# ASSIGNEMENT 1

## PAPER 1

## Solution 1) -

The paper takes on the tough challenge of making a scalable, reliable & efficient network infrastructure for handling distributed AI training at Meta. AI training jobs, like the ones for natural language processing, recommendation engines & also generative models, come with very heavy computation which also needs communication. These kinds of tasks usually need tight syncing of thousands of GPUs. This makes inter GPU communication a big bottleneck. For distributed training, we need high bandwidth and low-latency data transfers between GPUs. These leads to tough problems like traffic management, congestion control & making the system scalable. Existing ways to handle inter-node communication just do nnot cut it for these workloads. For instance, normal TCP/IP networks cause too much CPU overhead and bring in higher latency, while proprietary solutions like InfiniBand might work better performance-wise but are way too rigid and not fit for scaling to big deployments like what Meta has. On top of that, AI training often has traffic patterns that are bursty and unbalanced, which makes congestion worse and leads to inefficient performance. This problem gets worse as AI models keep getting bigger and need thousands of GPUs to work together in sync, which puts a lot of pressure on old networking methods. AI hardware, like GPUs with more memory and processing power (e.g., NVIDIA H100), adds more issues by needing higher bandwidth and newer designs. On top of this, congestion control methods like DCQCN (Data Center Quantized Congestion Notification) aren’t good enough for AI workloads that have complicated, hierarchical, and bursty patterns of communication. Because of this, the paper focuses on solving a big issue: how to make a high-performance network that can scale up and meet the unique needs of AI training. This includes making new designs for topology, routing, transport methods, and operational setups. At the end of the day, the main aim is to make a network that works reliably, predictably, and can handle scaling to tens of thousands of GPUs, letting AI models get trained more effectively.

## Solution 2) -

Yes, this is a really important problem because scalability and how efficient distributed AI training is have a direct impact on where modern AI tech goes next. The demand for AI models in industries is growing really fast. You see them being used for things like personalized recommendations, content analysis, generative AI, and massive language models (LLMs). These models need an enormous amount of computational power, which often gets spread out across thousands or even tens of thousands of GPUs. Training these big models quickly and efficiently isn't just nice to have—it's critical. It can lower costs and speed up how fast businesses, like Meta, can innovate and bring out new solutions. But the problem is, older networking methods like TCP/IP or even proprietary systems like InfiniBand have their limits. They're either too rigid, not flexible enough, or just can't scale the way AI needs them to. If these issues aren't fixed, training larger AI models would slow down big time because of congestion in the network, wasted bandwidth, and unpredictable delays. That means longer training times, higher costs, and wasted computational resources. And when you think about how AI apps are becoming part of everyday life, any delay in their development could ripple out and slow progress in big areas like healthcare, autonomous tech, and online communication. This problem matters even more for AI's future. As AI models keep growing in size and need more GPUs, older networking solutions won't be enough. Coming up with a more open and scalable setup not only helps Meta but also shows the rest of the tech world what's possible. By figuring out how to handle things like congestion, improve routing, and deal with bursty traffic in AI training, this paper lays down a path for a better and more efficient AI ecosystem. Fixing this makes large-scale training more practical, useful, and within reach for everyone working on AI.

## **Solution 3)-**

The paper solves the problem using RDMA over Converged Ethernet (RoCE). It is focusing on creating the specialized backend network for distributed AI training. This backend network is separate from the frontend, hence it can be optimized better. The network uses a Clos-based design which is scalable & provides low latency. It is supporting tens of thousands of GPUs, which makes it great for training in large scale. The paper explains how routing strategies were improved. It moved from basic ECMP to Enhanced ECMP (E-ECMP) & centralized traffic engineering (TE). These upgrades helped distribute traffic better, handle bursty patterns common in AI workloads, lower congestion. Another important change is how congestion is managed here. The paper moves away from DCQCN which was not effective and instead uses a receiver driven admission control system. This system works through the collective communication library  like NCCL.  Depending on network conditions  it adjusts traffic flow dynamically which boosts performance. The paper also talks about tuning the network and software together. For instance, Meta changed message sizes and buffer allocations to reduce latency and improve bandwidth use. Latency-sensitive operations like AllGather and ReduceScatter were improved to work well even in environments with higher RTTs.

Meta added observability tools too. These tools monitor network performance in real-time. They help find issues quickly by checking congestion metrics and RDMA hardware counters. This ensures the network runs smoothly even at scale. Overall, the paper contributes scalable RoCE networks that can handle thousands of GPUs. It offers better routing, improved congestion control, and operational insights. These solutions not only improve Meta’s infrastructure but also offer ideas for others in the industry.