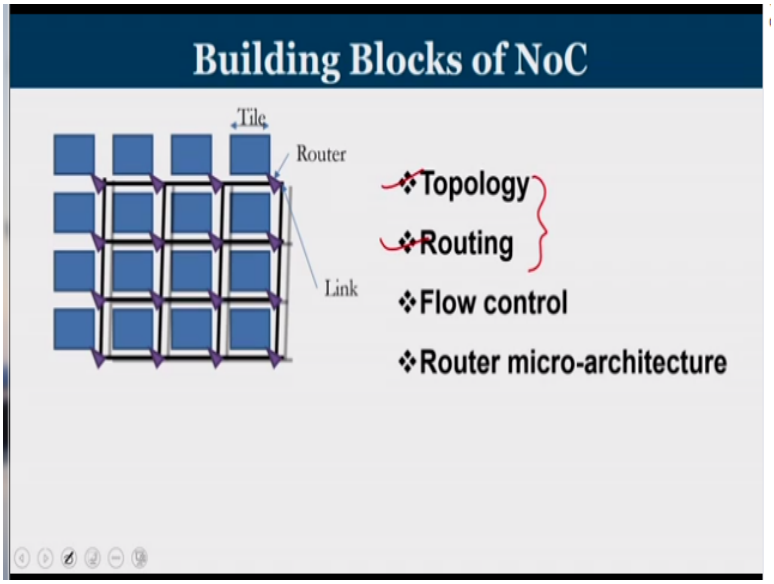
**Advanced Computer Architecture**
**Prof. Dr. John Jose**
**Assistant Professor**
**Department of Computer Science and Engineering**
**Indian Institute of Technology-Guwahati**

**Lecture-31**
**NoC Router Microarchitecture**

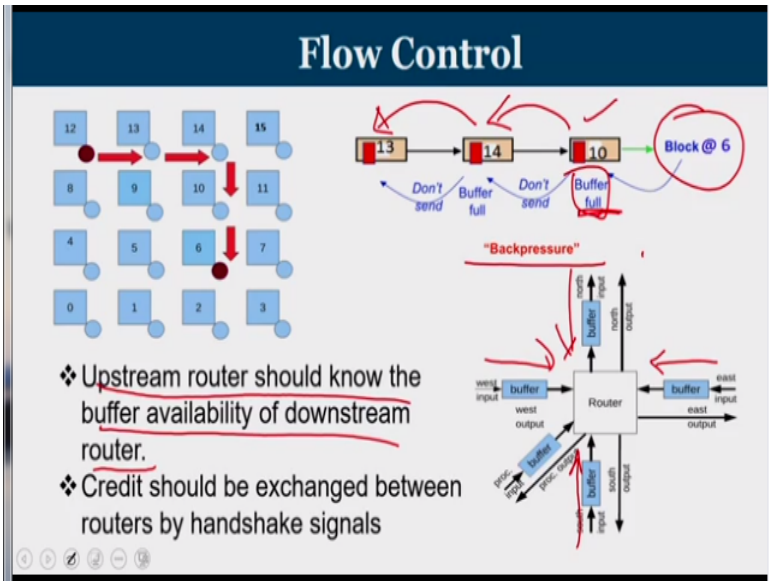
Welcome to lecture number 22, in this lecture our topic of discussion is NoC router micro-architecture. Last lecture we have seen how basically a network on-chip framework works and we have seen about routine and various topologies. Today our focus is more on how will you make sure that a packet from 1 particular router is reaching the adjacent router, what are the kind of handshake mechanisms between adjacent routers and what is the internal organization of an NoC router.

**(Refer Slide Time: 01:04)**




We have seen the building blocks of network on-chip as topology routing, these are the 2 things which we have already covered. And we will now focus on what is flow control and what is router micro-architecture.

**(Refer Slide Time: 01:17)**




The concept of flow control means an upstream router should know the buffer availability of a downstream router. So what do you mean by upstream router when a packet is moving from 13 to 14 then 13 is called upstream router and 14 is called at downstream router. So what it tells it the upstream router should know the buffer availability of downstream router, any router which is going to forward a packet should know whether there is a buffer that is available in the next immediate router, that is a successor router.

And credit should be exchanged between routers by handshake signals, so how will you get it 14 should have a feedback mechanism by which 14 will communicate the number of buffers available with it. So that 13 will know how many packets I can send. Similarly once a packet reaches 14 then as far as this packet is concerned when it moves from 14 to 10, 10 is the downstream router and 14 is the upstream router.

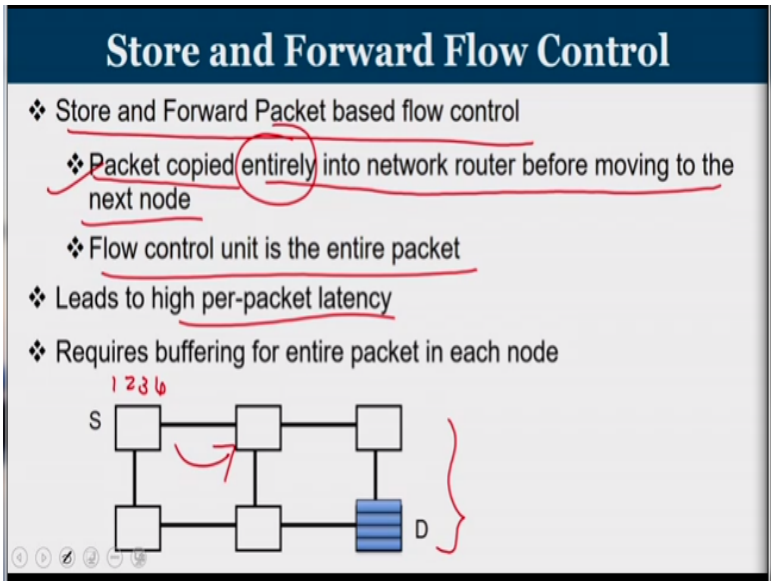
Similarly 10 will become the upstream and 6 will become the downstream at every point every router will have a credit channel through which information pertain into availability of buffers is been communicated. So when there is a case that you just imagine you have a block at the router number 6. Now when you have a block at the router number 6 how generally a block happens if the buffers are full in 6, then 6 is in a position where it no longer can accepts new packet from 10.

So when there is a block in 6 it is going to tell to 10 that my buffer is full you cannot send anything. So over a period of time packets that is coming from 14 will get accumulated in 10 and 10s buffer will be full that scenario will happen. And then 10 is going to say that do not send anymore packets to 14, so packets that will reach in 14, 14s buffer will be full and 14 communicate back to 13.

This mechanism by which the buffer availability status or buffer fullness is been communicated back all the way to the source is called back pressure technique. So you should know that every router in all it is input direction from north, south, east and west whenever packets are coming we first they are going to reside in the buffer. So before sending a packet from 1 router to another checking of availability of buffers is a must.

So flow control plays an important role in ensuring the smooth exchange of packets from one router to the downstream router with the help of efficient handshaking mechanisms.

**(Refer Slide Time: 04:00)**




Now let us see how a packet is moving from one router to another, we have learned in the flow control technique that there should be a handshaking mechanism wherein adjacent routers no each other what are their capabilities in terms of storage. Now how are the packets going to move, what is a granularity at which this packets are going to move that is what is why we are going to learn.

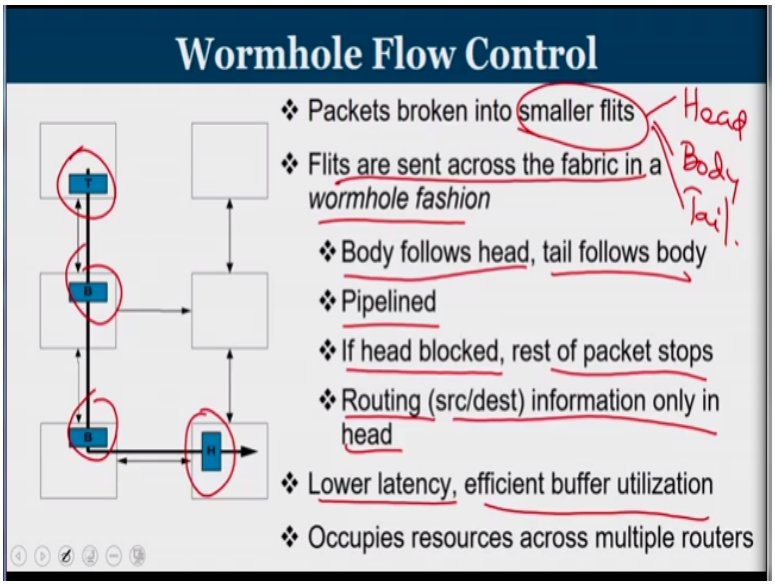
Our conventional networks, traditional computer networks use store and forward switching mechanism, you receive a packet fully stored it perform error detection and correction. And thn once everything is fine then from that router to the next router you are going to forward the packet, that is called the store and forward packet switching scheme. We will try to understand what is store and forward and then we will take it forward what are the specialized kind of switching that is been used in network on-chip routers.

So store and forward packet based flow control is a conventional one used in traditional computer networks. Now in this case packet is copied entirely into the network router before moving into the next node. And flow control unit is basically your packet, so the entire packet move from one router to the next router only after successfully receiving the entire packet it is been forwarded to the next downstream router.

So consider the case wherein you have to send 4 this a packet which consists of 4 splits from S to D. if it is a store and forward switching then this is going to lead high per-packet latency we will see why. But it is requires buffering of the entire packet in each node, so even though a packet is divided into 4 smaller units called flits. Only if all the flits reaches the next router then only it is going to be forwarded.

So you have the first flit coming, second flit coming, third forward by fourth, only at this point this packet is now completely stored. So packet is copied entirely into the network router before moving into the next one. And then the next step wherein all the flits are been moved and then finally from there it is going to reach the destination. That is a mechanism by which store and forward switching works because of this every router need to have capacity wherein it can store the entire packet. So if the packet size is going to 20 flits, the buffers should have 20 split capacity.

**(Refer Slide Time: 06:34)**




Now we know that packets are broken down into smaller flits and flits are send across fabric in a wormhole fashion. This is the switching techniques or the flow control techniques that is been employed in network on-chip. So we divide the packets into smaller flits that is what the concept we have know and they are consist of head flits then you have body flits and then at the end you have tail flits.

So head flit, body flit and tail flit constitute your total packet and the body flits will simply follow the head flit and tail follow the body flit and the movement is pipelined. See if the head flit is blocked the rest of the packet stops and routing that is source and destination information is all available only at the head. So we will try to understand this concept you have a packet which is further subdivided into smaller component called flits.

The first one is called head flit which has lot of control information especially what is a source address, destination address, sequence number or any other parameter. Now the head flit will move, the head flit is going to reserve a buffer for the remaining body flits and head flit need not wait there like what we have seen in store and forward packet switching mechanism where the entire packet needs to reach a router in order to move forward.

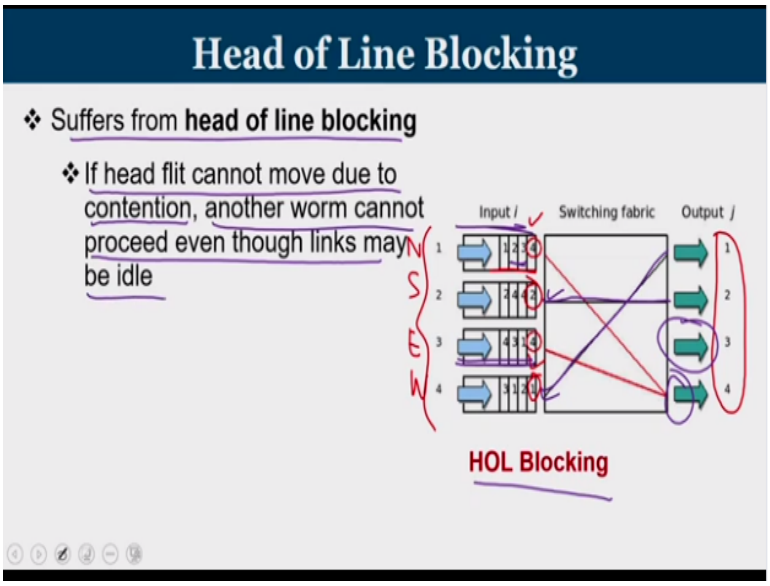
Here there is no such restriction as and when output packet or output port is available and there is buffer availability in the downstream router, every flit is free enough to move to the downstream

router. So head flit may move in the meantime body flit may come or if you look at any snapshot or any window the different flits of a same packet will be present across multiple routers and this concept is known as wormhole routing.

If you look at the diagram that is given here we can see that the head flit has already reached here and the 2 body flits are in adjacent routers and the last tail flit is in a different router. The advantage here is the buffering capacity of each of the router need not be equal to the size of a packet. We are never enforcing that all flits of a packet should reside inside a same router, so even if we have a buffering which is 1 or 2 flits still the whole concept will work.

And this lead to lower latency of the packets because we are not at all been blocked until your tail is going to reach as and when the head flies away to move further the head moves. Similarly everybody flit also moves like that, so this is an efficient utilization without having the entire packet buffers we can still manage easy forwarding of the flits. And generally a packet occupies resources across multiple routers.

**(Refer Slide Time: 09:19)**




Now let us try to understand another important concept it is called a head of line blocking, so think of a case that since we told we have buffers. Consider the case that you have 4 directions let us say north, south, east and west and we got flits that the residing inside these input buffers.

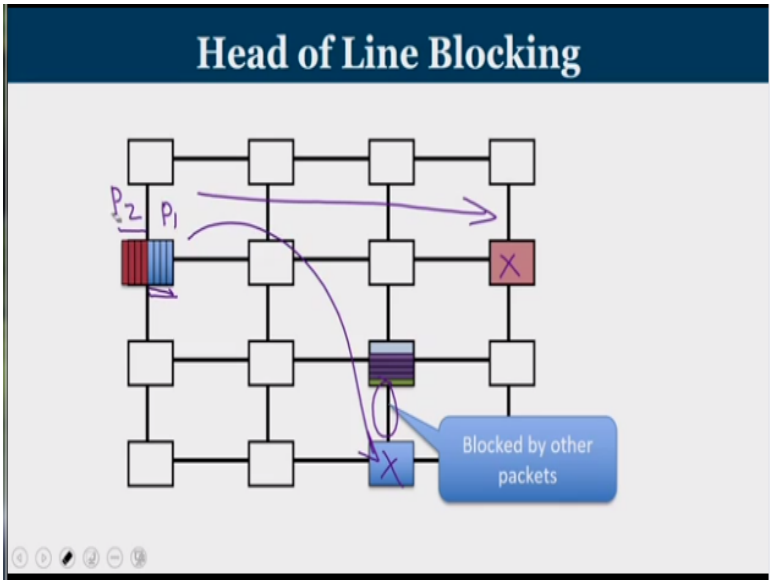
Now let us imagine these are the 4 different output ports where 1 indicates north, 2 indicates south, 3 is east, 4 is west.

The numbers here indicate what particular direction or which is the direction that each of these flit is looking for. So here you have the first flit in the queue is looking for output port 4 whereas here the first flit is looking for output port 4. In this case this flit is looking for output port 2 and here we have the very first flit looking for output port 1. Now look at this scenario what happens you have 2 front flits, the first flit in input port number 1 that is this.

And the first flit in input port number 3 both are competing for the same output port 4, we know that only one can be granted the other one has to wait. But whereas in this case there is only one candidate looking for 2 and one candidate looking for 1. So this can directly go to 2 and this can directly go to 1, we can see that nobody is able to move to output port 3. Because we do how flits which require output port 3 but they are not in the front of the queue.

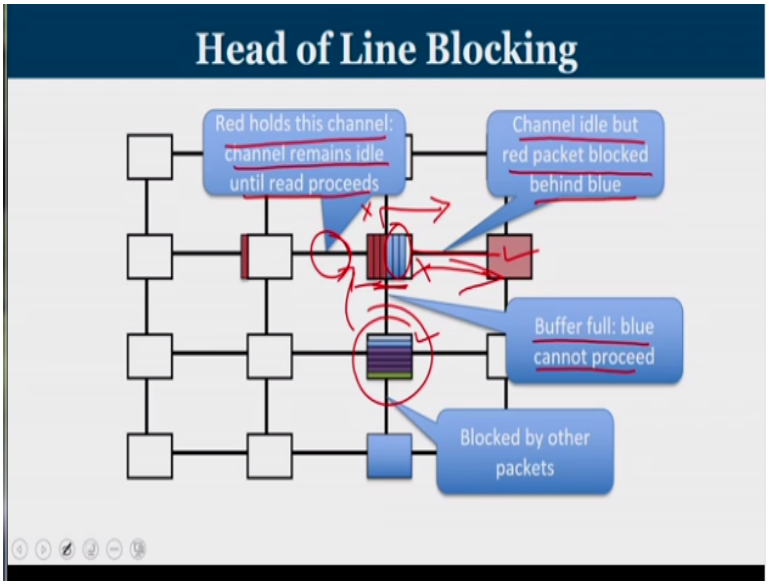
So this is known as head of line blocking, one of this 4 is actually not allowing a flit which actually want an output port but because of the organization it cannot be done. So generally this kind of buffering will suffers from head of line blocking. If head flit cannot move due to contention another worm cannot proceed even though links maybe idle.

**(Refer Slide Time: 11:24)**



Let us try to see with a help of an illustration how head of line blocking really happens, think of a case that you have a scenario wherein this particular green packet cannot move because there is a block there. Now imagine you have these set of flits which is marked by blue color you have 4 flits which constitutes their packet P1 whose destination is this blue. And your red color 4 flits of them are now residing inside a queue whose destination is this. So this is a scenario the blue ones wanted to go to this point and the red one wanted to go to this point.

**(Refer Slide Time: 12:11)**

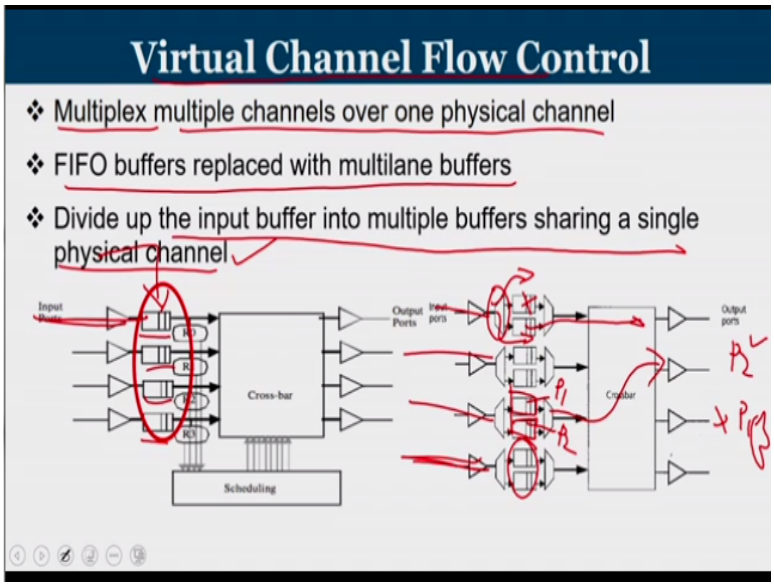



Let us now try to see what is head of line blocking, so the blue packets are coming flit by flit and they are going to occupy, so here we can see that 1 blue as occupied. Now this buffer is full, so because of the buffer is full the blue cannot proceed over there, so naturally due to the back pressure mechanism blue is getting accommodated in this router. So the subsequent blues will come and then followed by the red is coming.

But the reds destination it has to go straight, the channel is ideal but the red packet is black because it is occupying after blue in this queue. All other reds will come subsequently come over there, so now the red hold this channel, the channel remains ideal. So because of this even though the channel is ideally it cannot move. And this scenario is known as because a blue is residing there, the blue cannot move because of that the red which occupies after blue is getting blocked this is called head of line blocking.

So even though the red wanted to move in this direction, the buffer is free, the link is free but my predecessor is blocking me.

**(Refer Slide Time: 13:24)**



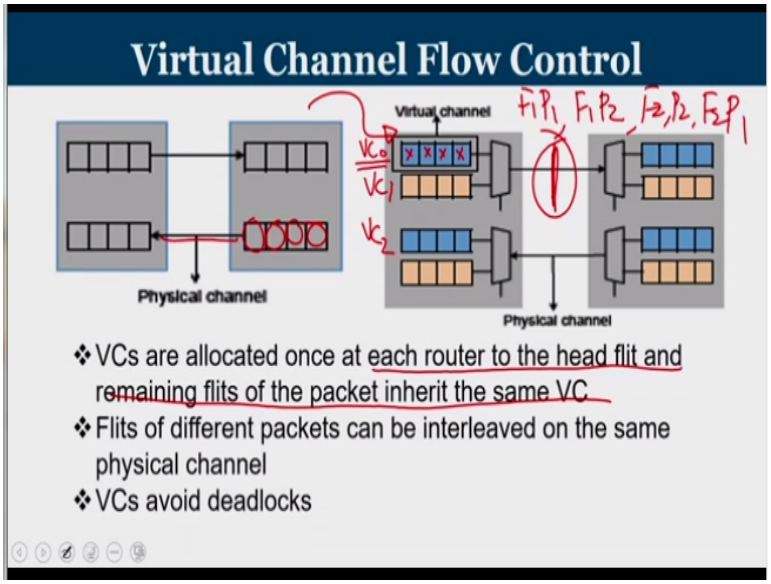

The solution for head of line blocking is virtual channel flow control, so what we do is multiplex multiple channels over a single physical channel. So generally what we have is the limitation what we have address now when you come through a link when you are going to enter into a router you have a FIFO queue you can see that a set of flit buffer space is been organized as a queue.

So FIFIO buffers are replaced with mutilane buffers, so what we generally do is divided up the input buffer into multiple buffers sharing a single physical channel. So you can see that when a flit comes you are using a demultiplexer using which the flit will either reside here or reside here. So depending on the output direction or depending on the packet number we are providing multiple virtual channels.

So this is known as a physical channel, you have many physical channels the point that which it enters the router it is being split across multiple channels they are called virtual channels. So even though one of them is blocked the other one which may come after this one can still move further. So we are avoiding head of line blocking and each one virtual channel will be accommodated by flit of 1 packet.

Similarly this virtual channel will be accommodated by flits of an another packet, so even though let us say this P1 flits of P1 flit of P2. Assume that flits of P1 reach this router before flits of P2 but if the output direction for P1 let us say it is this, this is for P1 but if it is a block there that means there is no buffer availability in the downstream. But if P2 wanted to go in this direction where there is buffer availability P2 will move out first, so this will avoid head of line blocking.

**(Refer Slide Time: 15:18)**

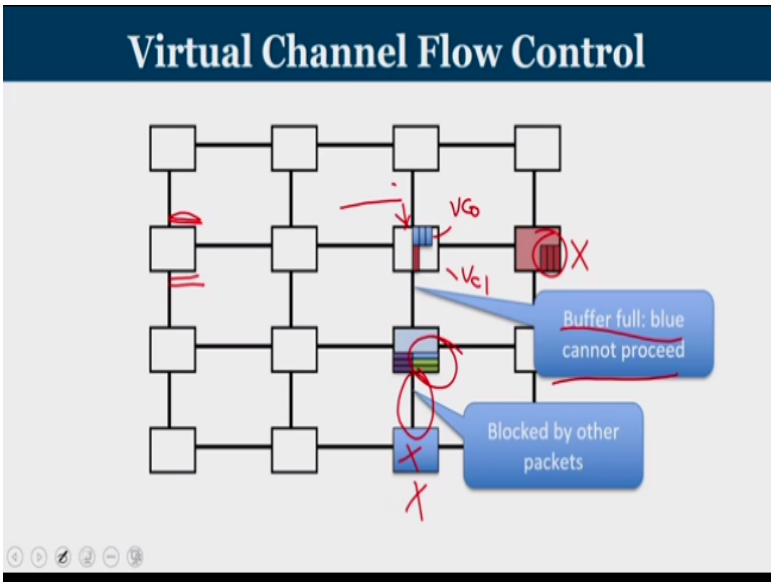



Now coming into virtual channel flow control, what we have seen the physical channel which terminates at the end of for buffers is now been divided into multiple lanes multi lines which is known as virtual channels. Virtual channels are allocated at each router to the head flit and remaining flits of the packet inherit the same virtual channel number. So this maybe the head flit and these are body flits of the same packet, so whatever you see for example this VC 0.

Let us say this is VC 1 then VC 2 if VC 0 is assign to one particular packet all flits of the same packet will come and reside inside the same virtual channel. Flits of different packets can be interleaved on the same physical channel. So if you look at here if can be flit 1 of packet 1 that is going followed by flit 1 of packet 2 followed by flit 2 of packet 2 it can be then flit 2 of packet 1 like that in whatever interleaved order it comes as long as it know which virtual channel they should go and sit, the problem is resolve.

So it will eliminate head of line blocking and it make sure that all flits of a particular packet are been organized systematically this virtual channels will avoid deadlocks.

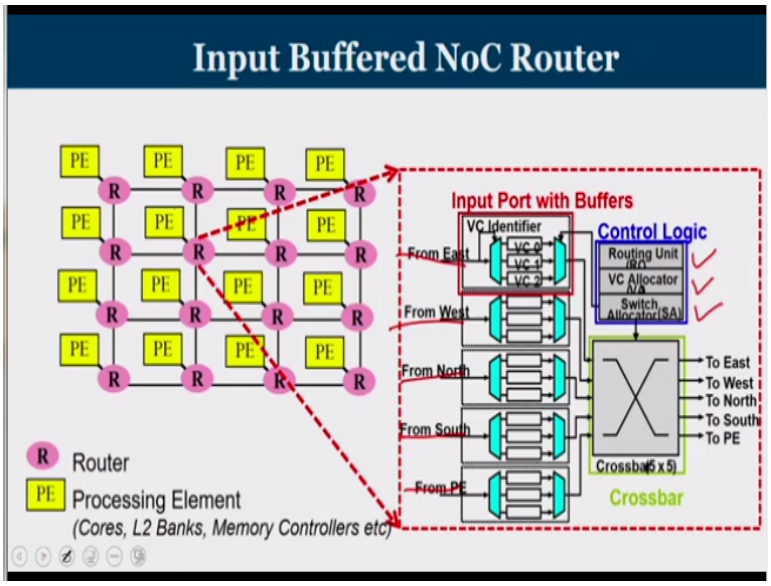
**(Refer Slide Time: 16:41)**




So now we will see how virtual channel flow control happen, the same scenario let us imagine that there is a block over here due to non availability of buffers in this particular router. Now you have blue packets or flits of blue packet which is looking for this as the destination and flits of red packet which is looking for this as the destination. So the blue is coming and then blue getting block, so imagine that this buffer is full, so buffer full blue cannot proceed.

So the blue is getting stop that this point, so all the subsequent flits of blue will come and reside there now your red is coming, so red will sit in a different virtual channel. So now you can see this is virtual channel 0 and this is virtual channel 1 and we know that red came after blue but red is to a different destination, so red is go into move further, so that it will reach the destination.

**(Refer Slide Time: 17:35)**

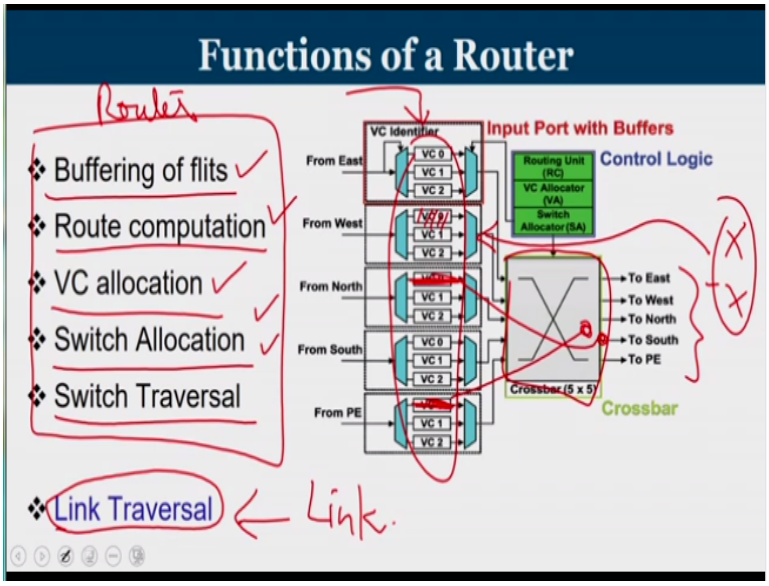



So even though blue is blocked there, blue is not allowing to block the red. Hence virtual channels will help us to avoid head of line blocking and enable smooth transition of packets from one router to another. So this is the general layout of a TCMP structure where we have processing elements connected by a grid of routers. Now we will see what is there inside the router, the routers has buffers which are generally called virtual channels.

And we have physical channels which terminate these are all the physical channels connected to the neighbor. So the physical channel is getting terminated on the virtual channels and we know that the purpose of a router is to connect or is to forward packets from input direction to output direction. So there is a cross bar which connects the input to output and then we have a control logic which facilitates the forwarding of the packet from one input to another.

There are basically 3 components for the control logic one is the routing unit, second one is a virtual channel allocation and the third one is the switch allocation operation.

**(Refer Slide Time: 18:42)**




We will now examine what are the functions of a router, we all know the routers are going to be the single point of entry for data from a tile into the network. Now apart from handling the packets of this source tile, the routers also acts as a shared resource for all other tiles. Because there packets are been carried through this router, so there is an intelligence that is available in the router this router should intelligent find out what is the source address and destination address of the incoming packets.

And how to do processing on it sometimes forward them and sometimes deflect them we will see what are the functions of the router. The first and foremost function of a router is for every incoming flit I need to first buffer them, so we make use of the virtual channels to buffer them. The second operation is route computation, route computation is the process by which for every incoming flit you have to find out what is the output port.

The process of finding out an output port for every incoming packet is called route computation. Next operation is called virtual channel allocation, we know that a packet upon reaching a router is first place inside these buffers known as virtual channels. Similarly if a packet that is residing in one of the virtual channel after performing the routing it need to reserve a buffer in it is downstream router, so the downstream router is here.

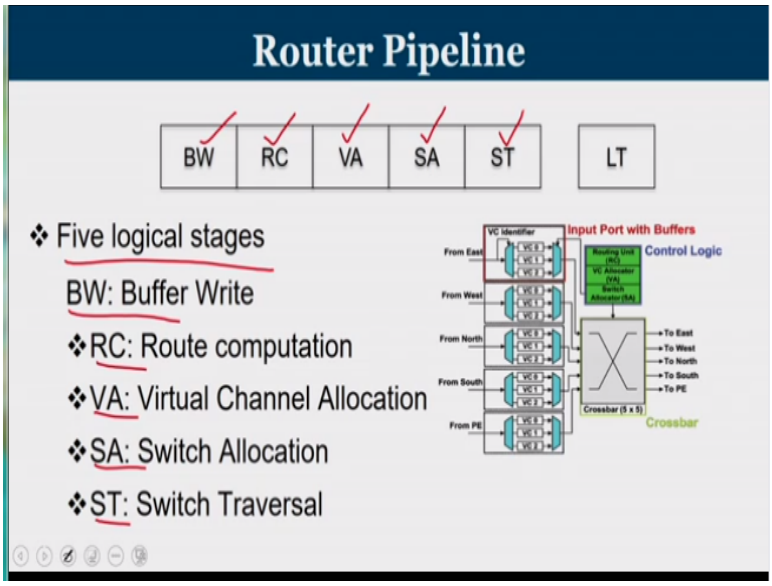
From the downstream router I need to get a feedback mechanism by which I will know which of the virtual channel is free. The process of reserving a buffer in the downstream router is known as virtual channel allocation. We now move onto switch allocation, switch allocation is a process, imagine that you have this particular flit which reside in VC 0 is looking for south direction.

Similarly a flit it is shimming in VC 0 of the processing element that also wander to move to the south direction. So if 2 flits are trying for a common output port then we need to resolve the conflict one has to be picked. So when multiple flits compete for the same output port the winner is chosen by a switch allocation algorithm. So first the flits will come and stay in the buffers perform the routing find out output port.

In the neighbor corresponded to that output port a buffer has to be reserved in the downstream router and then you have to win the switch. If there are multiple parties that is looking for the same output port, you have to choose the winner. Once you winning it then you are travelling through the cross bar switch that is a process by which a travel through the cross bar switch. And then at the end we have the link traversal, once you cross the cross bar then it are ultimately landing up in the link.

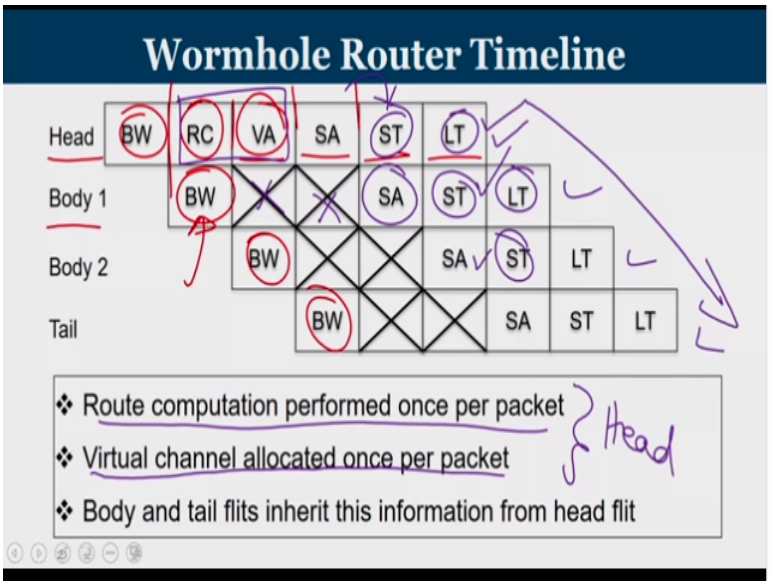
So these are the functions that happens inside the router and this is happening outside the router, so it is basically a link operation.

**(Refer Slide Time: 21:42)**




Let us see these are of 5 independent operation that we have seen buffer write, route compute, virtual allocation, switch allocation and switch traversal. And this can be done in a pipeline fashion, so we call it as a router pipeline. The simplest of pipeline is a 5 stage logical pipeline where you have a buffer write operation, independent of route computation operation which also is independent of virtual channel allocation, switch allocation and switch traversal, so it is a 5 stage router that we have seen.

**(Refer Slide Time: 22:13)**




Now can I pipeline it, so I perform the buffer write operation for the head flit in the very next cycle I will get the body flit. So that buffer flit, so in each and every subsequent cycle I am performing the buffer write when I am performing buffer write of the body flit or writing the

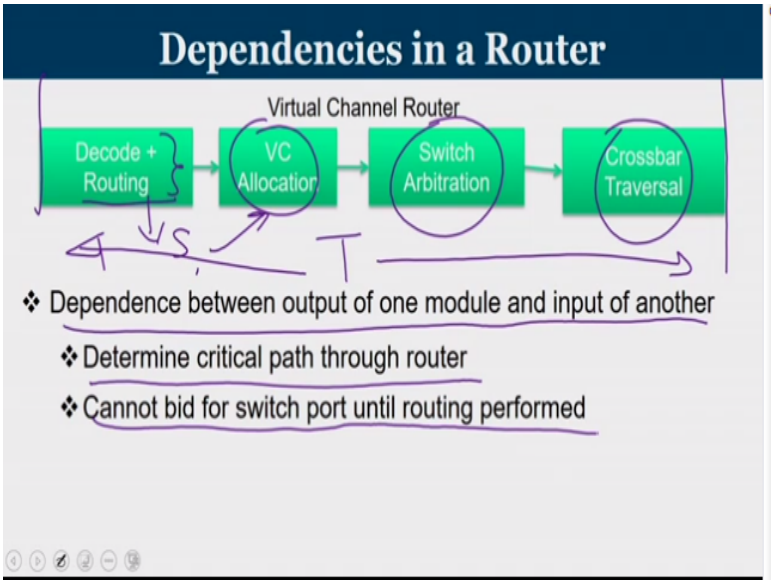
body flit into the virtual channel, the head flit undergoes route computation. Once a head flit undergoes route computation it goes to virtual channel allocation.

That also will be done only for the head flit and then we have switch allocation, switch traversal and link traversal. Virtual channel allocation and route computation these 2 operations are done only for the head flit. So during that time the body flits would not do any operation. Now if you can see that the link traversal and the switch traversal happens in a pipelined fashion when the head flit is performing the switch traversal.

The body flit is been going through switch allocation when head flit is traveling through the link first body flit is performing switch traversal, the second body flit is performing switch allocation. Like that if you look at here in every cycle one flit is through the link, this is possible even though each of the flit has to go through 5 different cycles inside a router because of the pipelined operation we get better throughput.

Route computation is performed only once per packet, virtual channel allocation is also done once per packet and that is done only on the head flit. Body and tail flits inherit this information that is the route computed for a head flit and a virtual channel allocated for a head flit. And there been simply followed in the case of a body flit and tail flit.

**(Refer Slide Time: 23:58)**



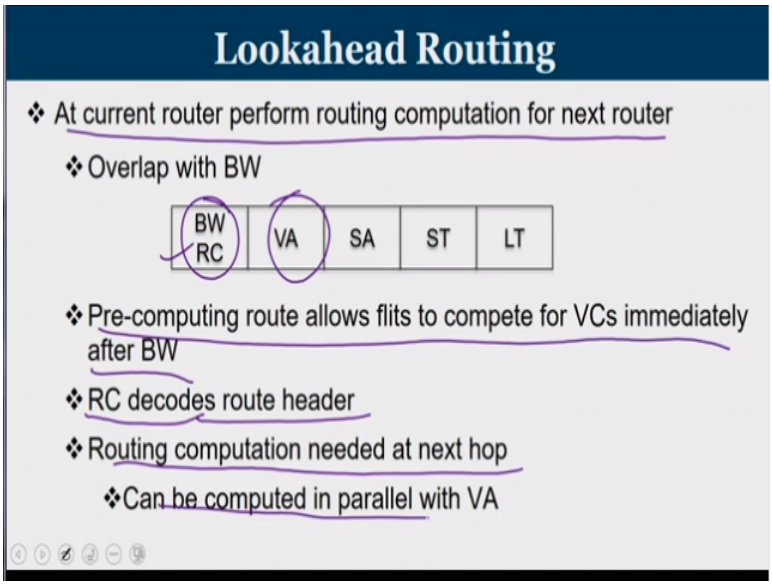
Now let us see what are the dependencies, if you decode what is there in the head then only you can perform routing only if the routing is over then only I can perform virtual channel allocation. If the routing tells that your output is south then I have to look into the south neighbor and find out whether there is a buffer available or not, that is the dependency between them. Now if I perform virtual channel allocation that means for a flit there is a buffer that is available.

If there is a buffer available all those flits who got successful buffers reserved in the downstream router they only participate in switch allocation. And if those who got switch they can only traverse through the crossbar. So the dependency is I can perform virtual channel allocation only if the routing is over. Similarly if switch allocation can be done only if the VC allocation is over similarly I can travel crossbar only if I am allocated with a switch.

So dependence is between output of 1 module and input of another is been shown in this diagram. So this shows the critical path through the router, so if I tell what is the total processing time in the router it is a time from this point to that point, that is a time what I am talking about. So if I wanted to reduce the amount of time a packet is pending in the router then each of these component if you try to reduce we will get.

So they are all basically in the critical path, meaning you cannot bid for switch port until the route is been performed because of this dependencies.

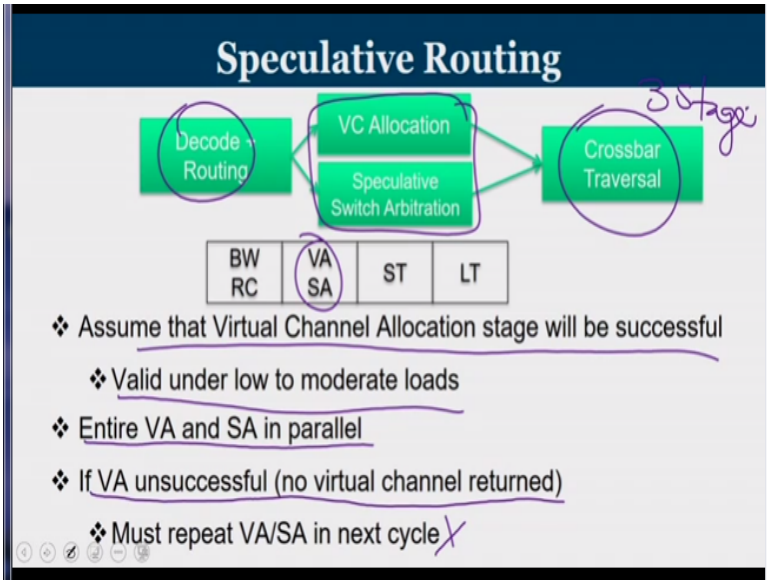
**(Refer Slide Time: 25:23)**




Now there is something called look ahead routing, at current router you perform the route computation for the next. So you can overlap buffer write with the route computation, so pre-computing route allow flit to compete for VCs immediately after the buffer write. So by this process I am performing the buffer write at the same time I am performing the route computation because of that immediately after that I can perform virtual channel allocation.

So the route **the route** computed is decoded into the header and routing computation needs to be done at every hop and can be computed in parallel with VA as well.

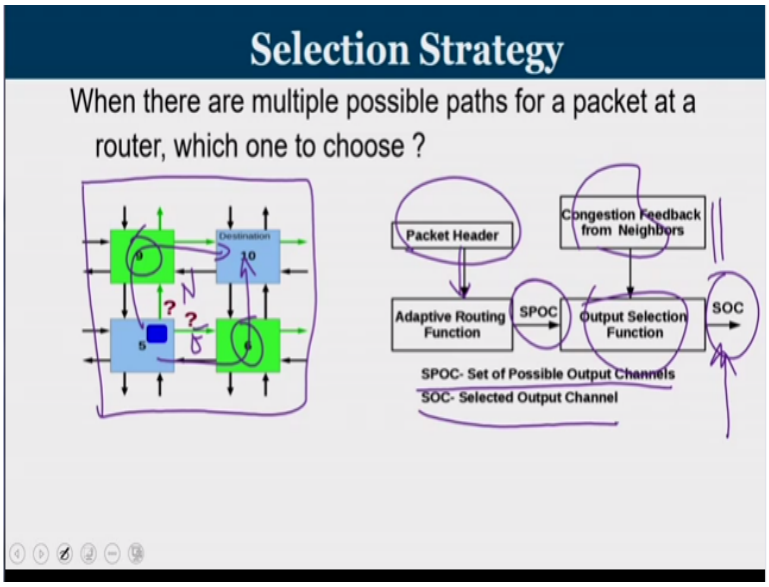
**(Refer Slide Time: 25:58)**



So this is the called the speculative routing where virtual channel and switch allocation or switch orbitarization is done together. So assume that virtual channel allocation stage will be successful and your virtual channel allocation and switch allocation happens in parallel. So this will valid under low to moderate load when the load is very high there is no guarantee that you will get a virtual channel allocated, so in that case switch allocation will fail.

If virtual channel allocation is unsuccessful then you are not going to perform the switch allocation operation. So you perform routing VC and switch allocation is done parallel and then after that you travel through the crossbar. So this essentially make it as a 3 stage router, 1 cycle for routing, one cycle for VC and switch allocation and one cycle for crossbar traversal.

**(Refer Slide Time: 29:50)**



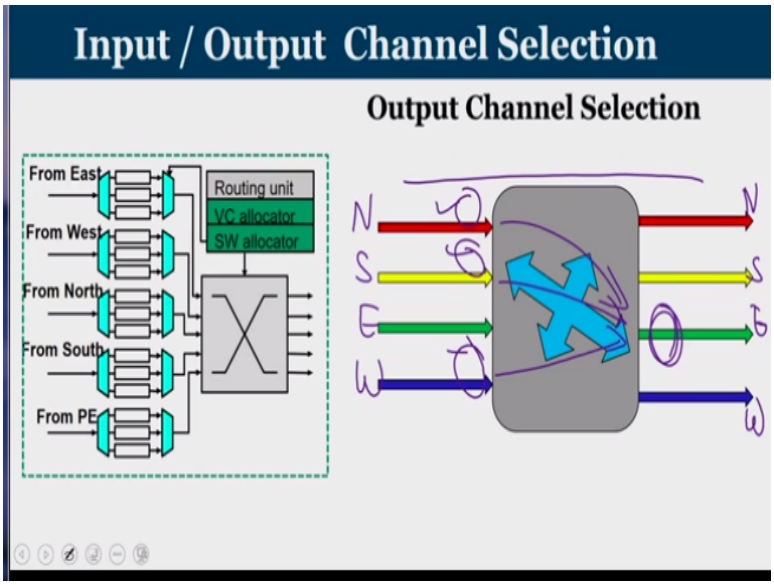

Now what is a selection strategy when there are multiple possible paths for a packet in a given router which one to choose. So imagine the case that you got a packet that is there at router number 5 what is shown in the diagram. If 10 is a destination either I can travel through this way or I can travel through this way. So 6 is a potential neighbor 9 is also a potential neighbor, so if both are desired output ports which is one that you are going to choose.

So generally from the header you will know what is a destination, so the adaptive routing function will return a set of possible output ports, this will tell that east is an output port, north also is an output port. And then based upon congestion, for example you can try some gather

some information that 6 will gill number of buffers in 6 and buffer availability of 9 is communicated by 9.

So based upon this congestion information that you obtain from the neighbors your output selection function will return 1 output port. So from many output port from set of possible output channel I am choosing one of the channel and that is been done. So this whole process of collecting the feedback and determining one of the output port is known as output selection strategy.

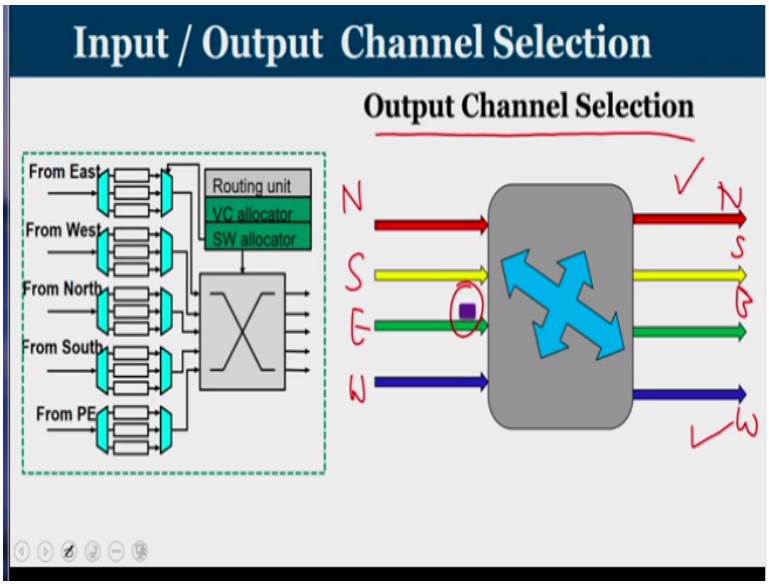
**(Refer Slide Time: 27:59)**




Now we **we** will just distinguish between what is input and what is output selection strategy. So consider the case that you have 3 flits that is coming. Let us imagine these are north, south, east and west input ports and similarly we have north, south, east and west input ports. So imagine you got flits which is been marked with this green color. And if all of them wander to move through 1 output port like this if they all wander to move through this output port the green color which of this flit should be given preference.

So the case is you have 3 flits coming flit 1, flit 2 and flit 3 all of them as looking for same output port I can permit only one at a time. So 2 has to be buffered and then they should try their luck in the subsequent cycles which of this packet I will choose that is called input channel selection similarly we will try to understand what is output channel selection.

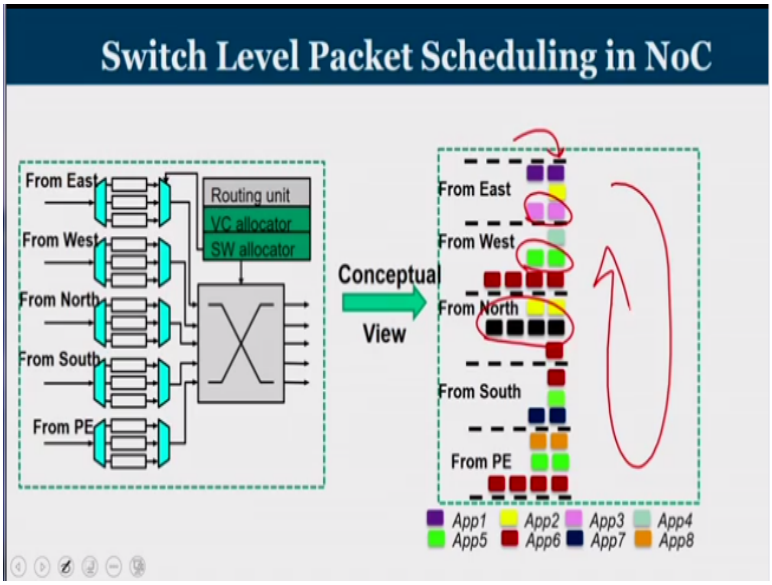
**(Refer Slide Time: 29:04)**




So imagine that this is north, south, east and west input ports and these are north, south, east and west output ports. So consider the case that you received a packet to this coming through the east input port reaching an NoC router. Now upon computation of the route it was found that it is eligible for traveling through the west as well as traveling through the north. Sometimes you may find more than 1 output port, so 1 packet more than 1 output port which output port choose that is known as output channel selection.

And we have seen in the previous case when you have multiple flits looking for the same output port which one to choose that is called input channel selection.

**(Refer Slide Time: 29:51)**

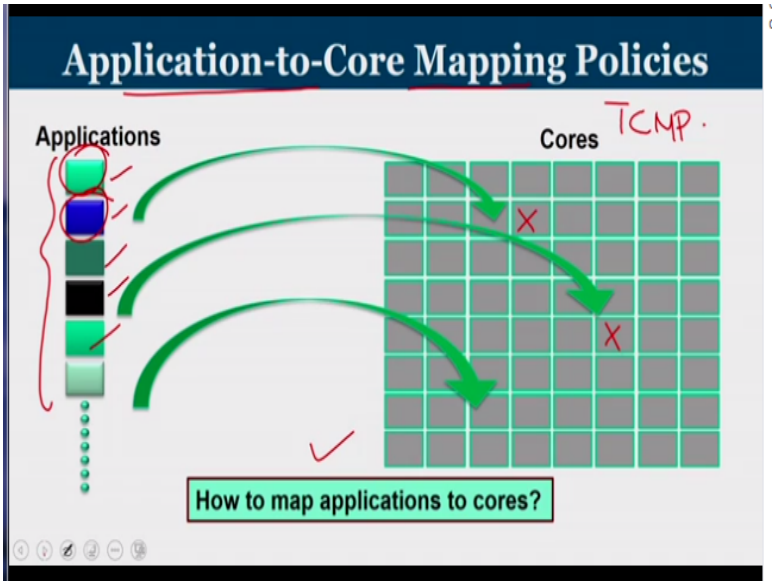



So you think you have flits of different packets, you can see that each of the virtual channel holds only saying color they are flits of the same packets. Here you can see same color, so this is a logical representation of flits of same packet occupying the virtual channel router. These are all belonging to different applications and the duty is you have to find out which one to choose here that is all about the duty of as scheduler.

So scheduling itself is a totally different domain but we stop also discussion with this level as far as the internals of NoC router is concerned. Before moving onto the next topic let us try to summarize what we were discussing till now. We started with how a packet is been forwarded to one router to another, so we are gathering this feedbacks this feedbacks are obtained from your neighboring routers in terms of credit information.

And this handshaking between 2 adjacent routers facilitate the smooth flow of packets from one router to another. we have seen about virtual channels which will avoid head of line blocking and wormhole routing that is generally employed in network on-chip routers. We have seen the internal structure of a structure of a router wherein you have a buffer write operation, we have route computation, virtual channel allocation, switch arbitration, switch traversal and then followed by link traversal and we have seen the pipelined structure of a router also.

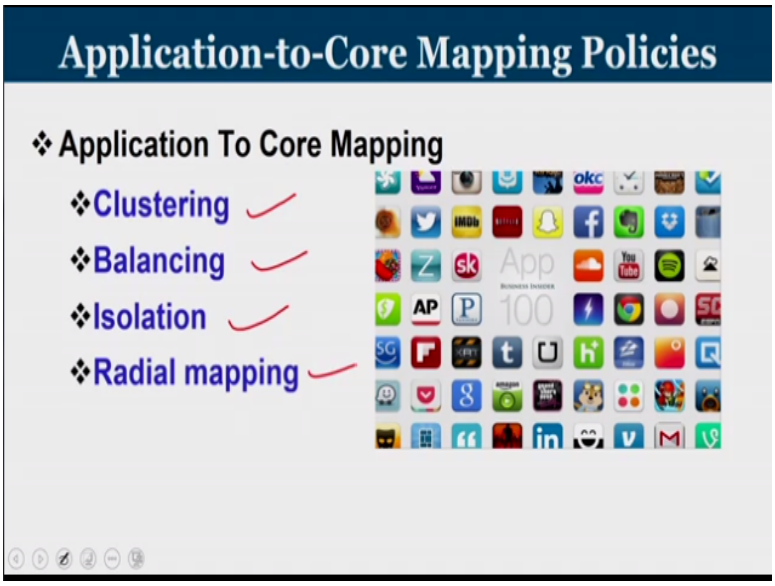
**(Refer Slide Time: 31:24)**




Our next topic of discussion for today's lecture is application to core mapping. We have seen that we are now our modern processer setups are called multi-cores called TCMPs tiled chip multi core processors. We have many cores are there and we have wide range of applications your whatsapp is running then facebook, then news feed for compiler program your media application many applications are going to run.

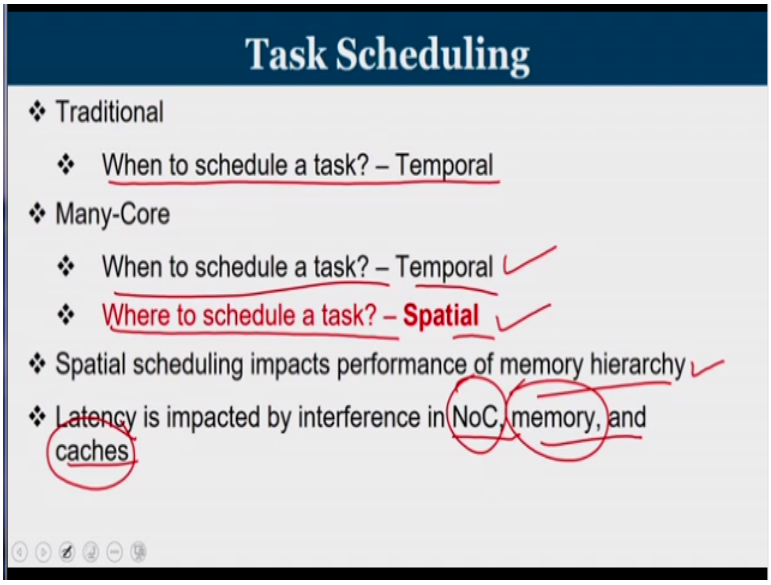
Now where an application has to be hosted this is the role played by an operating system, so how to map applications to core, that is called application to core mapping. It is generally done by the operating system which takes a core let us say this particular application has to run in this core. This blue application has to be run in this core, so each has it is own merits and demerits, operating system looks at many criteria in order to perform this mapping.

**(Refer Slide Time: 32:20)**




Application to core mapping, generally we have many applications and when you are going to map them we are going to learn 4 techniques that generally operating systems follow clustering, balancing, isolation and radial mapping.

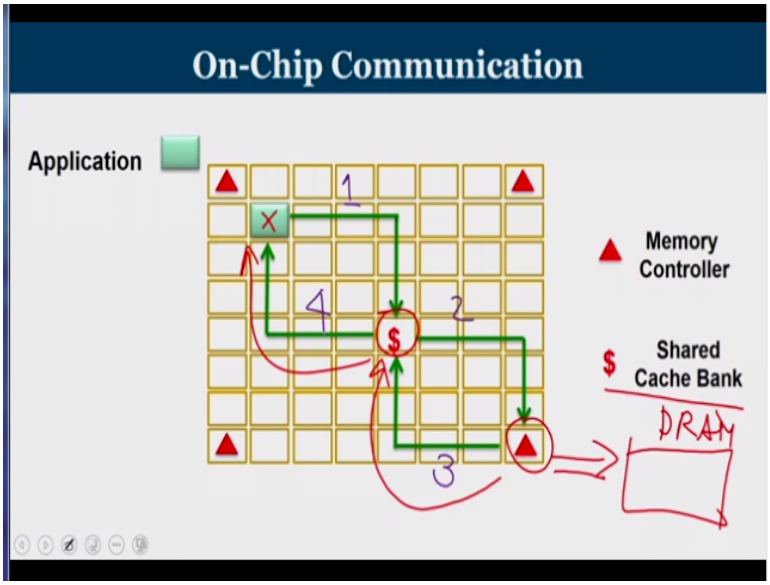
**(Refer Slide Time: 32:34)**




So what is basically task scheduling in a traditional microprocessor where there is only one CPU and multiple task to be done. There is only one question when to schedule a task that is called temporal scheduling. When you come to many core system where you have multiple processors as well as multiple task there are 2 aspect when will you schedule a task that is a temporal one and where to schedule a task in which core this particular task has to run that is called spatial scheduling.

So spatial scheduling impacts the performance of the memory hierarchy and latency is impacted by interference in network on-chip, memory as well as caches. We have understood these are shared resource by a processor you have caches, you have main memory and you have network on-chip all these are shared resource which are working together. So wherever your application is been running, the performance of network on-chip and memory hierarchy is always impacted by the scheduling that you do.

**(Refer Slide Time: 33:35)**




I draw your attention to an illustration of an on-chip communication, imagine that your application is going to run here. Now this particular application enter generally your memory controllers, we have learned about memory controllers during DRAM. This memory controllers are placed generally at the corner of the chip, so that the traffic also can be managed. Imagine that let us say this is the place where the L2 data is available.

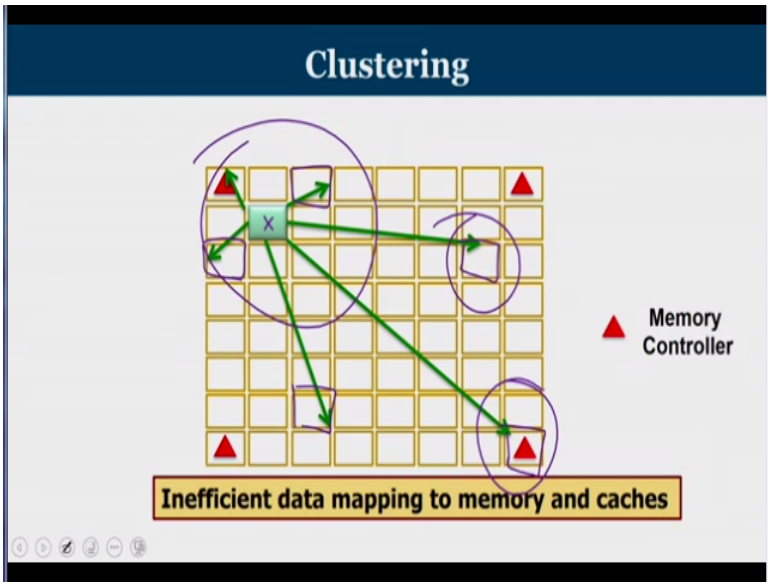
So any L1 cache miss will lead to a packet that is being generated, the packet will travel in XY direction or XY routing, reach this particular tie take the data and then come back. So an L1 cache miss at the source core will create a network on-chip packet which will travel through many routers, reach the destination tile where the L2 cache is been mapped. Take a block of data travel as multiple flits because your block of data cannot be accommodated in 1 flit.

So you have a sequence of body flits that will carry your cache block travel all the way return journey in XY routing and reach this source code, fill up the L1 cache block and then continue. Sometimes you can get you can get caches wherein it can be an L2 miss, so upon reaching the L2 cache only you will come to know it is a miss communication. So appropriately the memory address which is been spitted out and you find out based upon certain bits in the physical address which is the memory controller that will help in.

So based upon that you are going and reaching the memory controller you are remove off-chip, take from the this is the DRAM, go to the corresponding row, column, bank and all, get the data come back fill the L2 and then you reply the generate replay packet. So this happens as 1 if I write the sequence of operation a cache miss generated a miss request is going to the L2 tie upon checking it was found that it was an L2 cache miss.

So another request was generated to the appropriate main memory controller and the main memory controller is going to get this request convert into appropriate DRAM commands, take it off the chip into the DRAM, performs the operation, get the data, fill up the DRAM. And then you create a reply packet to the L2 tile, the L2 tile will fill up the block and then go to the L1 cache that is order in which on-chip communication happens.

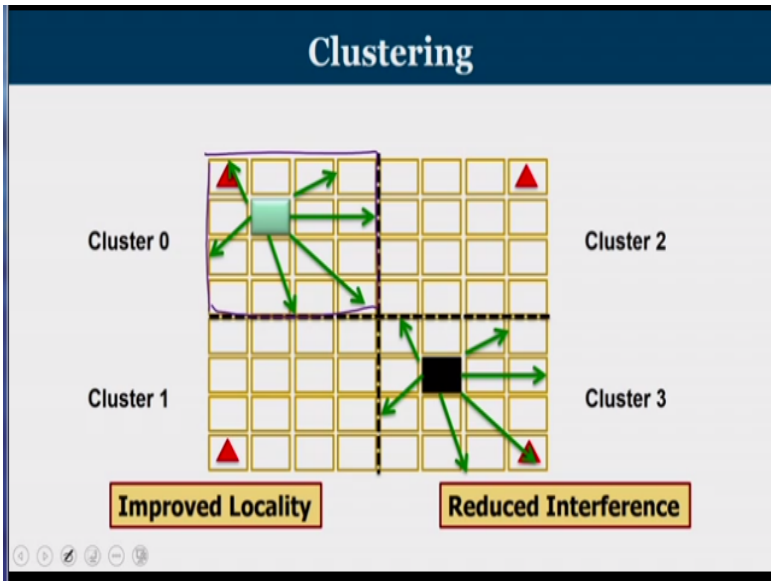
**(Refer Slide Time: 36:03)**



Now let us try to see what is a method by which a clustering happens, imagine that there is a program that is running in one of the application in a tile. Now let us say these are the tiles in which the L2 cache misses are been mapped or the L2 is located in these locations. So during the runtime of a program we get misses that are been generated to various portions of the chip and then reply packets are also been generated.

This is an inefficient data mapping because an application that runs somewhere in this sector of the chip has to get data from opposite side of the chip. So the request all the way have to travel long distances inside the chip it is going to impact this all time of the application.

**(Refer Slide Time: 36:58)**



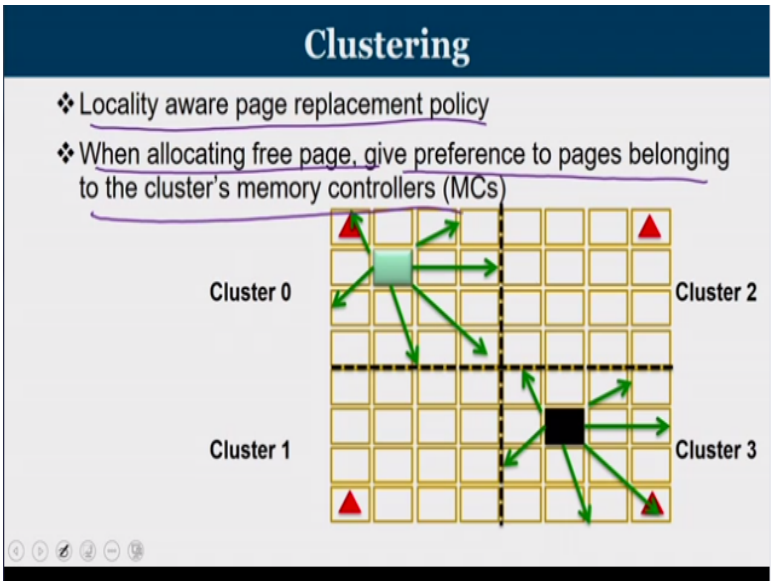

So what it does in clustering is let us say I logically divide the entire chip into 4 clusters based upon it is physical location. And if the operating system can assign addresses in such a way that everything what this particular application needs can be kept within the saying cluster itself. So packets from this particular core will never move outside this cluster, this method is known as clustering, so this will improve the locality.

Similarly there is some other application whose cache misses are clustered in cluster 3 itself, this will reduce inter cluster interference, reduced the interference. So clustering is a technique by which operating system carefully assigns the address because certain bits in the physical address

will determine which tile an address is mapped. We have learned it during the time we learned about cache memory is off TCMP system.

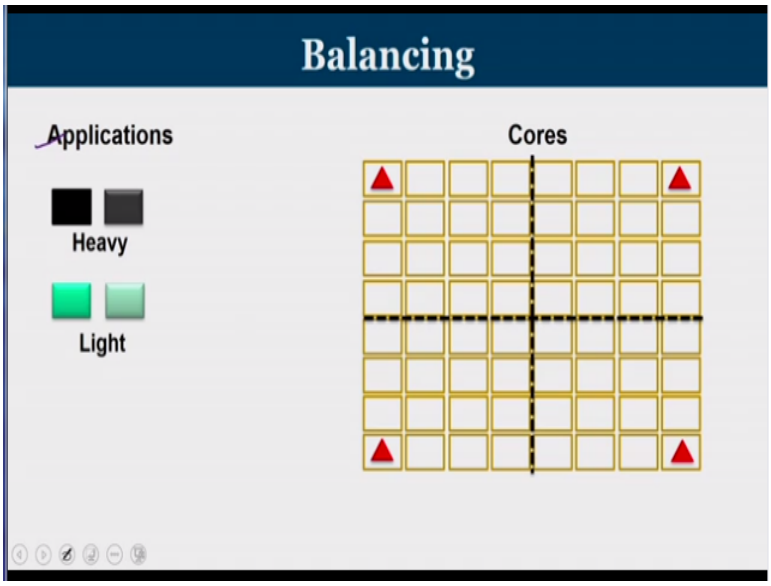
We will work out few problems also in the next tutorial session, you have to understand that it is operating system which assigns a physical address. And once the physical address is given few bits in the physical address will determine where the data is located when it is been placed on-chip. So if the operating system wants a data to locate in a specific tile then appropriately the bids are to be addressed, appropriately physical address has to be assigned, so one such method is clustering we should improve the locality.

**(Refer Slide Time: 38:29)**




Now we look into what essentially happens in this you are performing a locality aware page replacement policy. When allocating free page, give preference to pages belonging to the cluster's memory controller.

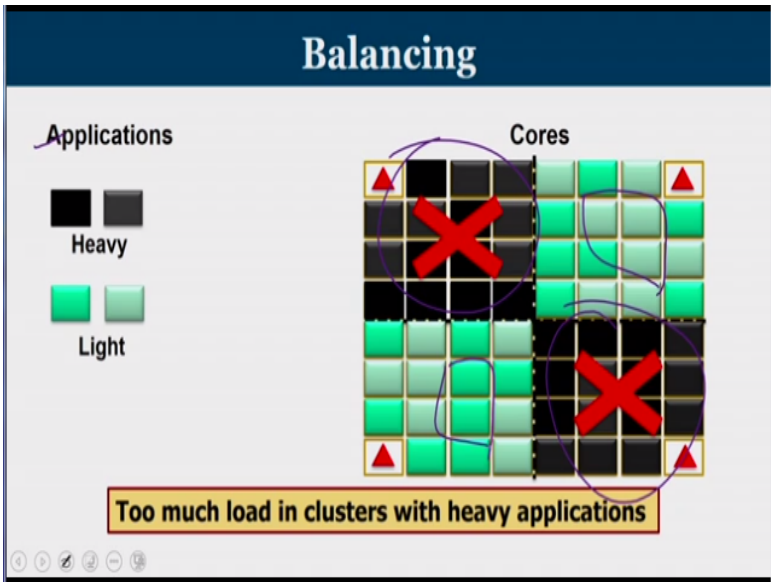
**(Refer Slide Time: 38:40)**




The next one is called balancing, we are going to have various applications, some applications are been termed heavy in NoC context and some are called as light in NoC context. Heavy applications are those which are going to generate more number of packets in a given unit time. The number of cache misses generated by these applications are much higher, so they are going to create more NoC packets.

So we call them as NoC heavy applications and there are some other applications which will generate cache misses only once in a while, so we call them as light applications.

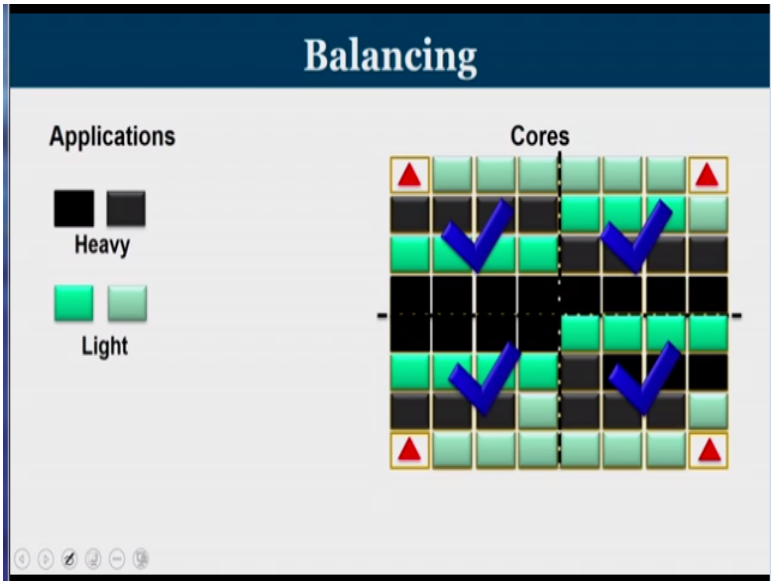
**(Refer Slide Time: 39:20)**



Now let us imagine the operating system is going to assign something like this where all the heavy applications come into 1 cluster and all the light applications move into another cluster, this is not a good approach. Because this 2 clusters, this cluster as well as this clusters is handling too much of a traffic, those routers are working more when compare to the other 2 routers, these 2 routers the routers are having less of job.

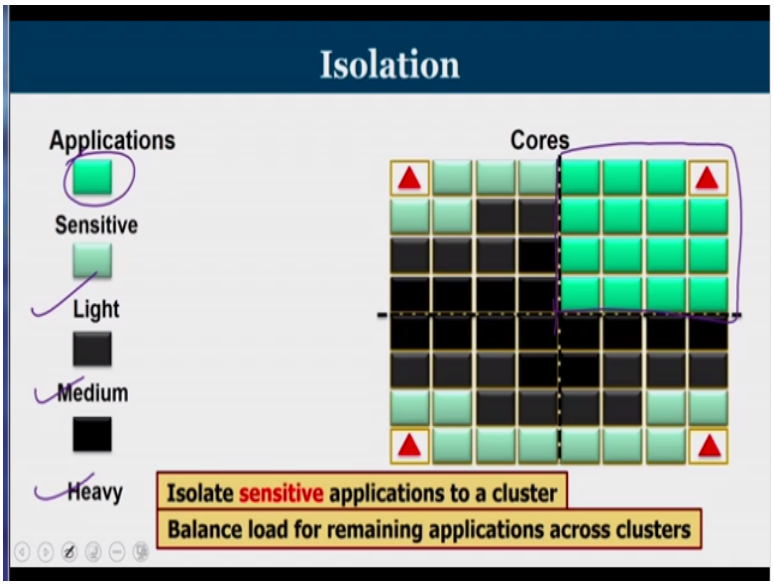
So too much load in certain clusters this lead to unimbalance in the amount of network activity that is happening. So for better life of chip it is always better to go for a balanced approach.

**(Refer Slide Time: 40:01)**




So the balancing can be something like this, every cluster will have both heavy applications as well as light applications.

**(Refer Slide Time: 40:07)**

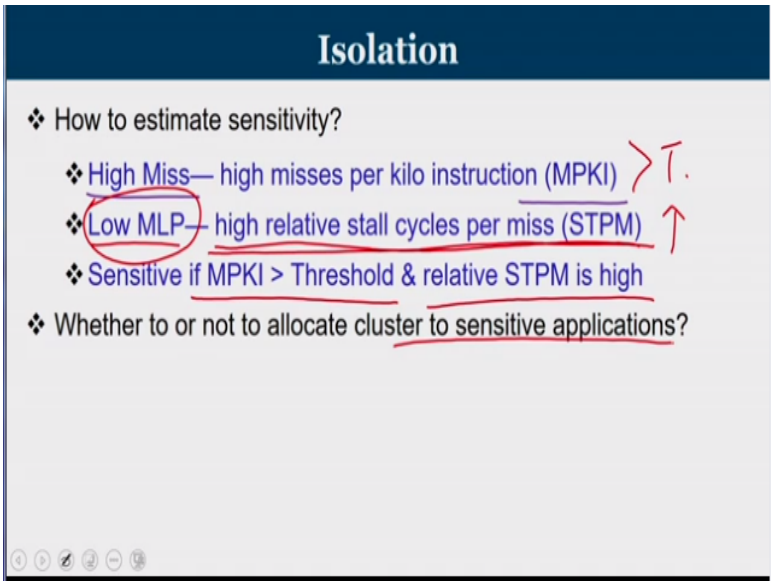



Now we move to the concept of isolation, here again we divide applications into medium, heavy and light certain applications are called as sensitive applications. The peculiarity of sensitive application is these applications upon generating certain misses the application is stalled. The rest of the application cannot continue until this misses are been returned, this is called sensitive application.

Generally it is preferred to keep all sensitive application restricted to 1 cluster, this is similar to the concept of let us say if you take the national capital in New Delhi. The VVIP officers, the residence of the ministers, prime minister they all are probably in a specific area through that maybe the general public travel is not supported. So that is what is called sensitive area or VVIP area because a traffic between them maybe the office of one of the minister to the other, these traffic movement should not go through the general public traffic.

So keeping a separate region for sensitive information or sensitive data that is what is known as isolation. And then the remaining portion I can fill up with both heavy and light application mix, so isolate sensitive applications to 1 cluster, balance load for remaining application across cluster.

**(Refer Slide Time: 41:32)**

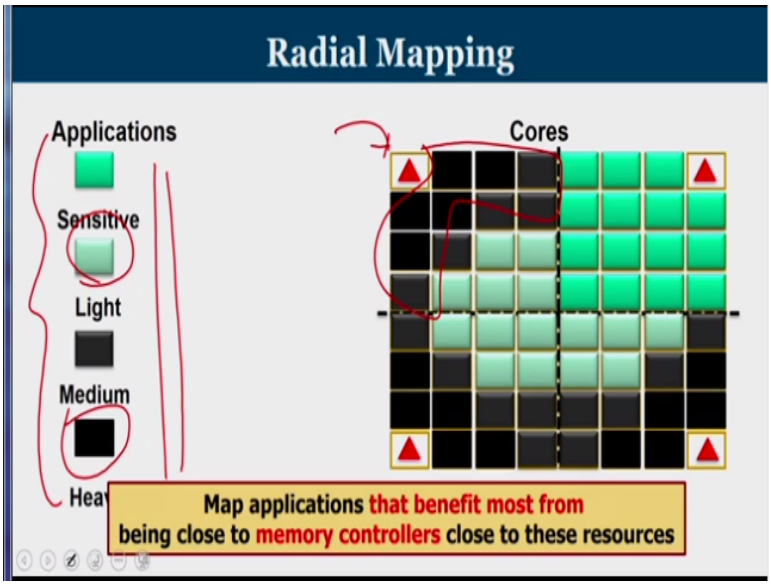



Now how will you find out this, how are you going to estimate the sensitivity, when you have high number of misses for an application and generally it is been measured as MPKI misses per kilo instruction. And low MLP, high relative stall cycles per miss, so there are multiple misses we have learned about non blocking caches. A cache memory can still facilitate hits even though it is processing a miss but there are can be a case where I cannot proceed to adjacent instruction.

Because I am relatively stalled, I am stalled because my previous miss is not at serviced. Such applications are said to have low MLP memory level parallelism. So it has high relative stall cycles STPM an application is said to be sensitive. If you MPKI is greater than a threshold and relative STPM is high, so this should be high and this is above a threshold value. Now the question is should be allocate cluster to sensitive applications still it is a debatable area.

These are all people are working on these topic to find out whether to allocate cluster to sensitive applications.

**(Refer Slide Time: 42:46)**




Now let us move into another technique of application to core mapping, it is called radial mapping. These classifications are already familiar to you, we divide applications into sensitive, light, medium and heavy. And then we have this cores wherein the sensitive application is been kept and we try to map application that benefit most from being closed memory controllers.

We know that if this is going to be the classification of application there is very high probability that the heavy application will move to memory controller more than that of light application. Heavy application mean they are more of L1 misses, the more L1 miss you have the more possibility that you may miss in L2 also. So try to keep the heavy applications as close as possible to memory controllers and then little bit medium application and then you have the light application.

So that these applications which are rated as heavy will get benefit of being closer to the memory controllers and you do for the rest of the controllers as well. So with this we come to the end of application to core mapping, so apart from the NoC router architecture that we discussed today. The last topic of discussion was the role played by operating system when you have multiple processing cores available and then you have lot of applications that is coming, how are you going to keep these applications across this cores.

And the application to core mapping is a very important area wherein active research is happening. This part is not part of any of the textbook it is published in recent research by some of the leading (()) (44:23) research group. The whole concept that I wanted to share by teaching you these concept it is a challenging task when it comes to spatial scheduling of applications. We learned about techniques like clustering, locality aware page mapping techniques.

We learned about balancing, isolation and radial mapping, more problem discussion would be there in the tutorial session that is part of this week, kindly refer to the tutorials and the solutions that is been discussed for the numerical questions that is been asked thank you.