2. Threading a.

Would it make sense to limit the number of threads in a server process?

Yes, for two reasons. First, threads require memory for setting up their own

private stack. Consequently, having many threads may consume too much

memory for the server to work properly. Another, more serious reason, is that,

to an operating system, independent threads tend to operate in a chaotic manner.

In a virtual memory system it may be difficult to build a relatively stable

working set, resulting in many page faults and thus I/O. Having many threads

may thus lead to a performance degradation resulting from page thrashing.

Even in those cases where everything fits into memory, we may easily see that

memory is accessed following a chaotic pattern rendering caches useless.

Again, performance may degrade in comparison to the single-threaded case.

b. Constructing a concurrent server by spawning a process has some advantages and disadvantages compared to multithreaded servers. Mention a few.

An important advantage is that separate processes are protected against

each other, which may prove to be necessary as in the case of a superserver

handling completely independent services. On the other hand, process spawning

is a relatively costly operation that can be saved when using multithreaded

servers. Also, if processes do need to communicate, then using threads is much

cheaper as in many cases we can avoid having the kernel implement the communication.

3. Resilience

a. **What components in a computer system do we know how to make resilient, and what technique is used? Name 3 such components.**

**Storage can be made resilient by implementing RAID. There are many RAID levels,**

Such as RAID 0-RAID5 and then some special kind of RAID which is RAID-6 or Netapp

Implements RAID-DP (dual parity).

RAID-0 -- Disk striping : RAID 0 splits data across any number of disks allowing higher data throughput

**RAID 1 (Disk Mirroring):** RAID 1 writes and reads identical data to pairs of drives.

**RAID 5 (Striping with parity):** RAID 5 stripes data blocks across multiple disks like RAID 0, however, it also stores parity information (Small amount of data that can accurately describe larger amounts of data) which is used to recover the data in case of disk failure.

### Power supply : By adding more than one power supplying module.

### Memory : BY adding ECC into memory.

### Processor: By adding Extra hidden core. IBM added extra core into their supercomputer.

**b. RAID (redundant array of inexpensive disks) is an example what type of recovery mechanism? Briefly describe 3 different levels of RAID.**

RAID is a disaster recovery technique.

RAID-0 -- Disk striping : RAID 0 splits data across any number of disks allowing higher data throughput

**RAID 1 (Disk Mirroring):** RAID 1 writes and reads identical data to pairs of drives.

**RAID 5 (Striping with parity):** RAID 5 stripes data blocks across multiple disks like RAID 0, however, it also stores parity information (Small amount of data that can accurately describe larger amounts of data) which is used to recover the data in case of disk failure.

**4. Laws**

**a. Describe Moore's Law.**

**Moore's law** is the observation that the number of transistors in a dense integrated circuit doubles about every two years. The observation is named after Gordon Moore, the co-founder of Fairchild Semiconductor and CEO of Intel, whose 1965 paper described a doubling every year in the number of components per integrated circuitand projected this rate of growth would continue for at least another decade. In 1975, looking forward to the next decade,he revised the forecast to doubling every two years. The period is often quoted as 18 months because of a prediction by Intel executive David House (being a combination of the effect of more transistors and the transistors being faster).

b. Describe Amdahl's Law.

* **Amdahl's Law** states that potential program speedup is defined by the fraction of code (P) that can be parallelized:

|  |
| --- |
| **1**  **speedup = --------**  **1 - P** |

* If none of the code can be parallelized, P = 0 and the speedup = 1 (no speedup).
* If all of the code is parallelized, P = 1 and the speedup is infinite (in theory).
* If 50% of the code can be parallelized, maximum speedup = 2, meaning the code will run twice as fast.
* Introducing the number of processors performing the parallel fraction of work, the relationship can be modelled by:

|  |
| --- |
| **1**  **speedup = ------------**  **P + S**  **---**  **N** |

where P = parallel fraction, N = number of processors and S = serial fraction.

**5. Programming models**

**a. Describe what shared address space and message passing is, and the difference between them? In what environments would one be used over the other?**

Shared memory models communicate by reading/writing to shared memory blocks, which are protected by semaphores or similar.

Message passing models (Erlang, for example) do not have any shared state; all synchronization and communication is done by exchanging messages.

1.In shared memory model, memory is shared by cooperating processes, which can exchange information by reading and writing data but in message passing communication takes place by means of messages exchanged between the cooperating processes.

2.Shared memory helps run processes concurrently but message passing cannot.

3.Message passing facility has two operations: send (message) and receive (message). The process of which has fixed or variable size.

4.Message passing is useful for exchanging smaller amounts of data, because no conflicts need be avoided. Message passing is also easier to implement than is shared memory for inter-processes communication.

5.In shared-memory systems, system calls are required only to establish shared-memory regions. Once shared memory is established, all accesses are treated as routine memory accesses, and no assistance from the kernel is required.

**In what environments would one be used over the other?**

In any systems where there is some requirement of transferring data and commands to a remote machine, message passing is used. Such as NFS or SMB.

For all the local processing which is bounded in one physical computer shared memory is preferred as it is faster than message passing although difficult to implement.

**6. Distributed Systems**

**a. Name some advantages of distributed systems over centralized systems.**

Advantages of distributed systems over centralized systems.

|  |  |
| --- | --- |
| **Item** | **Description** |
| Economics | Microprocessors offer a better price/performance than mainframes |
| Speed | A distributed system may have more total computing power than a mainframe |
| Inherent distribution | Some applications involve spatially separated machines |
| Reliability | If one machine crashes, the system as a whole can still survive |
| Incremental growth | Computing power can be added in small increments |

**b. Define a cluster of computers. Define a supercomputer. What is the difference between clusters and supercomputers?**

A **computer cluster** is a set of loosely or tightly connected **computers** that work together so that, in many respects, they can be viewed as a single system. Unlike grid **computers**, **computer clusters** have each node set to perform the same task, controlled and scheduled by software.

 A **supercomputer** is the fastest computer in the world that can process a significant amount of data very quickly. The computing Performance of a "supercomputer" is measured very high as compared to a general purpose computer. The computing Performance of a supercomputer is measured in FLOPS (that is floating-point operations per second) instead of MIPS. The **supercomputer** consists of tens of thousands of processors which can perform billions and trillions of calculations per second, or you can say that supercomputers can deliver up to nearly a hundred quadrillions of FLOPS.

**What is the difference between clusters and supercomputers?**

In a cluster, each machine is largely independent of the others in terms of memory, disk, etc. They are interconnected using some variation on normal networking. The cluster exists mostly in the mind of the programmer and how s/he chooses to distribute the work.

In a Massively Parallel Processor, there really is only one machine with thousands of CPUs tightly interconnected. MPPs have exotic memory architectures to allow extremely high speed exchange of intermediate results with neighbouring processors.

The major variants are SIMD (Single Instruction, Multiple Data) and MIMD (Multiple Instruction, Multiple Data). In a SIMD system, every processor is executing the same instruction at the same time, only on different bits of memory. Essentially, there is only one Program Counter. In a MIMD machine, each CPU has it's own PC.

**c. Define grid computing. Define cloud computing. What is the difference between grids and clouds?**

Grid computing is a processor architecture that combines computer resources from various domains to reach a main objective. In grid computing, the computers on the network can work on a task together, thus functioning as a supercomputer.

Typically, a grid works on various tasks within a network, but it is also capable of working on specialized applications. It is designed to solve problems that are too big for a supercomputer while maintaining the flexibility to process numerous smaller problems. Computing grids deliver a multiuser infrastructure that accommodates the discontinuous demands of large information processing.

**What is the difference between grids and clouds?**

|  |  |
| --- | --- |
| **GRID** | **CLOUD** |
| It is for application Oriented | It is for service oriented. |
| The resources are distributed among different computing units for processing a single task. | The computing resources are managed centrally and are placed over multiple servers in clusters. |
| Grids are generally owned and managed by an organization within its premises. | The cloud servers are owned by infrastructure providers are placed in physical disparate locations. |
| It operates within corporate network. | It can also be accessed through the internet. |
| It provides a shared pool of computing resource on an as needed basis. | It involves dealing with common problem using varying number of computing resources. |
| It is a collection of interconnected computes and networks that can be called for large scale computer task. | More than one computer coordinates to resolve the problem together. |

**7. Energy**

**a. Why power consumption is critical to datacentre operations?**

Power consumption is critical in DCs because as time progresses hardware costs keeps dropping and power cost keeps increasing ,in near future there might be some possibility where hardware cots is actually significantly lower than the power consumption cost to main the DCs.

Also more power consumption leads to more heat generation that leads to more power for cooling , which in turn increases the whole cots of DC operation.

**b. What is dynamic voltage frequency scaling (DVFS) technique?**

**Dynamic voltage** and **frequency scaling** (**DVFS**) is the adjustment of **power** and speed settings on a computing device's various processors, controller chips and peripheral devices to optimize resource allotment for tasks and maximize **power** saving when those resources are not needed.

**c. If you were to build a large $1B data center, which would require $50M/year in power costs to run the data center and $50M/year in power costs to cool the data center with traditional A/C and fans. Name 2 things that the data center designer could do to significantly reduce the cost of cooling the data center? Is there any way to reduce the cost of cooling to virtually $0? Explain why or why not?**

NO there is no way to reduce the cost to virtually zero as data centers are hosted inside buildings , there is always the requirement for keeping the airflow going inside buildings. For that running FANS are required which is going to consume electricity no matter what , so the cooling cost cannot be 0.

**8. Benchmarking**

**a. Throughput can be used to measure processors, memory, disk, and networks. What are the basic units of measurement for each of these?**

Processor : FLOPS (Floating point operation /second)

Memory : Transfer rate in GB/s

Disk: IOPS (Input /output /second)

Network: bits/second

10. Scalability

a. What does it mean for a system to be scalable? b. Describe the difference between strong scaling and weak scaling experiments.

**Scalability** is the capability of a system, network, or process to handle a growing amount of work, or its potential to be enlarged to accommodate that growth.[[1]](https://en.wikipedia.org/wiki/Scalability#cite_note-1) For example, a system is considered scalable if it is capable of increasing its total output under an increased load when resources (typically hardware) are added.

A system whose performance improves after adding hardware, proportionally to the capacity added, is said to be a ***scalable system***.

**b. Describe the difference between strong scaling and weak scaling experiments.**

Strong scaling : Data element size remains same on which processing has to happen as number of systems increases in numbers.

Weak scaling : Data element size increases on which processing has to happen as number of systems increases in numbers.

1. Processors

**a. Why did processors from the 1980s not need cache-coherent processors?**

Cache coherence problem only arrives in modern multiprocessor system where two or three levels of caches are present and same data is there in the cache of multiple processor on which some kind of conflicting (read/write) operation may happen which may destroy the consistency of the data, so they need to be protected .

in 1980 the systems were mostly single processor and was also having only one cache, so there was no problem of inconsistency.

**b. Assume that you have a x86 processor that has the following characteristics: 100 cores, 10 hardware threads per core, 1GHz speed for AVX640 instructions (vector of 640 bits) with 5 Fused Multiply-Add (FMA), 10GHz speed for all other instructions with 1 FMA and 2 instructions per cycle, 1MB L1 cache, 10MB L2 cache, and 100MB L3 cache, with a power envelope of 100 watts. What is the theoretical double precision (DP) TeraFlops per second?**

GFlops = (CPU speed in GHz) x (number of CPU cores) x (CPU instruction per cycle) x (number of CPUs per node).

5 FMA (Fused multiply-add) in AVX 640 .

multiply and add are 2 instruction in one cycle.

so 5 FMA = 5\*2 = 10 instruction.

DP arithmetic so, : AVX 640/64\*5\*2 = 10\*5\*2 = 100 DP FLOPS/cycle.

total cycle 1 GHz \* 100core \* 10HwThreads \* 100DP = 10000\*10 = 100000 GFLOPS.

another one

(1FMA+2 instruction) = (2 instruction + 2 instruction) = 4 instructions.

GFLOPs = 10GHZ \* 100core \* 10 HwThreads\* 4 instruction = 100 \* 100 \* 4 = 40000 GFLOps.

100000 + 40000 = 140000GFlops. / 1000 = 140 TFlops.

**9. Networking a. Name two network technologies you would use in building a large scale computing system? One network should be used to optimize cost, while the other should be used to optimize performance.**

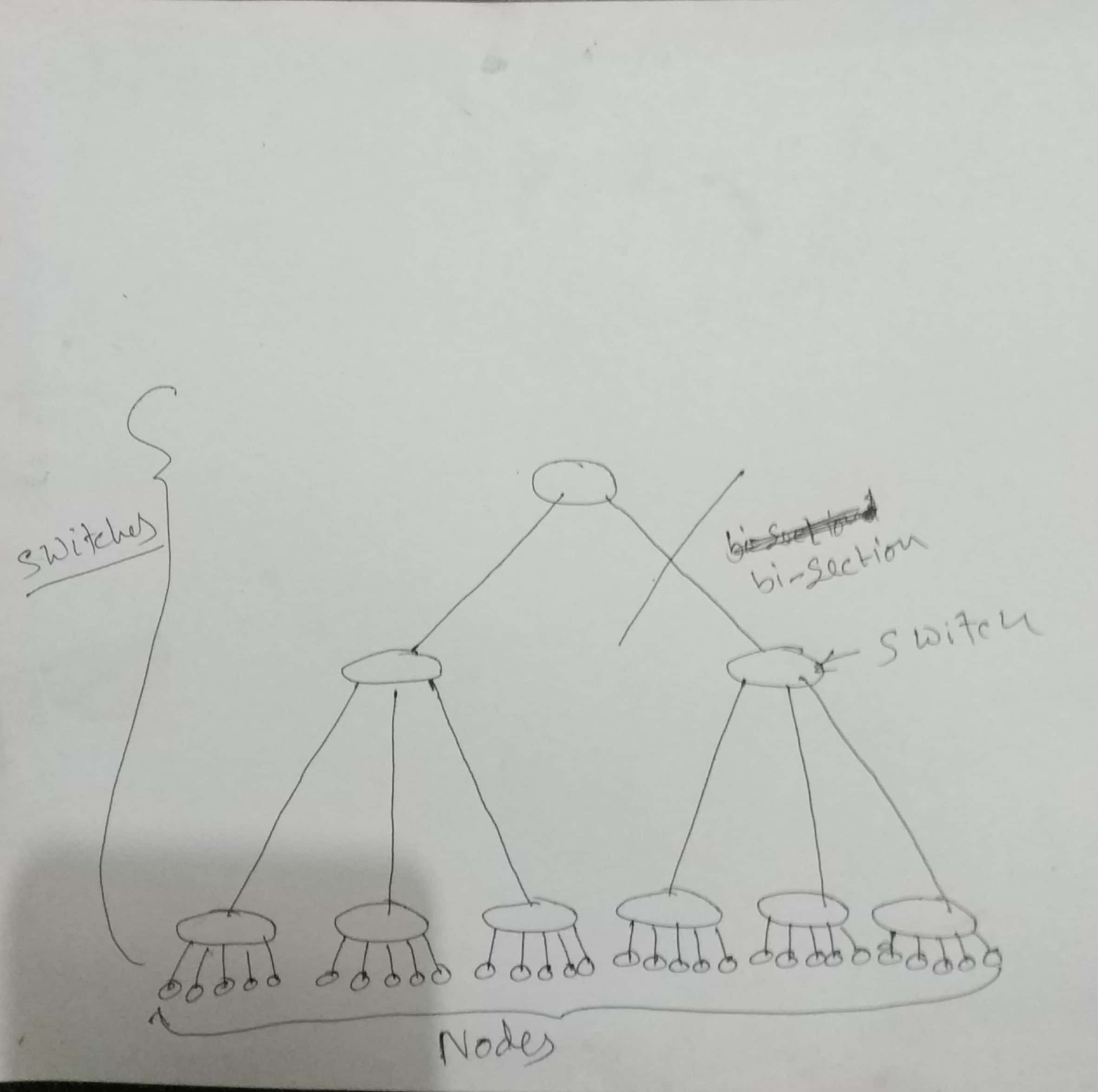
Optimizing performance , generally for supercomputer.

Dragonfly, Torus .

Optimizing Cost:

Fat -tree with higher port count switchs.

**b. Assume you have a cluster with 30 nodes. You have 1 network card per node with 1Gb/sec Ethernet Full Duplex, and have access to 6-port switches (also 1Gb/sec Ethernet Full Duplex) in order to build a Fat Tree network architecture. Draw a picture of the Fat Tree topology for your 30-node cluster (clearly show the switches, cables, and nodes).**



**(a) How many switches in total do you need?**

9

**(b) What is the bi-section bandwidth of your network?**

1GB/s

**(c) What is the bi-section bandwidth of your network in Gb/sec (round to nearest Gb/sec)?**

1GB/s

**(d) Assuming each switch incurs a 100-microsecond forwarding delay, and networking stack requires 40- microseconds to process network messages (e.g. TCP/IP) on each side (e.g. sender and receiver), what is the best-case and worst-case latency you can expect from this network topology?**

Base case : 40 + 100 + 40= 180 microsecond when both sender and receiver are connected to the same switch.

Worst case : 40 +100 + 100 + 100 + 100 + 100+ 40 = 580 microsecond. When sender and receiver are connected on two different bi-section.