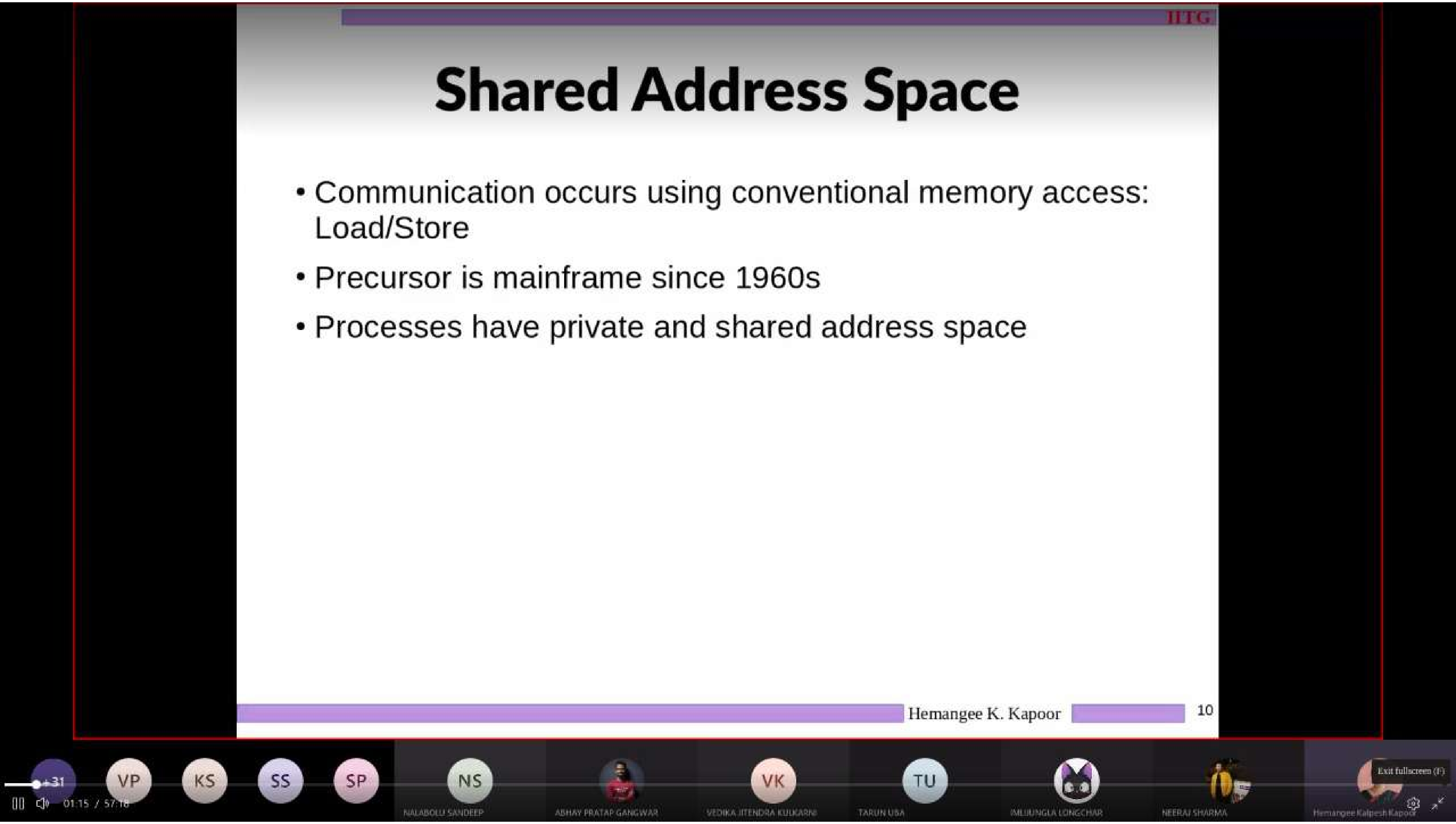
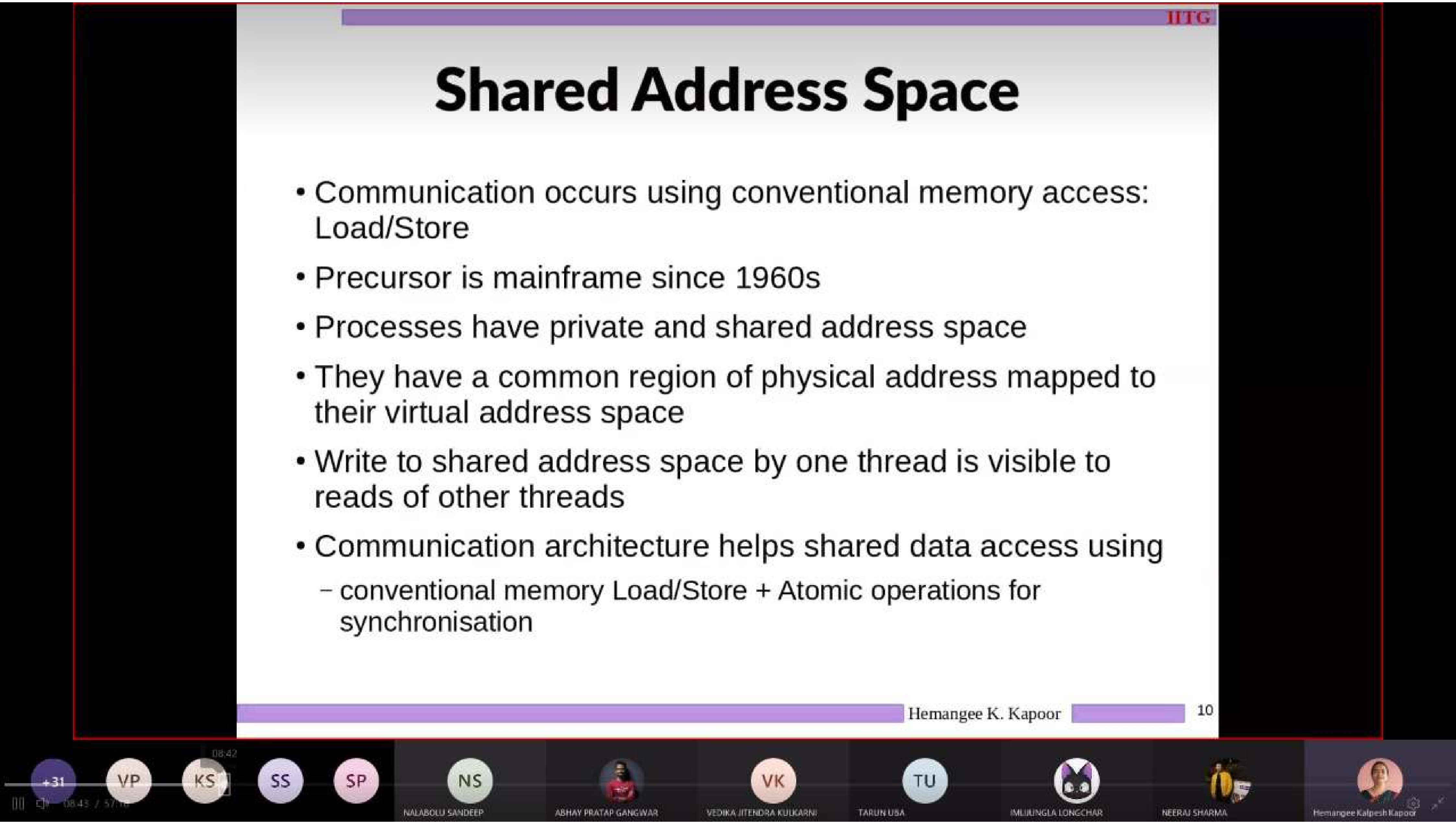


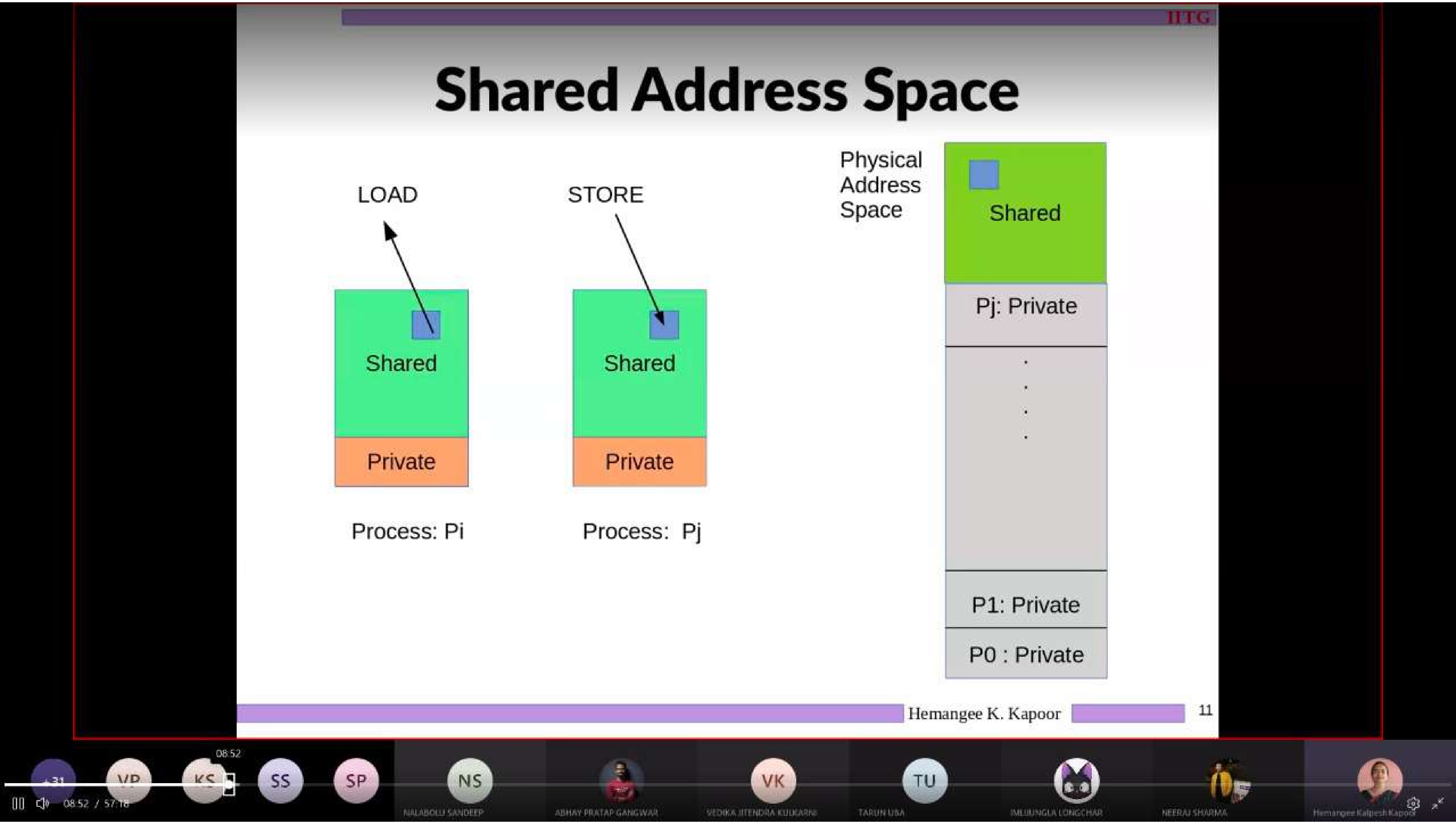
Shared Address Space

- Communication occurs using conventional memory access: Load/Store
- Precursor is mainframe since 1960s
- Processes have private and shared address space

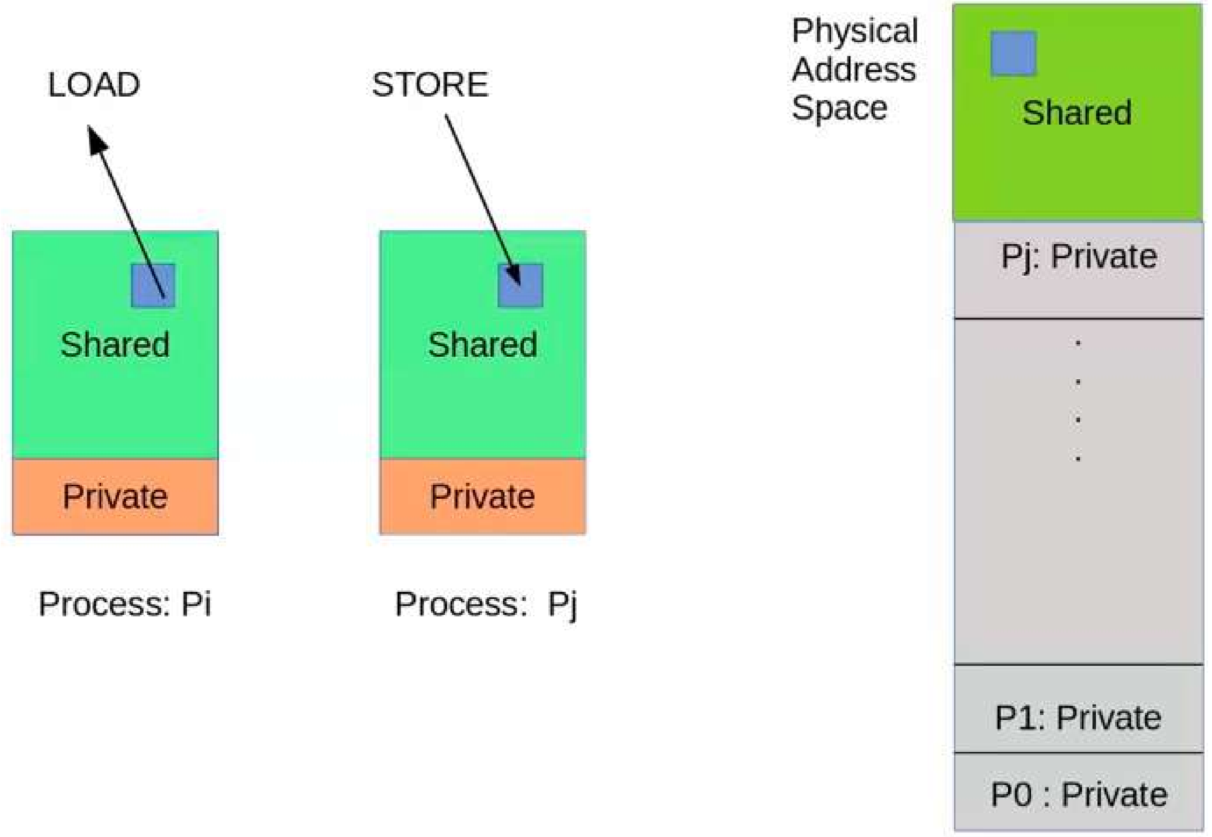


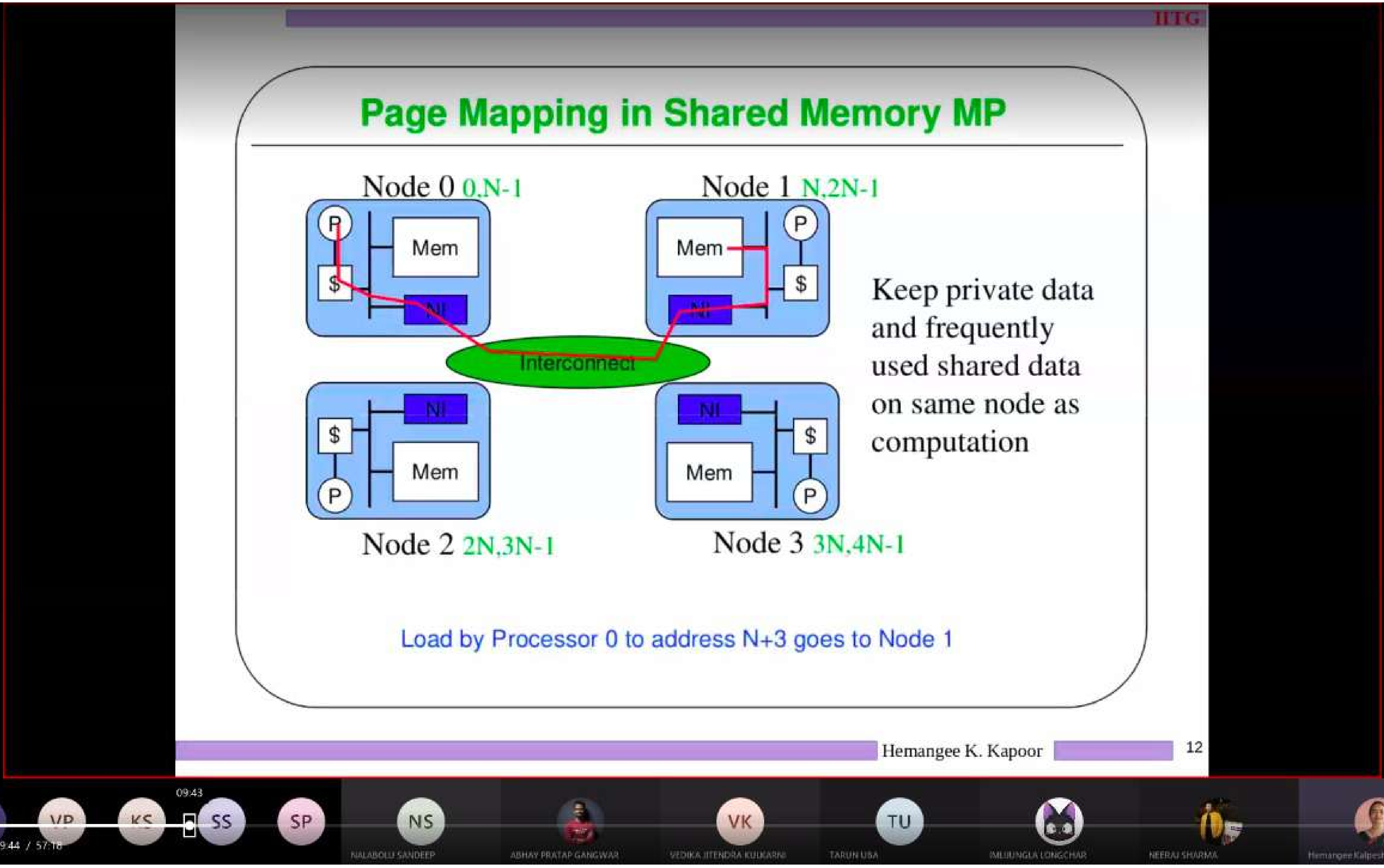
Shared Address Space

- Communication occurs using conventional memory access: Load/Store
- Precursor is mainframe since 1960s
- Processes have private and shared address space
- They have a common region of physical address mapped to their virtual address space
- Write to shared address space by one thread is visible to reads of other threads
- Communication architecture helps shared data access using
  - conventional memory Load/Store + Atomic operations for synchronisation

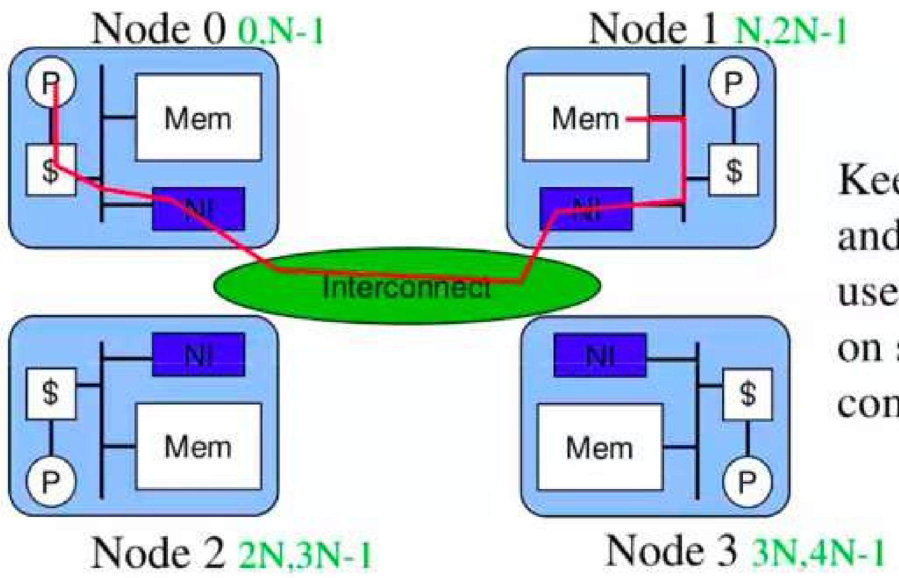


Shared Address Space





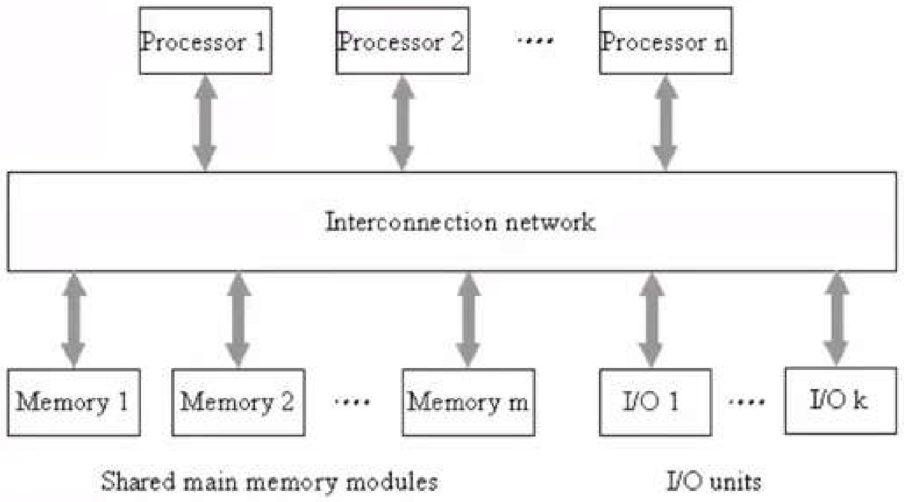
Page Mapping in Shared Memory MP



Keep private data and frequently used shared data on same node as computation

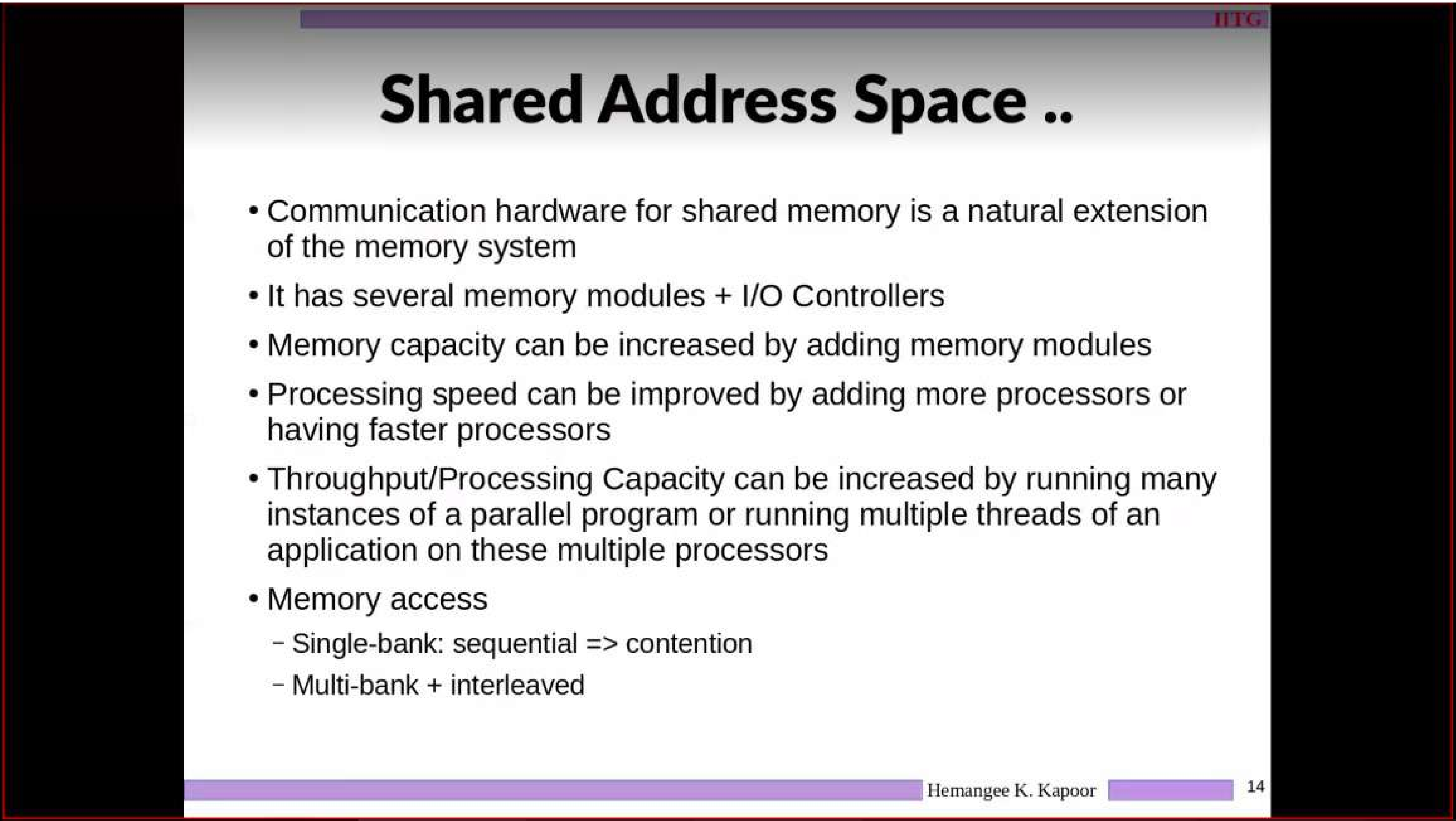
Load by Processor 0 to address N+3 goes to Node 1

Shared Address Space



+31 VP KS SS SP NS TU

NALABOLU SANDEEP
ABHAY PRATAP GANGWAR
VEDIKA JITENDRA KULKARNI
TARUN USA
IMLIJUNGLA LONGCHAR
NEERAJ SHARMA
Hemangee Kalpesh Kapoor

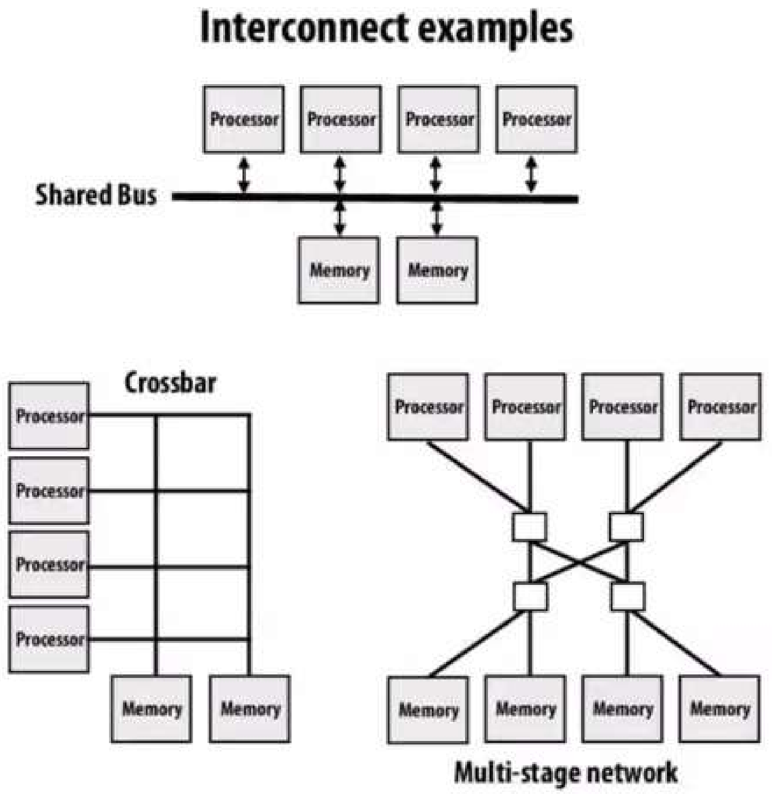


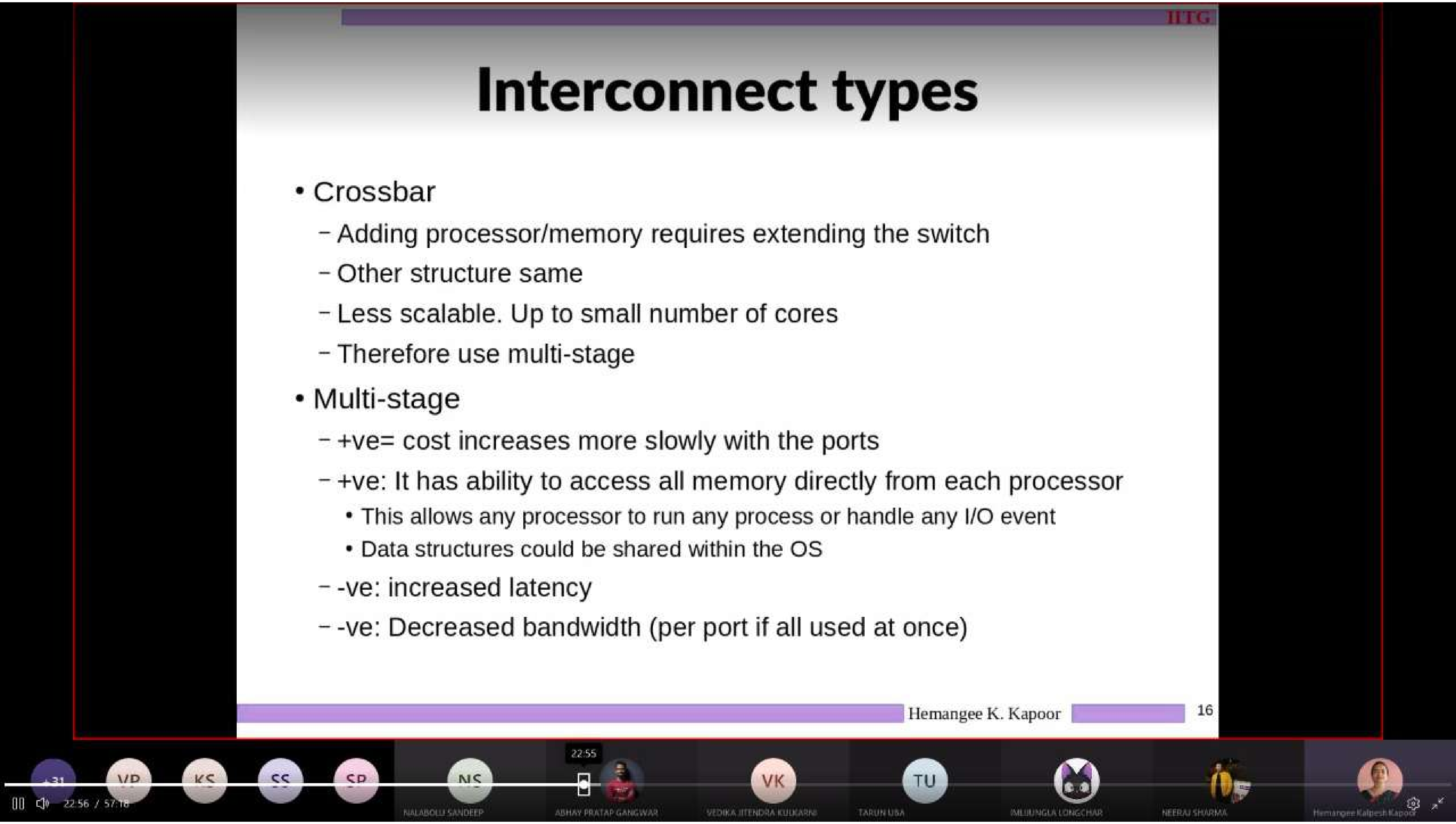
Shared Address Space ..

- Communication hardware for shared memory is a natural extension of the memory system
- It has several memory modules + I/O Controllers
- Memory capacity can be increased by adding memory modules
- Processing speed can be improved by adding more processors or having faster processors
- Throughput/Processing Capacity can be increased by running many instances of a parallel program or running multiple threads of an application on these multiple processors
- Memory access
  - Single-bank: sequential => contention
  - Multi-bank + interleaved

Press Esc to exit full screen

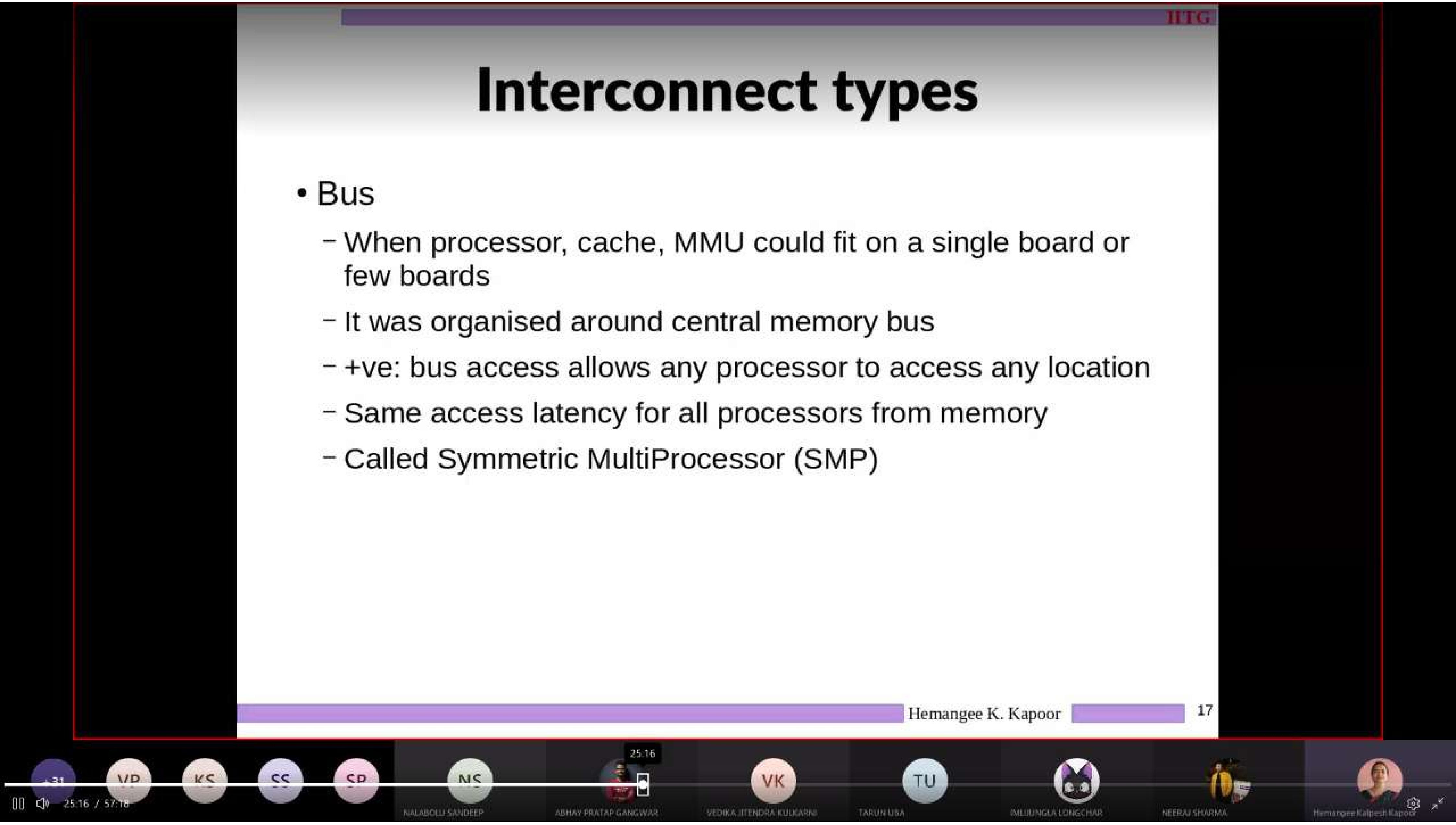
Interconnect types





Interconnect types

- Crossbar
  - Adding processor/memory requires extending the switch
  - Other structure same
  - Less scalable. Up to small number of cores
  - Therefore use multi-stage
- Multi-stage
  - +ve= cost increases more slowly with the ports
  - +ve: It has ability to access all memory directly from each processor
    - This allows any processor to run any process or handle any I/O event
    - Data structures could be shared within the OS
  - -ve: increased latency
  - -ve: Decreased bandwidth (per port if all used at once)

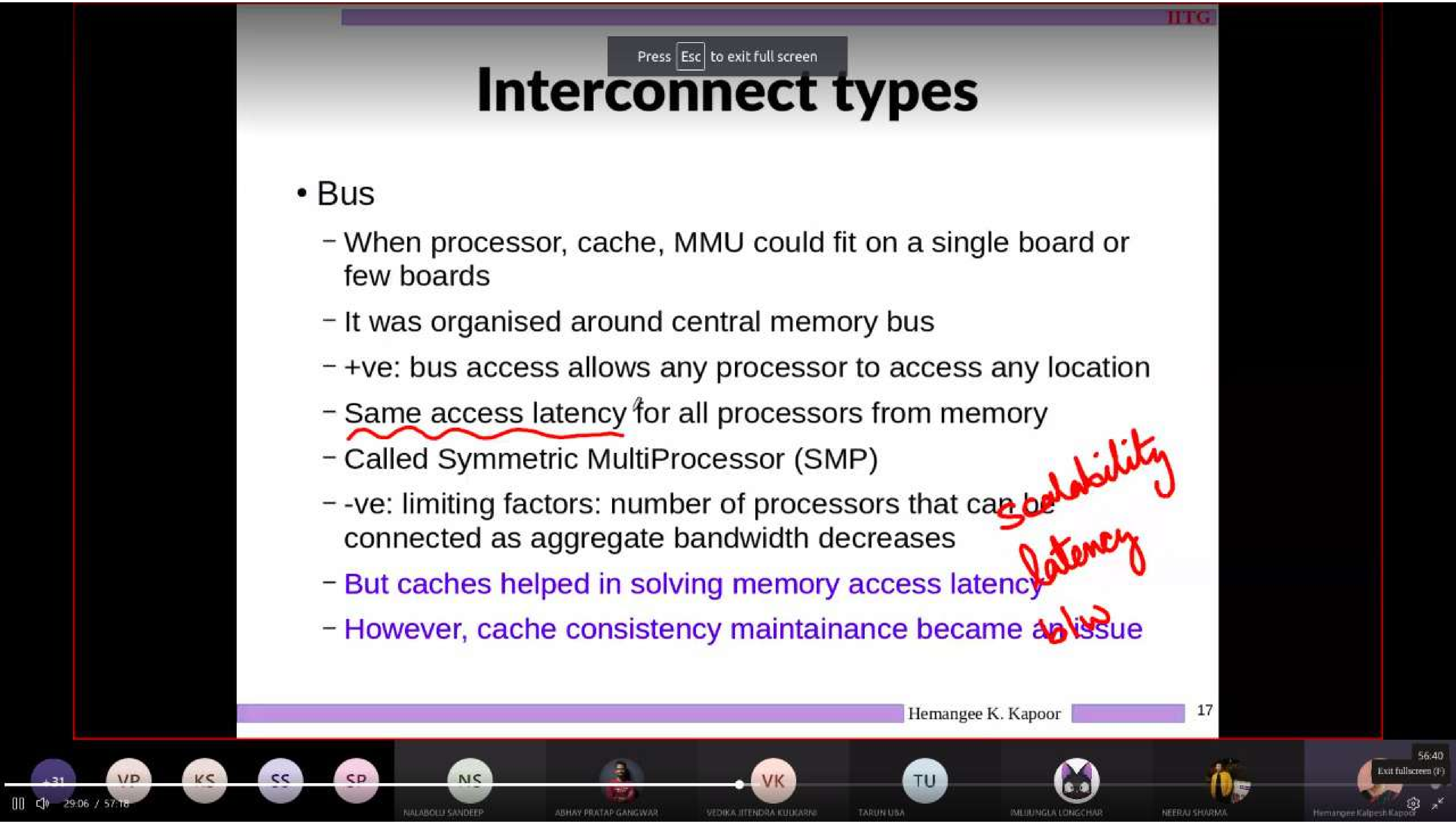


IITG

Interconnect types

- Bus
  - When processor, cache, MMU could fit on a single board or few boards
  - It was organised around central memory bus
  - +ve: bus access allows any processor to access any location
  - Same access latency for all processors from memory
  - Called Symmetric MultiProcessor (SMP)

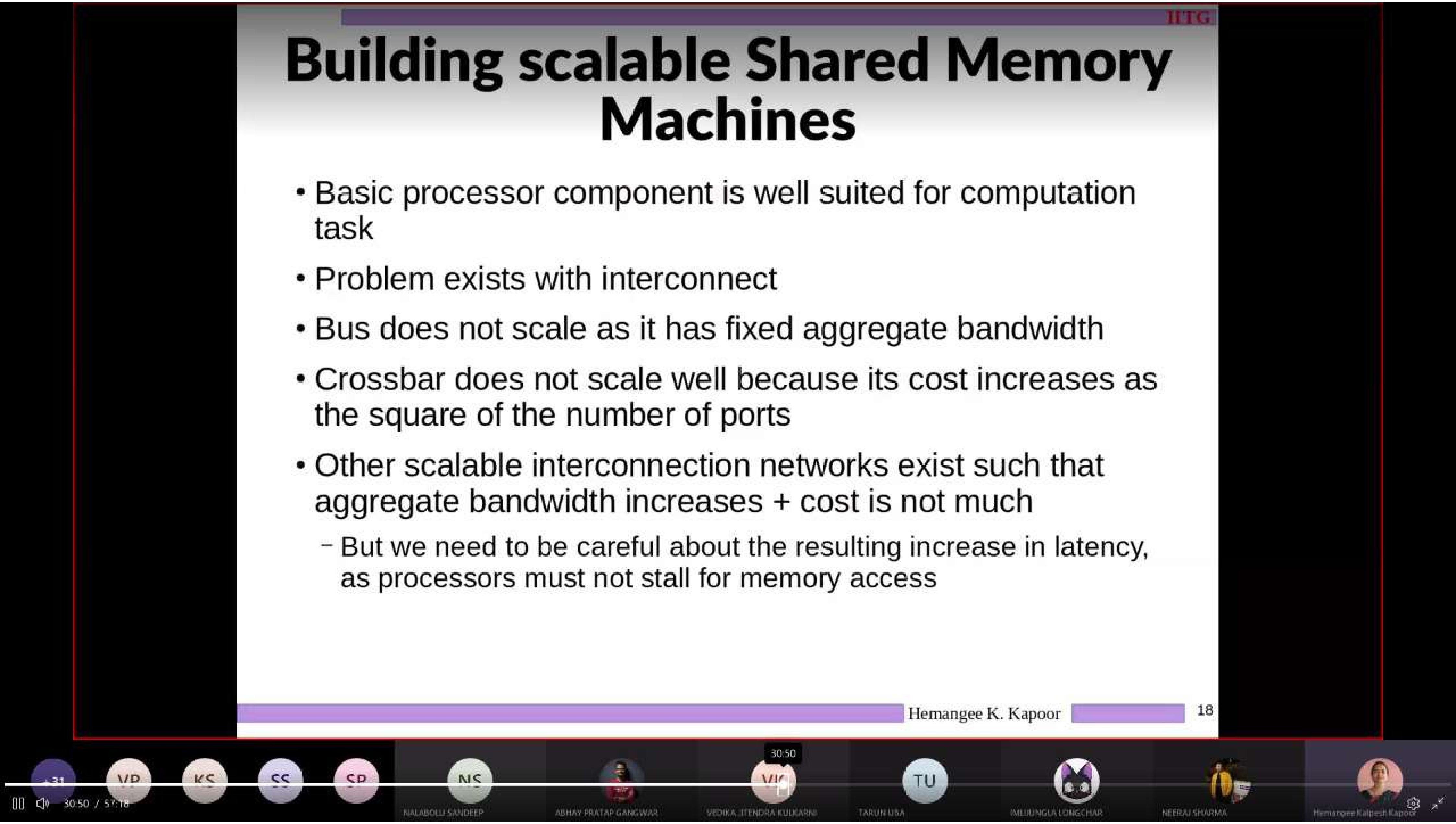
Hemangee K. Kapoor 17



Interconnect types

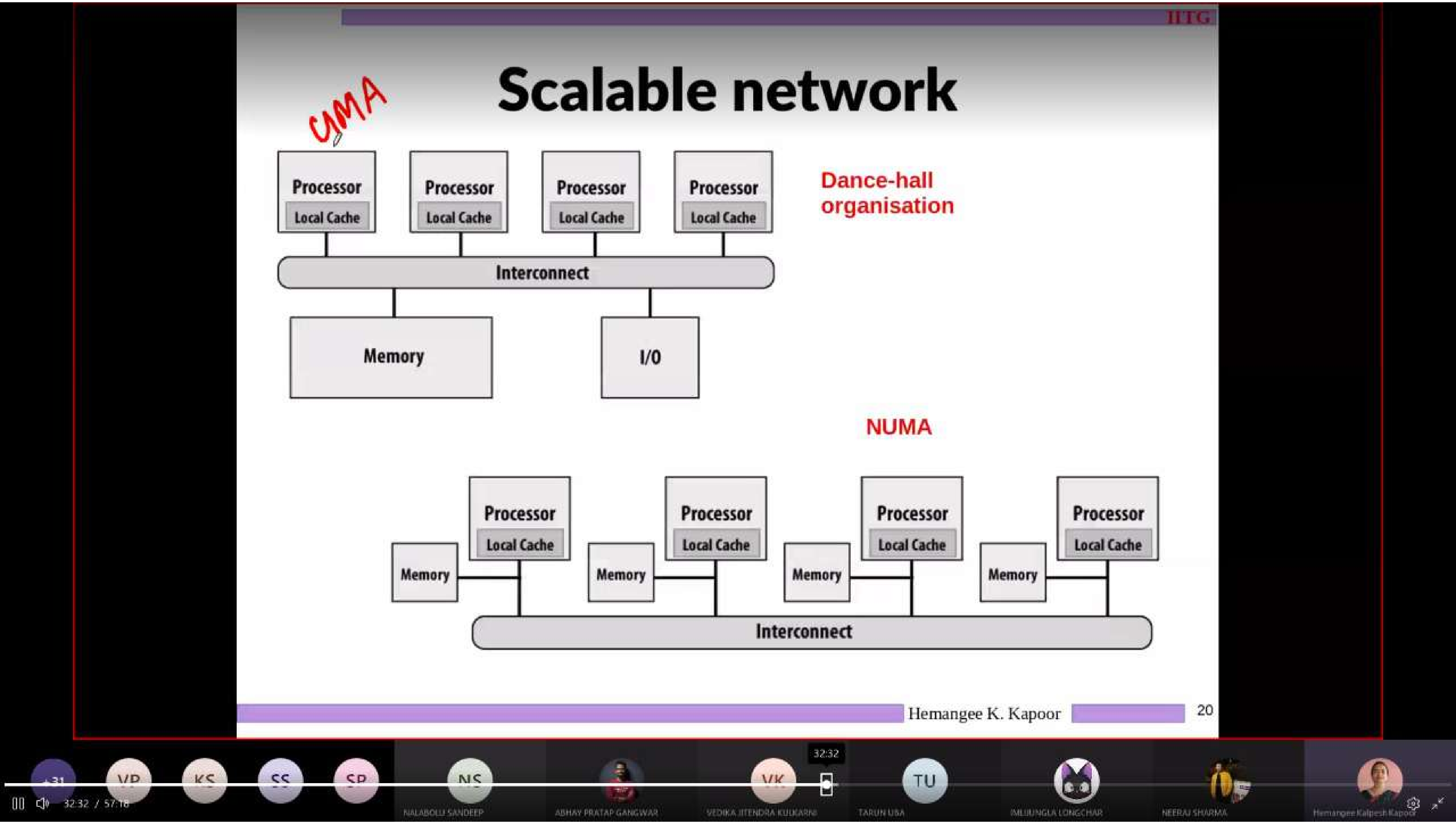
- Bus
  - When processor, cache, MMU could fit on a single board or few boards
  - It was organised around central memory bus
  - +ve: bus access allows any processor to access any location
  - Same access latency for all processors from memory
  - Called Symmetric MultiProcessor (SMP)
  - -ve: limiting factors: number of processors that can be connected as aggregate bandwidth decreases
  - But caches helped in solving memory access latency
  - However, cache consistency maintainance became an issue

Scalability
latency
b/w

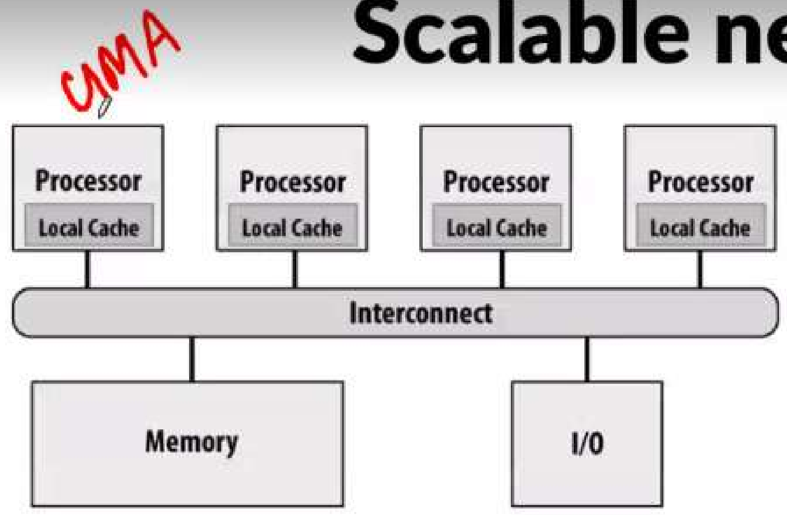


Building scalable Shared Memory Machines

- Basic processor component is well suited for computation task
- Problem exists with interconnect
- Bus does not scale as it has fixed aggregate bandwidth
- Crossbar does not scale well because its cost increases as the square of the number of ports
- Other scalable interconnection networks exist such that aggregate bandwidth increases + cost is not much
  - But we need to be careful about the resulting increase in latency, as processors must not stall for memory access

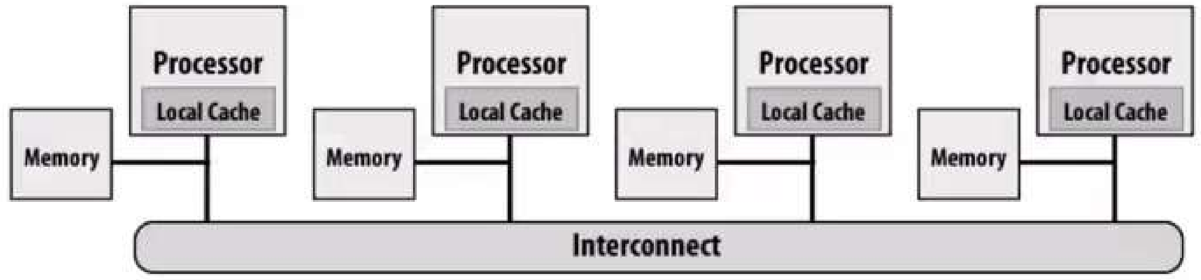


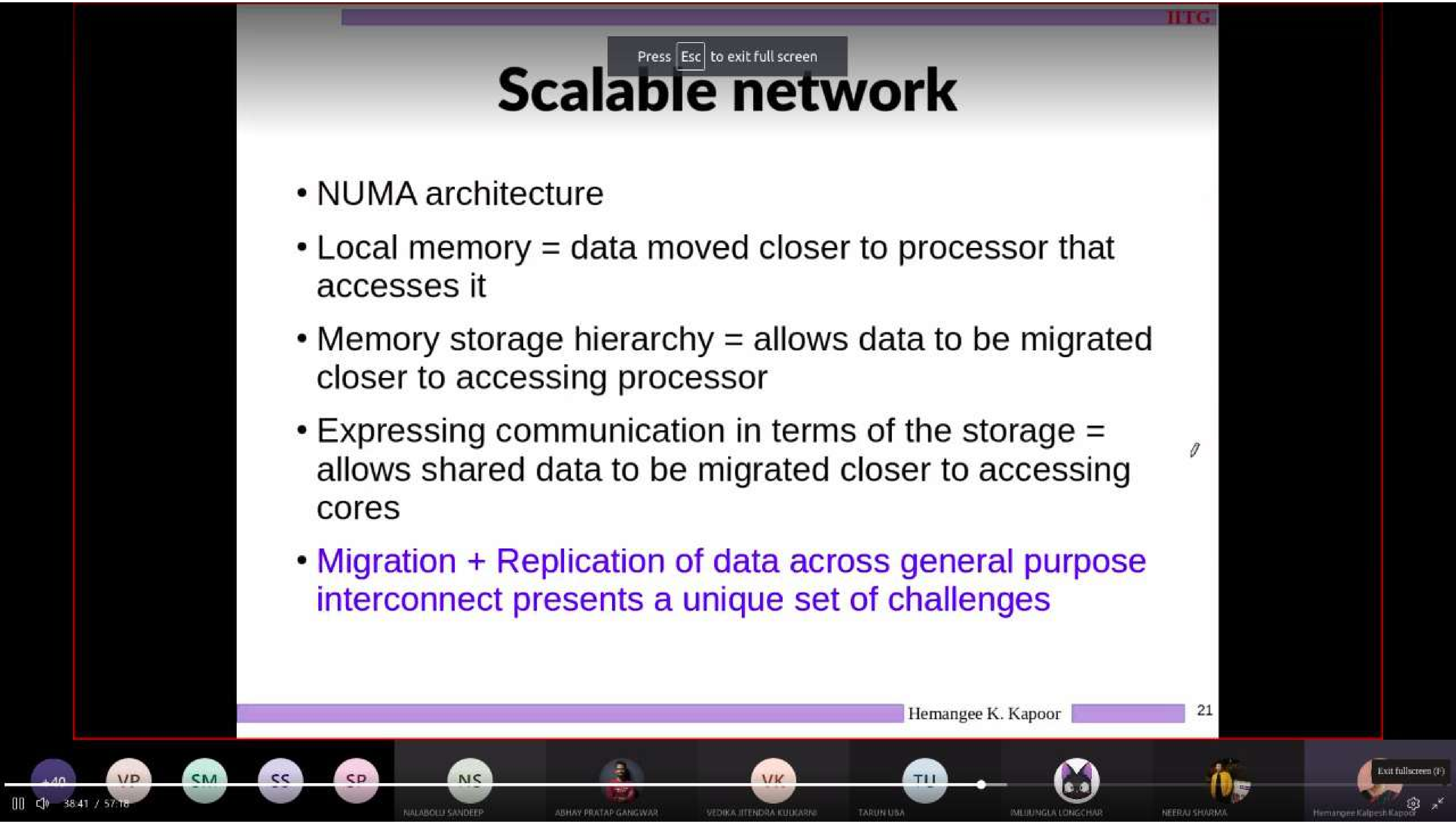
Scalable network



Dance-hall organisation

NUMA



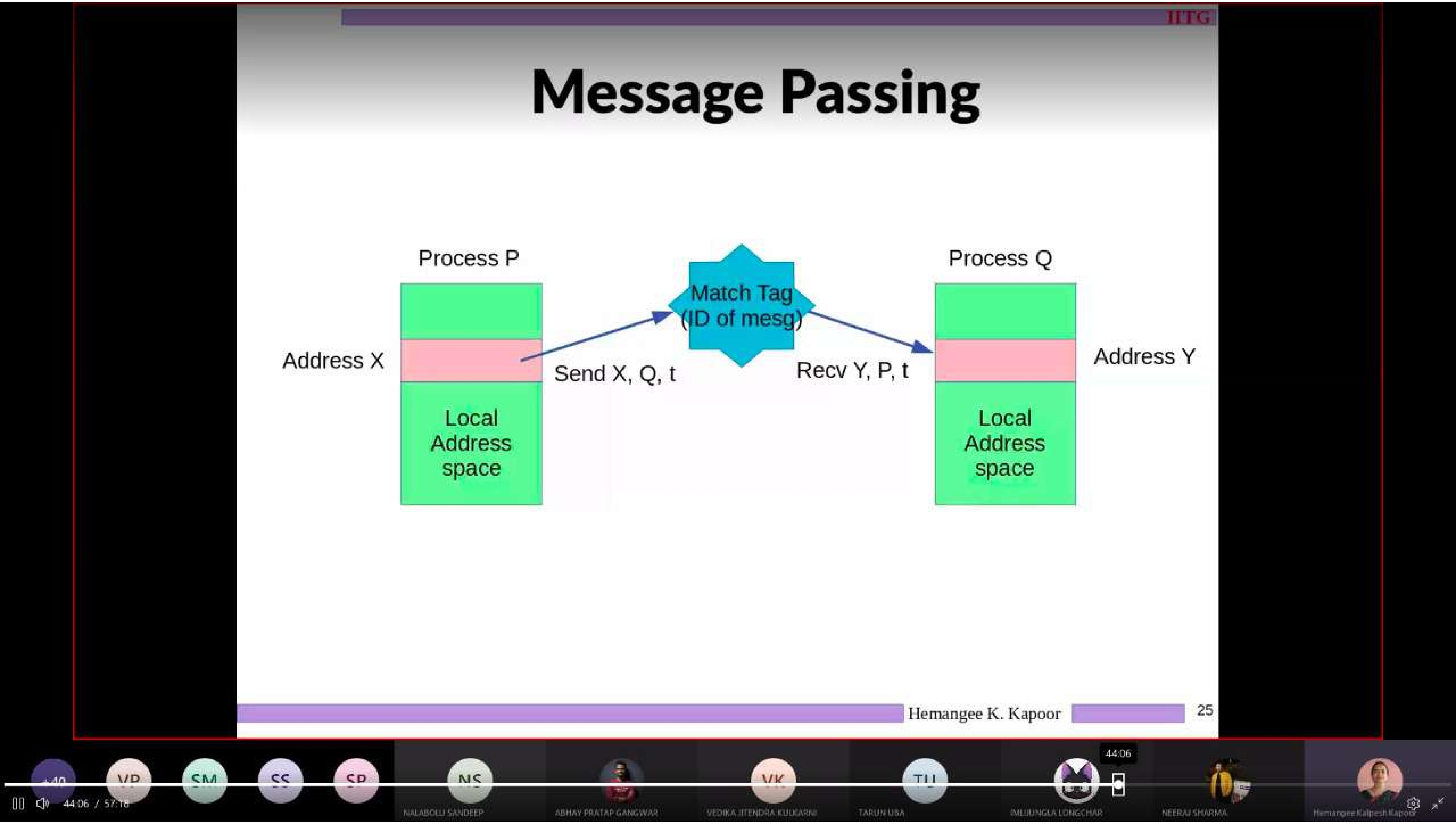


IITG

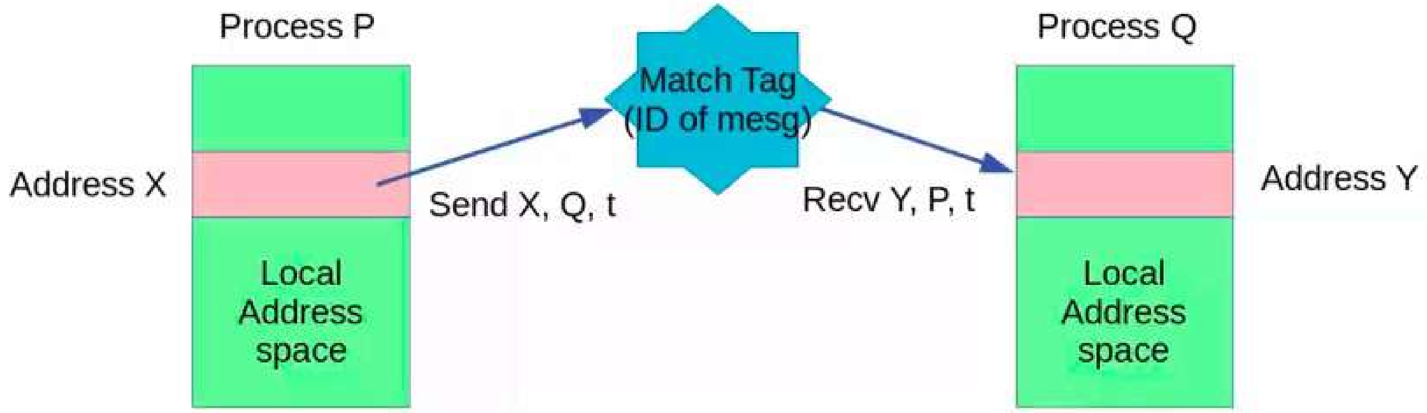
Press Esc to exit full screen

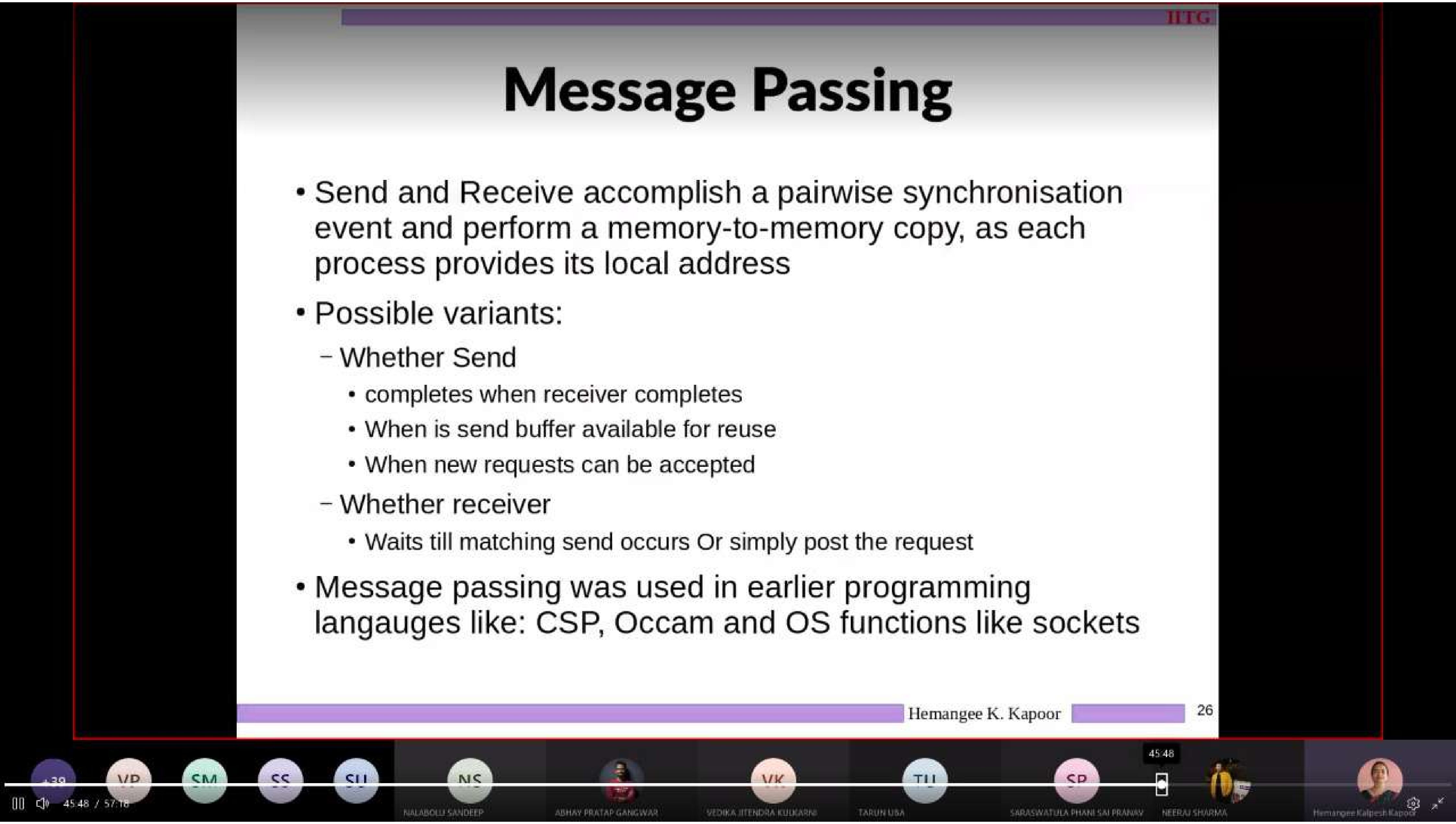
Scalable network

- NUMA architecture
- Local memory = data moved closer to processor that accesses it
- Memory storage hierarchy = allows data to be migrated closer to accessing processor
- Expressing communication in terms of the storage = allows shared data to be migrated closer to accessing cores
- Migration + Replication of data across general purpose interconnect presents a unique set of challenges



Message Passing

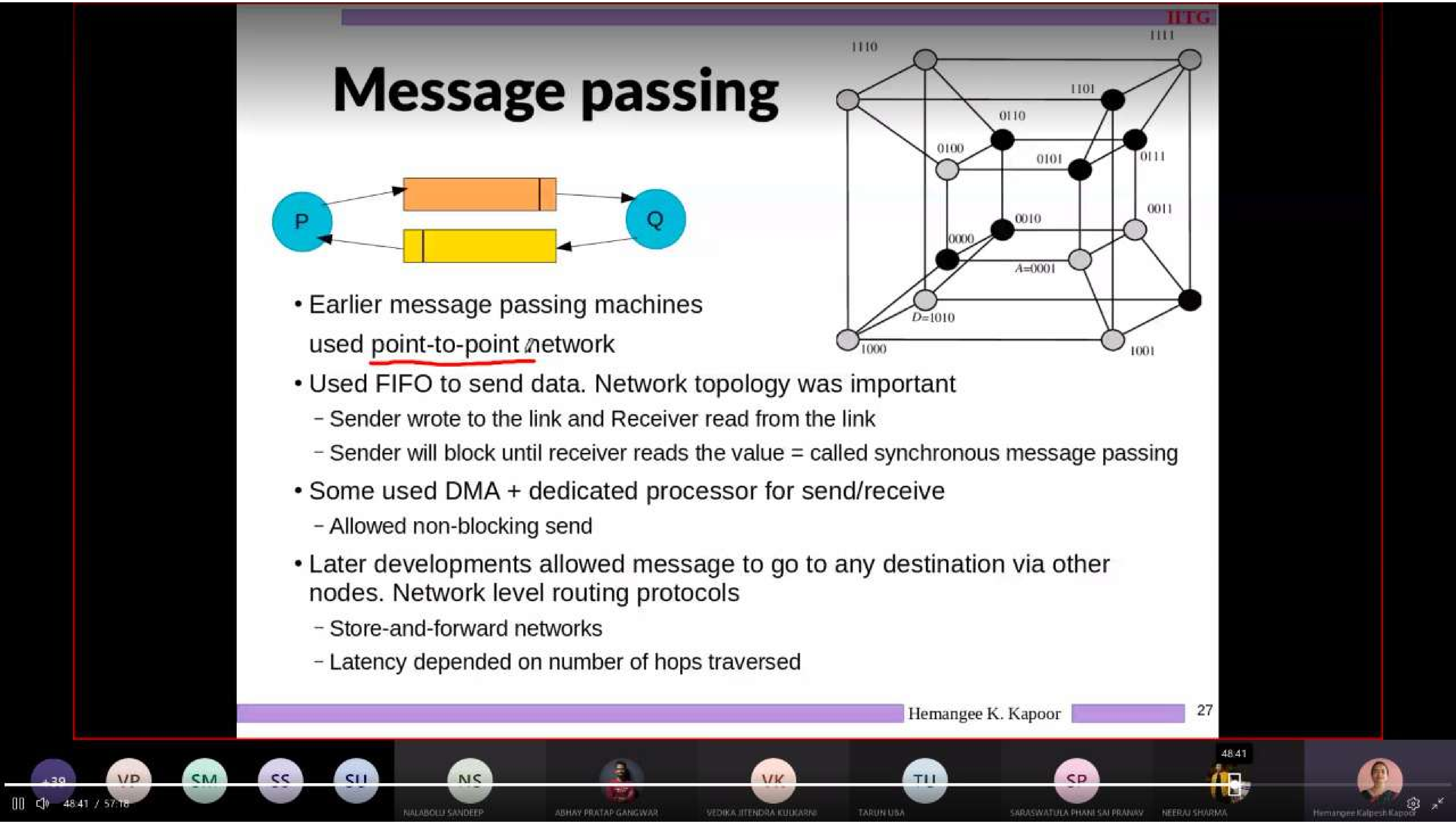




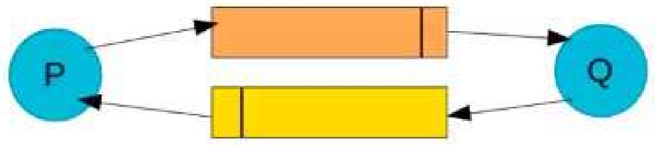
IITG

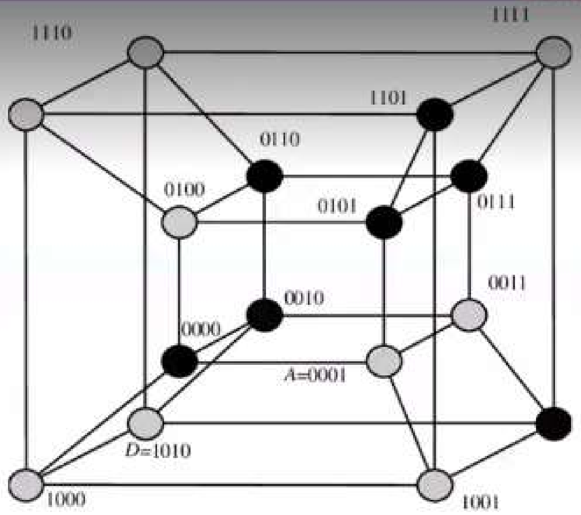
Message Passing

- Send and Receive accomplish a pairwise synchronisation event and perform a memory-to-memory copy, as each process provides its local address
- Possible variants:
  - Whether Send
    - completes when receiver completes
    - When is send buffer available for reuse
    - When new requests can be accepted
  - Whether receiver
    - Waits till matching send occurs Or simply post the request
- Message passing was used in earlier programming langauges like: CSP, Occam and OS functions like sockets



Message passing

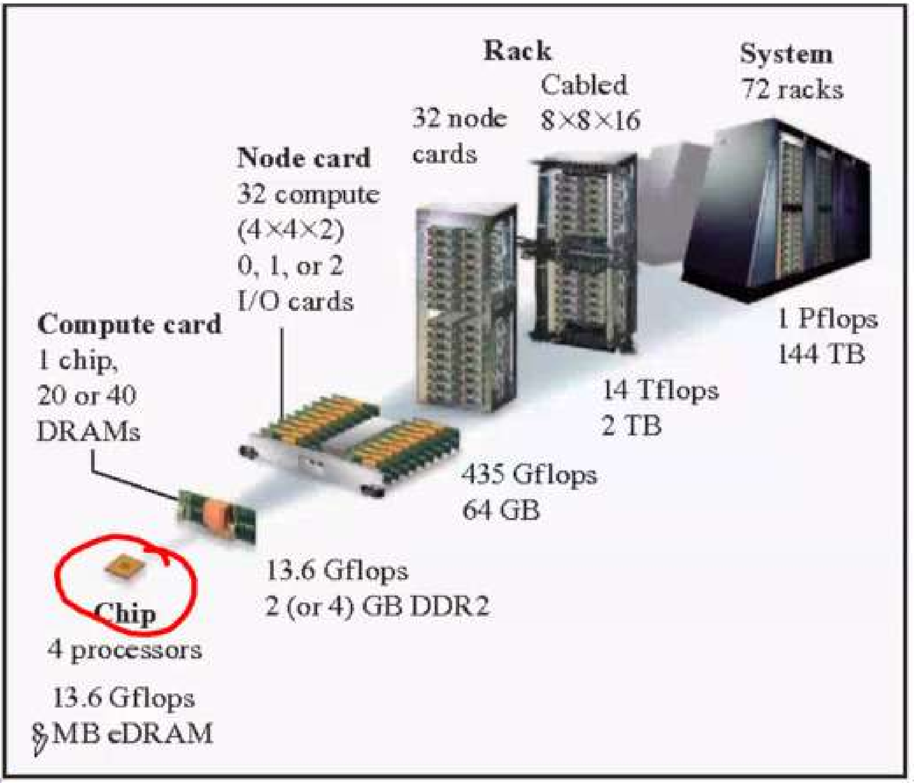




- Earlier message passing machines used point-to-point network
- Used FIFO to send data. Network topology was important
  - Sender wrote to the link and Receiver read from the link
  - Sender will block until receiver reads the value = called synchronous message passing
- Some used DMA + dedicated processor for send/receive
  - Allowed non-blocking send
- Later developments allowed message to go to any destination via other nodes. Network level routing protocols
  - Store-and-forward networks
  - Latency depended on number of hops traversed

Press Esc to exit full screen

IBM Blue Gene/P supercomputer



+39 VP SM SS SU NS NALABOLU SANDEEP ABHAY PRATAP GANGWAR VK VEDIKA JITENDRA KULKARNI TU TARUN UBA SP SARASWATULA PHANI SAI PRANAV NEERAJ SHARMA Hemangee Kalpesh Kapoor

54:30 / 57:18

Exit fullscreen (F)