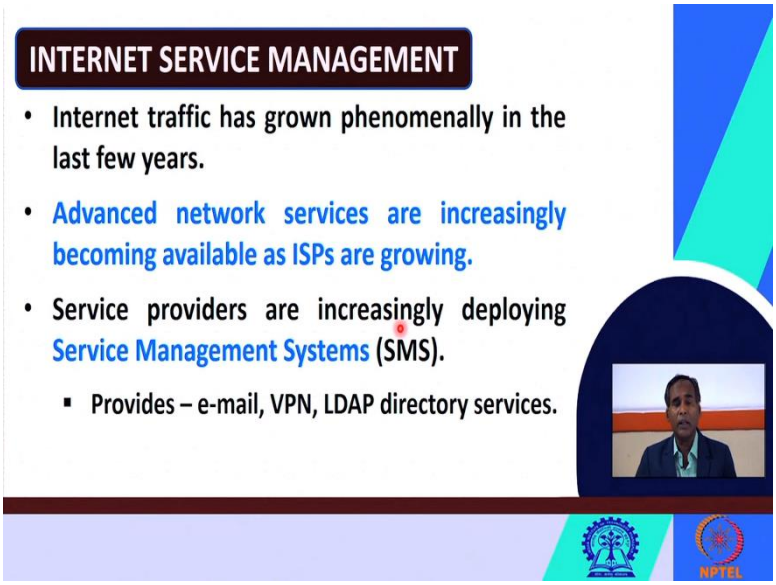
We first see some of the keyword you will use are ISP, switches, fly-by-wire aircraft, the transaction response and deterministic transaction etc.

(Refer Slide Time: 01:09)



So, last class we have discussed about the process control system was one of the example where real-time database is used. So, today we will discuss three more applications like Internet Service Management, Space Control System, Network Management System where there is a need of using real-time databases or these applications, they use real-time databases.

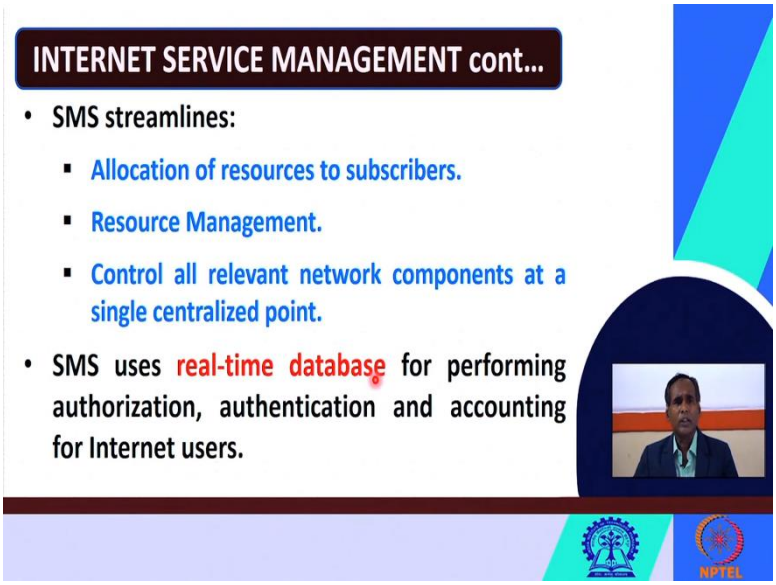
(Refer Slide Time: 01:30)

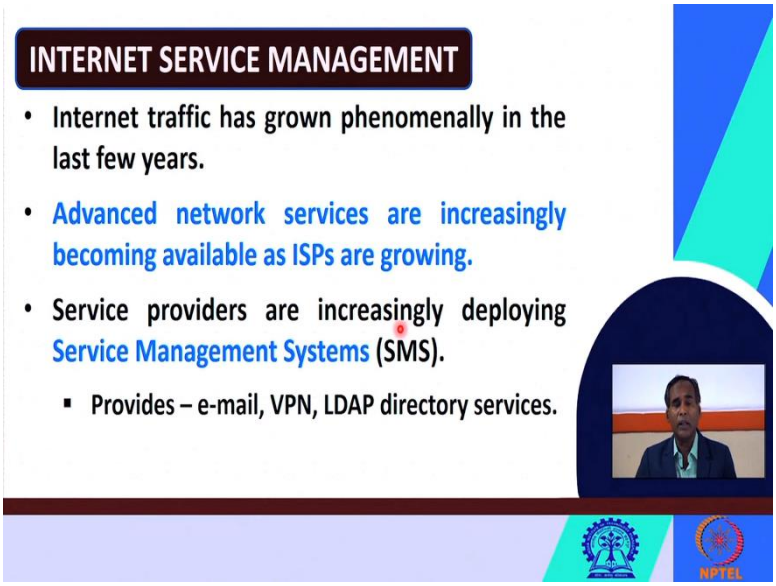


We will first see about the Internet Service Management or ISM. So, you know that the internet traffic nowadays has grown phenomenally in the last few years. In the last few years, the internet traffic has grown phenomenally. So, the advanced network services they are increasingly becoming available as ISP are ongoing, as the internet service providers are growing the advanced network services, they are increasingly becoming available.

The service providers they are also increasingly deploying the service management system. So, these internet service providers, they are increasingly deploying the service management system or we should call as a SMS. So, these service management systems, they provide the services such as the email, VPN, LDAP directory services etc. So, these services are provided by this service management system.

(Refer Slide Time: 02:24)





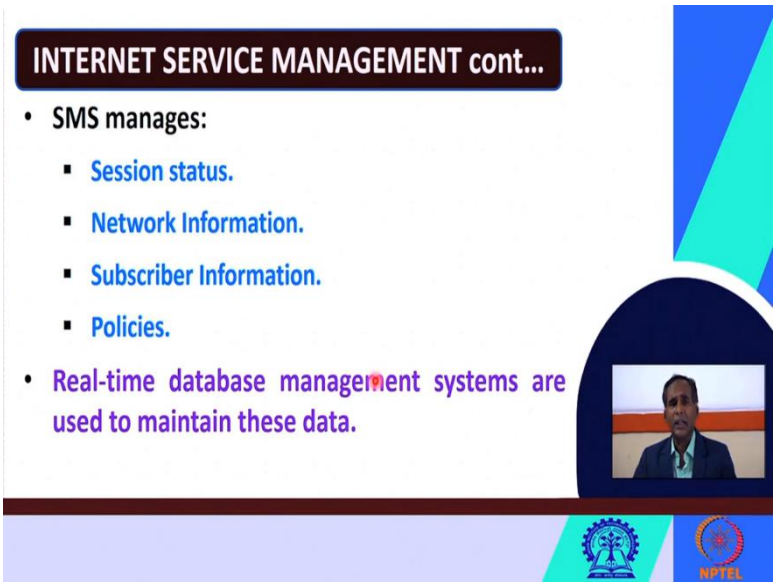
So, let us see what this service manual system, it does more. The service manual system, it is streamlined, it streamlines allocation of the resources to the subscribers. How? The different resources can be allocated to these various subscribers that is performed by this service management system. It also streamlines the resource management, how the resources can be managed efficiently. The service management system also controls all the relevant network components at a single centralized point.

So, this service management system, as I have already told you, or this ISP or internet service for internet service management, the service providers they are using service management systems. This service manual system, it uses a real-time database because here, the database is

also associated with some timings. So, SMS uses or this service management system, it uses real time database, why do it use real-time database?

The service management system uses real time database for performing authorization, for performing authentication and for accounting for the different internet users. So, in order to perform authorization, authentication and accounting for internet users, this service management system it uses the real time database. So, nowadays many of the Internet Service Providers they use this database management system in turn SMS it uses the real-time database for performing the authentication, authorization and accounting for the internet users.

(Refer Slide Time: 03:56)



So, this service management system it manages the session status, whether the session is it is active or it is idle or it has been completed or terminated. So, SMS manages the session status, it also manages the information about the network. It manages the network information; it also manages the information about the different subscribers. It also manages the different policies. So, in order to manage all those things, in order to maintain all those data, this SMS uses real-time database management system.

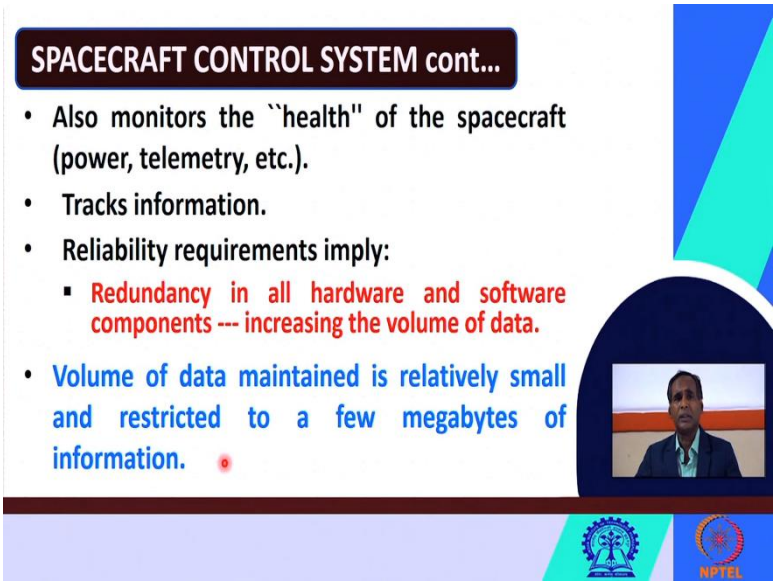
So, in order to manage in order to maintain all this data, the real time database management systems, they are used or SMS uses real-time database management systems in order to maintain this data. This is how we have seen in case of internet service management how real-time database management system is used.

(Refer Slide Time: 04:49)

We will see next the other application where also real-time database is used that is spacecraft control system. So, in a spacecraft control system, a control system is used which is responsible for the successful overall operation of the spacecraft. So, this control system it is also responsible for receiving the command-and-control information from the ground computer and this control system it also maintains a contact with the ground control using what, antenna, receivers and the transmitters.

So, this control system it also maintains contact with what, with the ground control using what different antennas, different receivers and different transmitters. This control system, it also monitors several parameters through sensors mounted on and within this spacecraft. So, you know in this spacecraft there are so many sensors and actuators are connected. This control system it monitors the different parameters through the sensors which are mounted on and within the spacecraft.

(Refer Slide Time: 05:59)



So, it also monitors the health of the spacecraft. So, this controller it also monitors the health of the spacecraft that means what is the power consumption, telemetry, etc. So, these things are monitored by this control system. The control or the controller or the control system, it also tracks it also keeps track of different information. Now, for this spacecraft control system, there is a heavy reliability requirement. It must be highly reliable otherwise if a minor mistake will happen that whole spacecraft system it will just collapse.

So, there is a heavy requirement of reliability and these reliability requirements, they imply that how can ensure what better reliability. In order to ensure a better reliability in spacecraft control system, you can use a redundant number of hardware and software components so that if your one hardware or software component fails the rest of the hardware and software components, they can carry forward the job. They will perform, the other operations the system will not collapse. So, in order to maintain better reliability, what is required, you should use redundant number of hardware and software components.

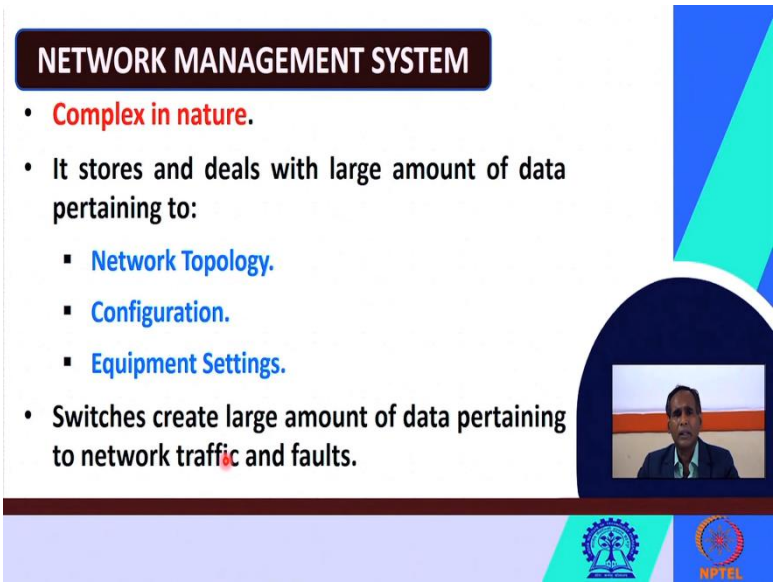
And if you will use, please see if we will use a redundant number of hardware components and software components, obviously it will increase the volume of the data. So, how then if the volume of data is more, how will we manage this? So, you have to manage by using a real-time database.

So, volume of data can be maintained by using a real-time database here. So, the volume data maintained in case of the space control, spacecraft control system of course, the volume of data maintain is little bit is small and it is restricted to a few megabytes of information in comparison to some other applications like this internet service management system, we have seen in

comparison to that here the volume of data maintained is of course relatively small and it is restricted just to or it is restricted only to a few megabytes of information.

You please compare with this information or this the last example we have discussed, internet service management if you take that one and compare this spacecraft control system, here the volume of data maintained is relatively small and less in comparison to this internet service management and this volume of data is restricted here to only a few megabits of information.

(Refer Slide Time: 08:24)



Next let us see another example where real-time database is required. So, that example is network management system. So, normally you know network management system is very complex in nature, you see a large data centres, large computer centres, so here a network management system is very much complex in nature and the network management system what does it do, it stores and it deals with a large amount of data.

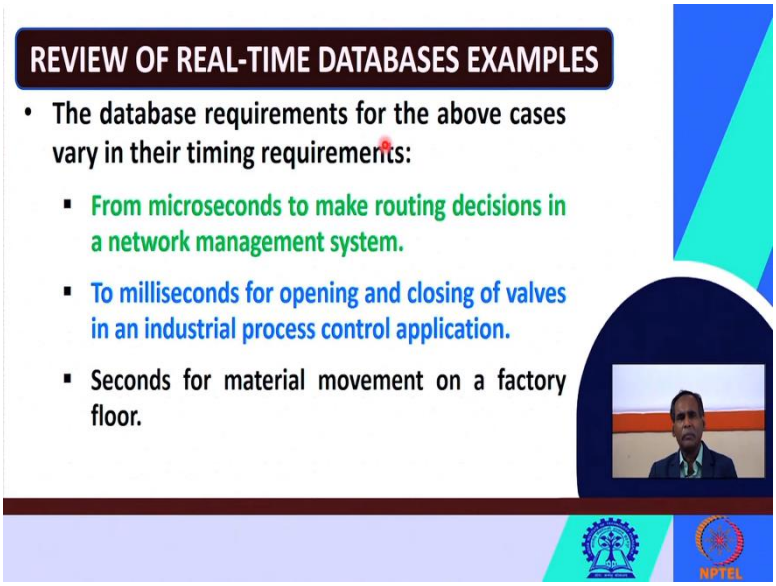
The network management system, it stores and deals with a large amount of data which are pertaining to the topology of the network, configuration, the equipment settings and so many. So, this network management system, it keeps track of a very large amount of data pertaining to network topology, configuration, equipment settings etc. And you know in a large data centres or large computer centres, there are so many switches are used. We have already discussed what a switch is and the switches, they create large amount of data pertaining to network traffic and faults.

So, in larger computer centres or data centres, these switches they create they generate large amount of data pertaining to the network traffic and the faults etc. So, since a large amount of data they are generated, they have to be managed properly.

They have to maintain how? So, in order to manage in order to maintain this heavy amount of data, this large amount of data you have to use a real-time database system because here also, the data base it is associated with some timing, what properties. So, here the data in order to maintain in order to manage the data generated that is in order to maintain the large amount of data generated by the switches, you have to use real-time databases.

So, we have seen another example of the application like network management system where you also require the use of real-time databases. Similarly, many other applications are there which require the use of real-time databases. You can see from the books or from internet sources.

(Refer Slide Time: 10:24)



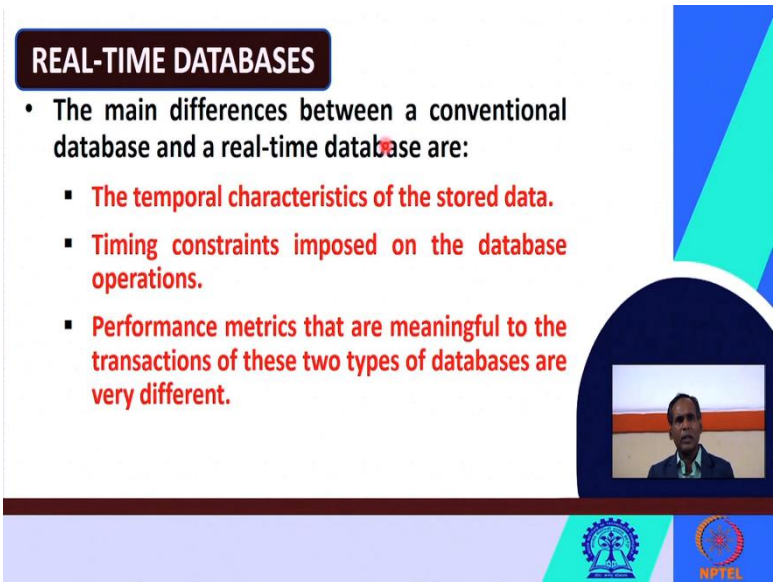
Now, we will see, we have seen that actually many examples where real time databases are required. Now let us see, I get this summary. Let us have a review of those examples that we have seen where real time databases are used. So, the database requirements for the above examples that we have discussed, they may vary. Vary in what? They may vary in their timing requirements; the timing requirements may vary from internet service management to this network management system or to the spacecraft system or it may vary from this internet service management to the process control application.

So, you can see that the timing requirements may vary from some microseconds to millisecond or even up to the order of seconds. Where it is a microsecond, say if in case of network management system, you know the routing decisions, they have to be done within some microseconds. Whereas if you will see the industrial process control application, the opening and closing of the valves, this will take us within some milliseconds and if you go for a factory floor maybe still melting factory or aluminium making factory, they are the material moment it will be finished it may be completed within some seconds.

So, that is why I am saying here, the database requirements in the different applications they may vary in timing requirements. In some cases, the timing requirement might be some microseconds, maybe example is the routing decisions in a network management system. In some cases, these timing requirements may be of the order of milliseconds for example, in a process in an industrial process control application, the timing requirement for opening and closing of valves is of the order of milliseconds.

While it is a little bit flexible, it takes more time like if in a factory floor, the movement of material may take time which is of the order of some seconds. So that is why the timing requirements for different, what real-time applications or real-time data bases may vary from microseconds to seconds. So, some of the operations may take microseconds time whereas some of the operations may take milliseconds time whereas some of the operations may take time which is of the order of some seconds.

(Refer Slide Time: 12:41)



Now, we have already seen about what is the difference between conventional database and real-time database. We have seen some of the examples where real-time databases are required.

Now let us go into deeper of the real-time databases. So first, let us see how do the real time databases differ from the conventional databases.

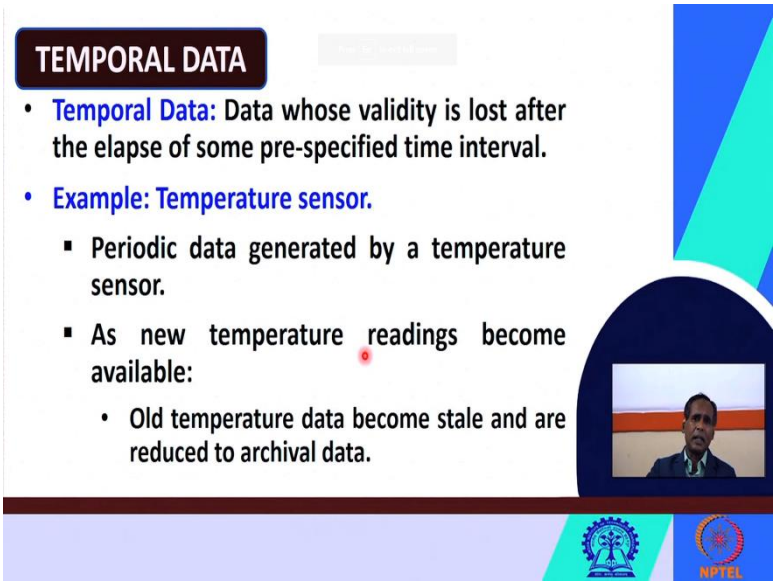
So, normally, the what main differences between a conventional database and the real-time database are as follows. Main three important differences are there. Number one is the temporal characteristics of the stored data. The data that is stored, so, what is the temporal characteristic? So, that is a big difference between conventional database and real-time database, we will see shortly on these temporal characteristics.

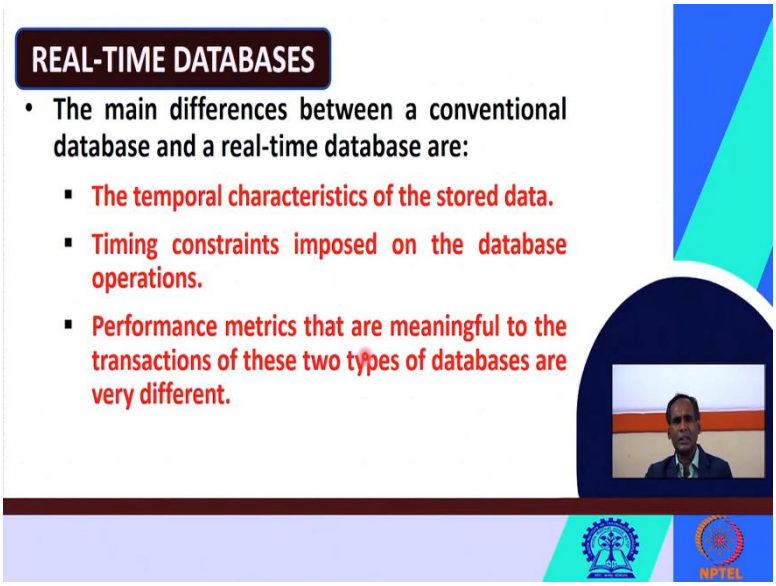
Then what are the timing constraints those are imposed on the database operations. So normally, in case of a traditional database, those timings are little bit flexible, but in case of what real-time database those timing constraints are very strict, you have to strictly follow, you have to strictly adhere to those timing constraints or the transactions they will miss their deadlines and one more difference is in the performance metrics.

What performance metrics you are using for the conventional databases, they may not be meaningful they may not be suitable for the real-time databases. So, we have to discuss some new performance metrics which will be more suitable for this what real-time databases. So, performance metrics that are meaningful to the transactions of what your conventional databases may not be suitable for this real-time database.

So, those are metrics they are different. So, the performance metrics which are meaningful to the transactions of these two types of databases are very different, performance metrics for use for conventional database is not suitable for the real-time databases. We will see now three these three important differences in much detail.

(Refer Slide time: 14:33)





Let us see first the temporal data. First one is temporal data characteristics or temporal characteristics. So, what do you mean by temporal data? Temporal data is a data, whose validity is lost after the elapse of some pre-specified time interval. So, here you have to pre-specify you have to pre-determine some time interval. And if that time interval that pre-specified time interval is elapsed, then the validity of the data is lost. This is called as temporal data.

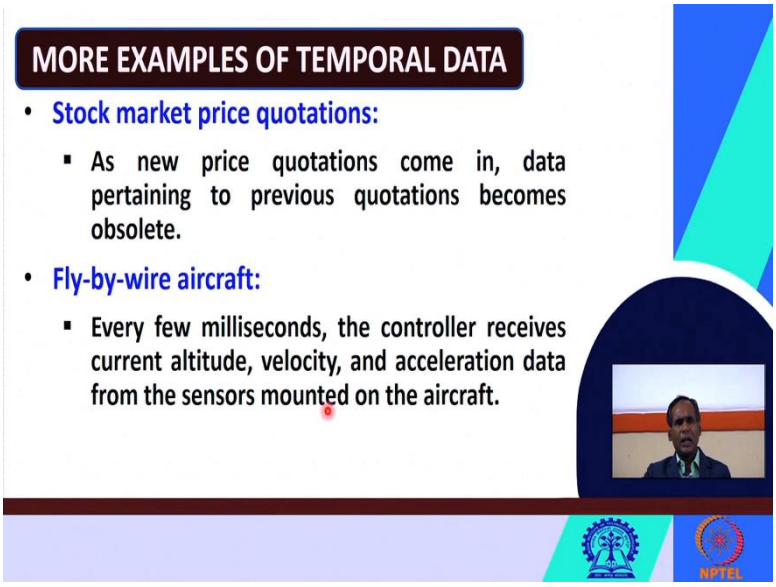
So, temporal data is some data whose validity is completely lost after the elapse some pre-specified or some pre-determined or some fixed time interval. Let us take an example of temporal data. Let us take the case of this temperature sensor in a blast furnace or in a chemical chamber.

You know the periodic data normally they are generated by a temperature sensor in a blast furnace or a chemical chamber the data those are generated by a temperature sensor normally they are periodic data. They are periodic in nature. The periodic data is generated by a temperature sensor, this is an example of what a temporal data why because suppose every millisecond the current temperature is being read.

So, as the new temperature readings become available, then obviously, what happens the old temperature data becomes stale and are reduced to archival data. Suppose in a blast furnace, at a T equals 0.1 millisecond or just a T is equal to 1 millisecond, you have seen that temperature = 100 degrees centigrade and then at T = 2 milliseconds, temperature is suppose now 105 degrees centigrade, then what will happen? So, when we got the temperature for T = 2 milliseconds obviously, what is the temperature at T = 1 that is invalid, its validity is lost because that data will not use because that is already that becomes obsolete.

So, this is a very good example of temporal data. The periodic data generated by a temperature sensor in a blast furnace or in a chemical chamber is a very good example of temporal data. Here are the new temperature readings become available, the old temporal data becomes stale it becomes obsolete and they are induced to do what archival data. This is something about this temporal data, we will see some more examples.

(Refer Slide Time: 16:50)



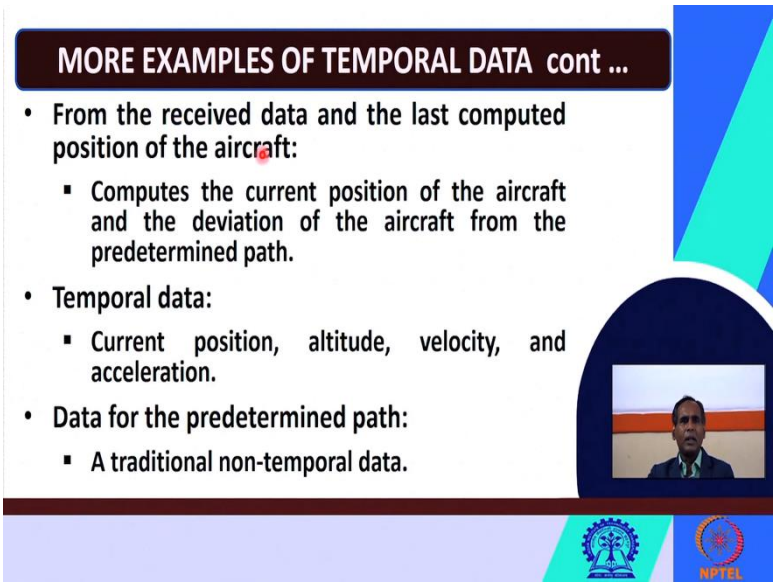
You know about stock market price quotations, here what happens as the new price quotations come in what will happen, the data pertaining to the previous quotations automatically becomes obsolete. So, right now, the stock market price it changes very frequently. So, if time is equal

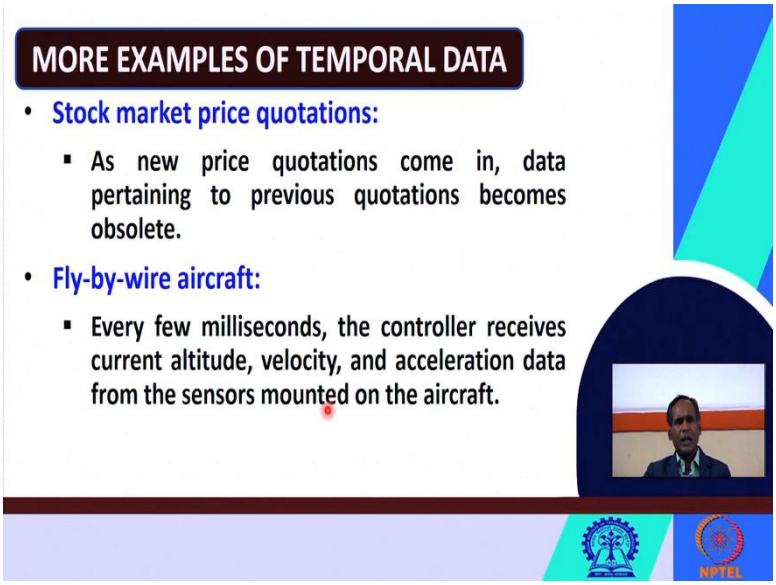
to T1, you have got what is the price quotation and at time is equal to T2, then the new price quotation comes.

Obviously, what is the price quotation at time T is equal time is equal to T1 that becomes obsolete. So, this is a very good example of temporal data. So, in stock market to price quotations as the new price quotations come in, then the data pertaining to the previous quotation they automatically become obsolete. Another example is fly-by-wire aircraft.

You know when an aircraft is flying, in every few milliseconds the controller of the fly-by-wire aircraft, it receives the current altitude, current velocity and current acceleration data from where from the sensors which are mounted on the aircraft. So, the sensors they read the current altitude, current velocity and current acceleration and within every few milliseconds, the sensors read those that are related to the current altitude, velocity and acceleration.

(Refer Slide Time: 18:11)





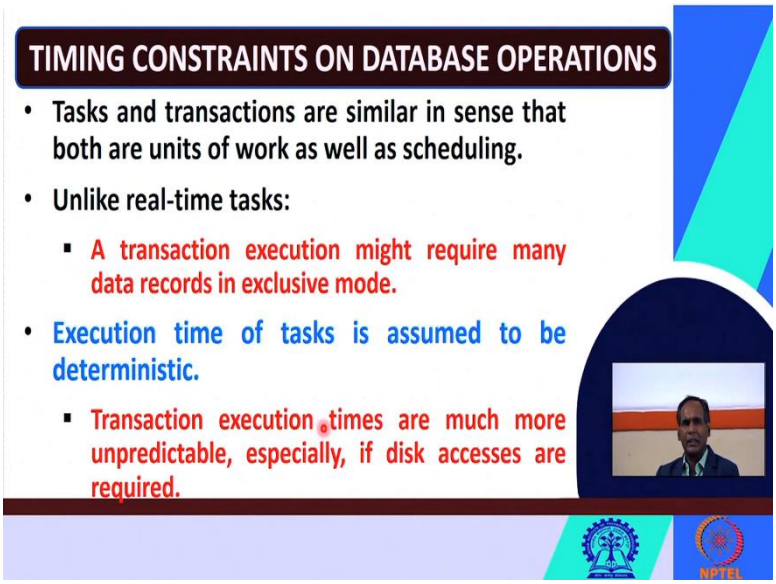
Then what happens? From the received data that means from the received data on the altitude, velocity and acceleration what happens, from the received data and the last computed position of the aircraft, the controller here it computes the current position of the aircraft and the deviation of the aircraft from the predetermined path. So, here you see what I am saying an aircraft is flying, in every few milliseconds the sensors, they read the current altitude, velocity and acceleration those values.

From the received data and the last computed position of the aircraft, the controller it computes the current position of the aircraft and also it measures the deviation of the aircraft from the pre-determined path. So, here you see as it gets the recent what data that means, suppose that the T is equal to what a 1, it gets the data regarding the velocity, acceleration, altitude etc. or at T is equal to 1, if it gets the current value of altitude, velocity, acceleration then automatically the value which it has reached, what got that T is equal to 0 that becomes obsolete.

Similarly, when these sensors read the value at time is equal to T2 regarding the current altitude, velocity and acceleration these data then obviously, the data collected at T is equal time is equal to T1, the data collected regarding the current altitude, values and acceleration, at time is equal T1 they become obsolete when these data are read at time is equal to T2. Hence, these data are very good examples of temporal data. So, what are the temporal data here in case of the fly-by-wire aircraft?

So, the data such as current position, altitude, velocity and acceleration, they become the temporal data whose value it is lost when any new value comes in. But here, the data for the pre-determined path you may treat it as a traditional non-temporal data. So, the data for the pre-determined path, it may be considered as a traditional non-temporal data.

(Refer Slide Time: 20:18)



Then we have seen the first constraint that the temporal data these the first difference, temporal characteristics of the data then you will see the second difference between traditional and real-time data that is the timing constraints on the database operations. Before going to the timing constraints on database operations, let us first see the differences between tasks and transactions. So, what happens the tasks and the transactions normally they are similar. How they are similar?

Normally, they are similar in the sense that both are units of work as well scheduling. If you take tasks or if you take transaction, they are similar because they are both are units of work as well as scheduling. Unlike the real time tasks, a transaction what it may does, a transaction execution might require many data records in exclusive mode. So, this does not happen in real-time tasks but in case of transactions, a transaction execution might require many data records in exclusive mode.

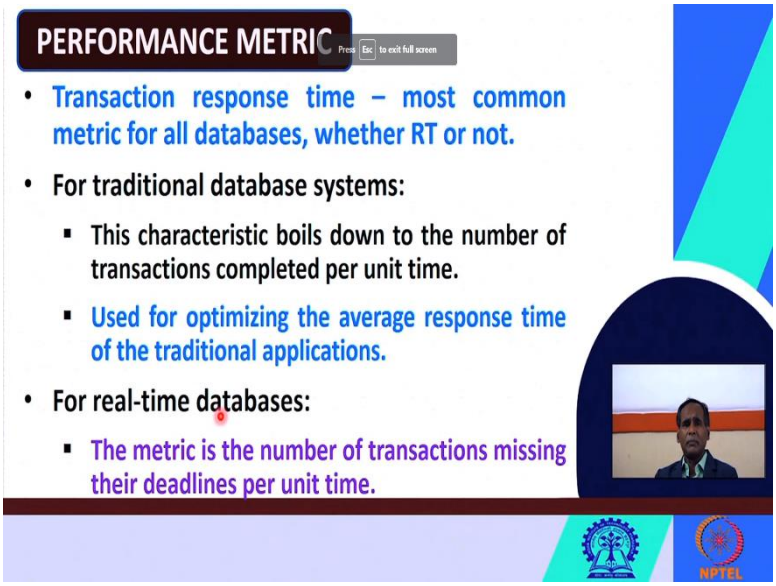
The execution time of tasks is assumed to be deterministic. So, normally when you are talking about the tasks, the execution time of the tasks, it is assumed to be deterministic whereas the execution time of transactions or the transaction execution times they are not deterministic, they are much more unpredictable. The execution time of tasks is assumed to be deterministic whereas the transaction execution times are much more unpredictable especially, if disk accesses are required. So, if you are accessing some, if you require some disk accesses then this transaction times, they are much more unpredictable.

You cannot say when that transaction will be completed. So, this is a little bit difference between the tasks and this what the transactions So, what I am saying here, there is some timing

constraint associated with on these tasks. A task must have to, it is deterministic, execution time of the tasks is assumed to be deterministic that means a task support must be completed within this much of time. That timing constraint is associated.

So, the execution time of a task is assumed to be deterministic. You have to give special emphasis you have to remember that a task has to be completed within this much of time and that time is normally deterministic whereas in transactions, the transaction execution times they are much more unpredictable particularly, when you are using when you are requiring disks accesses then obviously, the transaction execution times are unpredictable. The execution time of tasks are assumed to be deterministic.

(Refer Slide Time: 22:52)



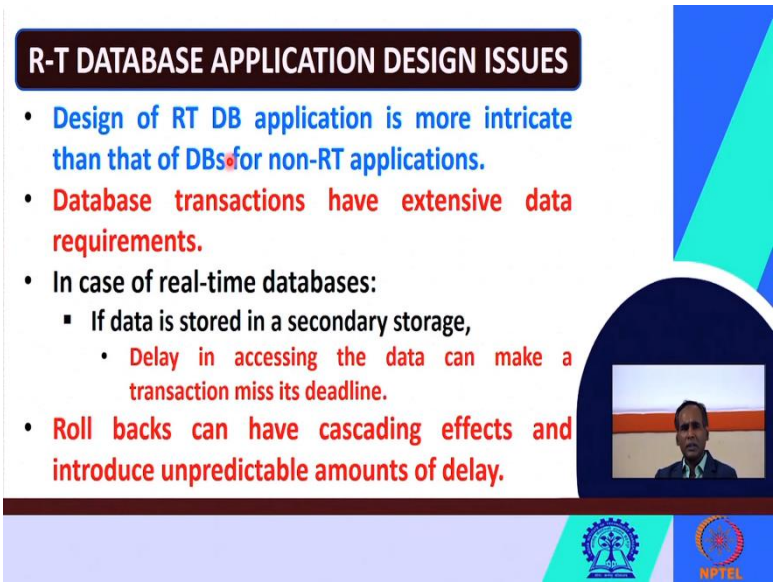
Then the another the last difference in the performance metric. So, normally the transaction response time it is the most metric for all databases. So, whether you are using real-time database or non-real time database, one of the most popular metrics for the databases is transaction response time. For a traditional database system, these characteristics boils down what, how you will find out the transaction response time? For traditional database systems, these characteristics boils down to the number of transactions completed per unit time.

For traditional or conventional database system, these characteristic or this metric that is a transaction response time, it boils down to what, it boils down to the number of the transactions which are completed per unit time. Say, number of transactions completed per second and this metric transaction response time it is used for what, it is used for optimizing the average response time of the traditional application. So, this response, this metric called transaction response time it is used for optimizing the average response time of the traditional applications.

But for the real-time databases, what is happening? For this transaction response time may be a good metric for what traditional databases, conventional databases, but for real time databases this is not a suitable metric. In real-time databases, we are interested in another metric. We are interested for the metric where we can easily find out the number of transactions which are missing the deadlines per unit time that means per second, how many transactions are missing their deadlines because you are discussing what real-time database is. Some timing constraint is there and if a transaction is completed and by that time the deadline is passed, who will accept that transaction.

So, the transactions must be completed before the deadline is achieved, before the deadline is met. So, for real-time databases, this transaction response time is not the suitable metric. The suitable metric is defined as follows. What is the suitable metric? The suitable metric is the number of transactions missing their deadlines per unit time. For example, how many numbers of transactions are missing their deadlines per second? That will be a suitable metric in case of real-time databases in comparison to the transaction response time in case of the traditional databases.

(Refer Slide Time: 25:20)



Now, let us quickly look at the database real-time database application design issues. What issues you must consider thoroughly while designing the real-time database applications? Normally, you know design of real-time database application is more intricate, it is more difficult than that of the databases for non-real time applications. So, designing of databases for non-real time applications is simple, but designing a real-time database application is much more intricate is much more difficult.

The database transactions have extensive data requirements. So, in case of the traditional or non-real time applications, the database transactions have what, they have extensive data requirements. The data requirement is more but in case of real time databases, if data is stored in any secondary storage, what will happen. So, here also in real-time data also or in real-time databases also, the database transactions they have extensive data requirements in case of the real time databases, if the data is stored in a secondary storage, what will happen. Secondary storage means just like disk storage.

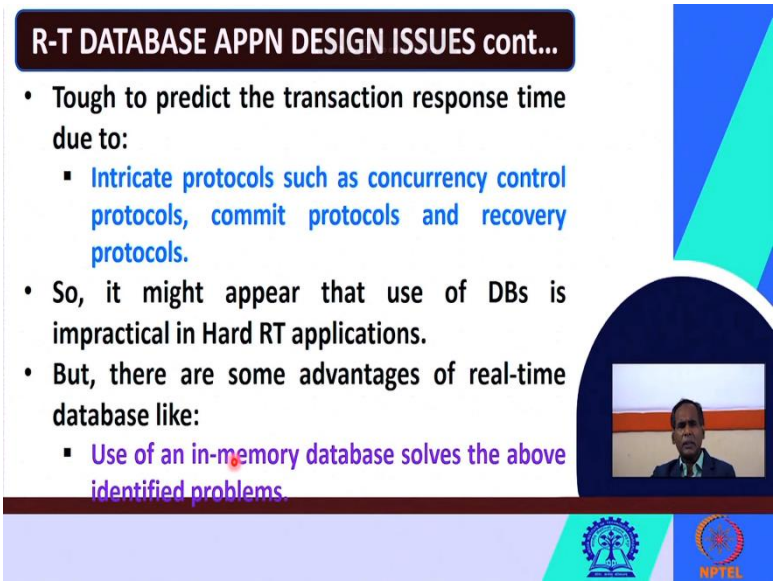
So, delay in accessing the data can make a transaction miss a deadline. I am saying again. In also real time databases, the database transactions, they have extensive data requirements. They have extensive data requirements, in the case of real-time databases, since the database transactions have extensive data requirements if in that case, the data is stored in a secondary storage such as disk storage then there will be heavy delay, large delay in accessing the data which may make a transaction miss its deadline.

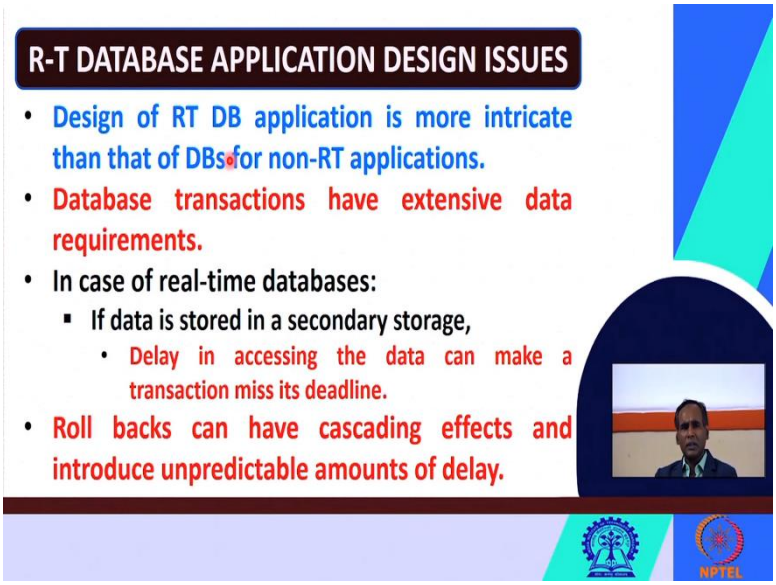
Since, if you are storing the data in a secondary storage like disk storage and you know assessing data in case of a secondary storage takes much more time and by this time, the transaction may miss its deadline. So that is why what should be the storage medium that is an important design issue in case of designing real-time databases. You should avoid using disk storages for storing the data in a real-time database and also you know, rollbacks can have cascading effects and introduce unpredictable amounts of delay.

If you are using secondary storage just like storage, disk storage the delay will be more by that time the transaction might miss its deadline. Also, you know that the rollbacks they can have cascading effects, what is cascading about etc. we have already seen. So, the rollbacks can have cascading effects and these rollbacks may introduce unpredictable amounts of delay. So, by doing so, during this time of delay or during this time what will happen, some of the transactions again may miss its deadline. So, whether to use rollback or not, you have to judiciously think while designing what relational databases.

While designing relational databases, you have to think of whether to use rollback or not because if you will use rollback, they may have cascading effects and they will use they will introduce unpredictable amounts of delay.

(Refer Slide Time: 28:26)





It is very much tough to predict the transaction response time. In case of this what real-time database, it is very much difficult it is very tough to predict the transaction response time. Why? Because due to the use of intricate protocols such as concurrency control protocols, comment protocols and recovery protocols. So, due to the use of all these protocols, it is very much difficult to predict the transaction response time in case of what real-time databases. This is another important design issue. So, you may think that use of databases is impractical in hard real-time applications.

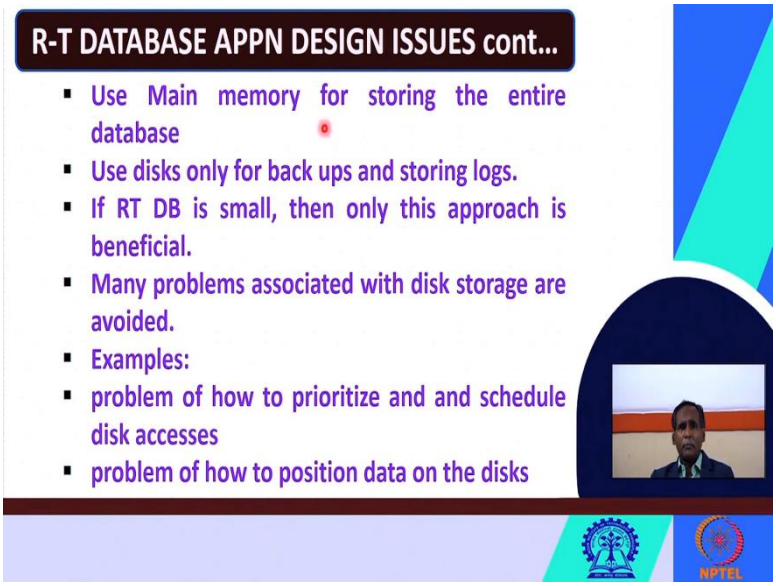
You may think that no if you only use these databases, so much delayed and by that time the what transaction might miss the deadline. So, it is not feasible to use databases in hard real-time applications. So, from the above discussions it might appear that use of databases may be impractical in hard real-time applications, but still there are some advantages of real-time databases. So instead of these drawbacks, but there are still some advantages of using real time

databases. I have already told you, you should not use secondary storages like disk storages in case of what real-time database then what you should use?

You should use in-memory database, you should use the main memory to store the entire database and if you will use an in-memory database, it will solve the above identified problems. So, I have already told you some problems like delay, and these cascading effects, unpredictable amounts of delay, etc. So, it is also very much difficult to predict the transaction response time. So, all those problems are very much related to the use of these what secondary storage or disk storage to store the database. So, if we can use the main memory to store the entire database, then all those problems may go away.

So, now but there are several or some advantages of using real time databases like if we will use or use of an in-memory database solves the above identified problems. You can, if you can use a main memory to store the entire database, then all those problems can be avoided.

(Refer Slide Time: 30:40)



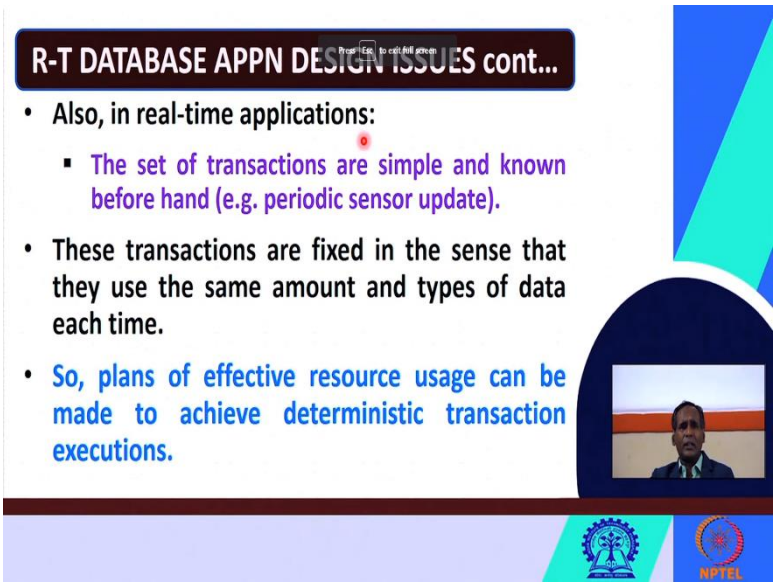
So, what do you, what is the suggestion? You should use main memory for storing the entire database, while you may use the disk only for the backup purposes and only for the storing the logs otherwise the entire database should be stored in the main memory. Then see if your real time database is very large than this approach will be fail because you know a memory size is less. So, this approach will be beneficial only when the real-time database its size is small, its size is sufficiently small then only this approach is beneficial that means using main memory for storing the entire database will be beneficial.

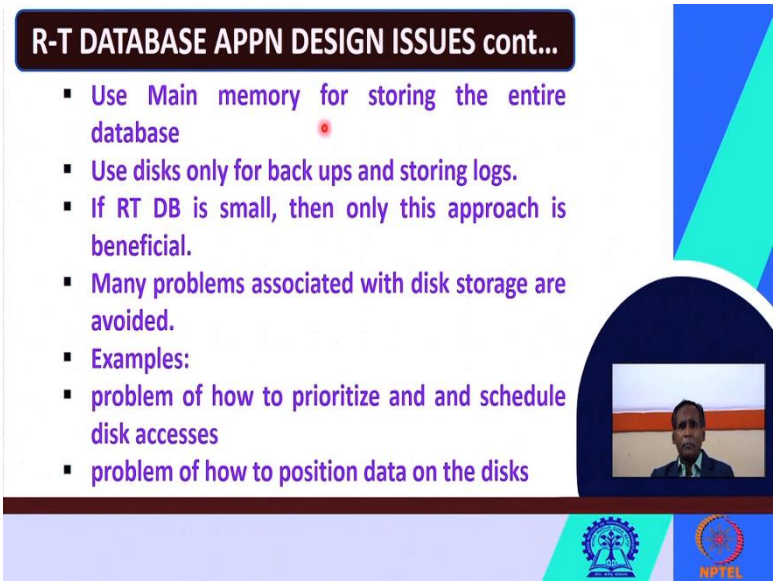
If we will use the main memory for storing the entire database, you can get also several advantages or in other words I can say that many problems associated with the disk storage they can be successfully avoided if we can use a main memory for storing the entire database. What problems can be avoided? For example, if you are using the disk storage, you have to determine how to prioritize and schedule the disk accesses but if you are using a main memory, this problem can be avoided.

Similarly, if you're using a disk memory or disk storage, you have to decide you have to determine how to position the data on the disk so that the disks assess time this can be minimized. What if you are what using main memory? Then this problem can be successfully avoided. So, these are some of the problems related to disk storage which can be avoided if we can use this main memory for storing the entire database.

So, this is how this is another design issue that you should consider while design the real-time databases. Use main memory for storing the entire database and use disk storages only for the backup purposes and storing logs. So, these main memories can be arranged hierarchically while you are considering the real-time databases. So, this is something about these some design issues for real-time databases.

(Refer Slide Time: 32:43)





Another advantage you can see or another design issue we can see while designing real time databases is as follows. Also, in real-time applications the setup transactions are simple and known beforehand. In case of real time applications, what are the transactions they have to be performed? The set of transactions they are very much simple and the set of transactions are known beforehand, you know them a priori, they are known beforehand. For example, the periodic sensor update.

So, how the sensor is updating the periodic data like say how the temperature data updating? How the sensor updates the temperature values? So, periodic sensor updates. So, in real-time applications the setup transactions they are simple and are known beforehand for example, the periodic sensor updates they are known beforehand. These transactions are fixed. So, in data with real-time applications, these transactions are fixed. Fixed means in which sense? These transactions are fixed in the sense that they use the same amount and same type of data each time.

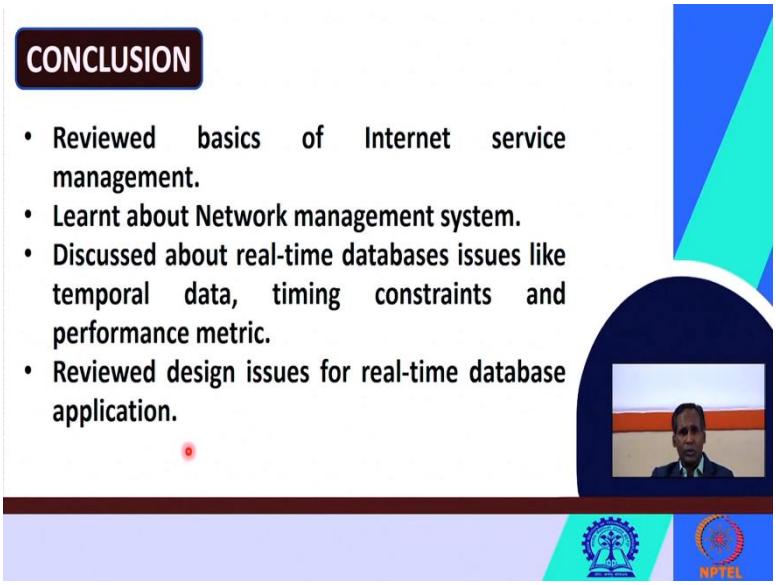
So, in real-time applications, these transactions they are fixed that means they use the same amount of data as well as same time of data each time. So, what is the important design issue here? So, plans of effective resource usage can be made to achieve deterministic transaction execution. So, since these transactions are fixed, so plans of effective resource usage can be made. You should made plans for effective resource usage; those plans should be made efficiently.

Why? In order to achieve deterministic transaction execution so that you can achieve deterministic transaction executions. The transaction executions can be done in fixed amount

of time. So, plans of effective resource usage they can be made to achieve deterministic transaction executions.

So, one more advantage I think I have forgotten to tell while you are using the main memory, this transaction response time also it can be minimized because in case of what disk this transaction execution time. It takes a lot of time but in case of this if you are using main memory, the transaction response time can be reduced significantly. So, we have seen some of these design issues for real-time database applications.

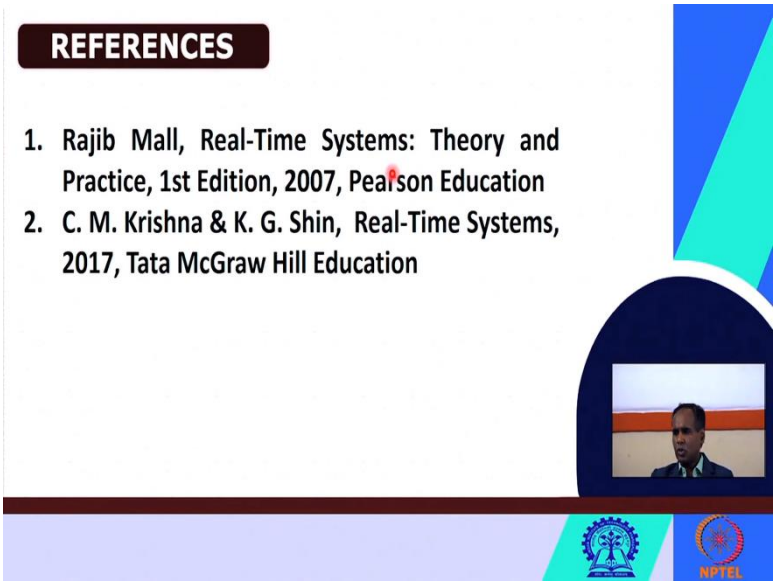
(Refer Slide Time: 35:12)



So, today we have discussed we have reviewed the basics of some applications where real-time databases are used such as internet service management, network management system and this spacecraft control system.

We have also discussed the differences between traditional databases and real-time databases. I have told you there are three important differences are there like temporal characteristics of data, the timing constraints and the performance metrics. We have also reviewed some of the design issues for real-time database applications.

(Refer Slide Time: 35:45)



We have taken those things from these two books. You can see some more contents in those two books regarding the topics that we have discussed on today.

(Refer Slide Time: 35:53)

Thank you very much.