Simulating the Future: The Role of PFC in AI Data Center Network Accuracy

As modern data centers evolve to support high-throughput, low-latency applications, particularly AI workloads, real-time analytics, and RDMA-based communication, the need for accurate modeling of congestion control mechanisms has continued to grow. Congestion control methods, such as Priority-based Flow Control (PFC), play a key role in enabling lossless Ethernet and ensuring efficient, predictable network performance.

Scala Computing includes full PFC support for network simulations run on their platform, providing a vital tool for data center network engineers and machine learning engineers who need realistic simulation of advanced networking features. This white paper underscores the need to account for PFC in simulations of modern data centers and how Scala's simulation platform delivers greater fidelity for congestion control modeling.

Priority-based Flow Control (PFC)

Priority-based Flow Control (PFC) is an enhancement to Ethernet flow control that allows individual traffic classes (priorities) to be paused independently. Originally introduced to address the absence of built-in congestion avoidance in the RoCE transport, PFC also helps prevent transport-layer timeouts caused by the loss of critical packets, such as the last packet or last ACK in a message. PFC is defined by IEEE as a layer 2 link level function and is a foundational technology for enabling **lossless Ethernet**.

Unlike traditional Ethernet or TCP/IP environments that rely on packet loss and retransmissions to signal congestion, PFC enables **selective pausing of traffic** to prevent buffer overflows while allowing higher-priority traffic to continue. This fine-grained control is crucial in scenarios where loss of even a single packet can degrade performance or disrupt sensitive transactions.

Without PFC Support:

- Packet drops at the Ethernet layer trigger retransmissions at higher layers, such as TCP, wasting **bandwidth and increasing latency**.
- Applications like RDMA fail to achieve their designed efficiency.
- AI/ML clusters can experience packet drops, lowering overall utilization and raising costs.
  Note: *Introducing PFC can result in Head-of-Line blocking. Simulation can be used to compare methods and understand tradeoffs.*



Importance of Accurate Congestion Modeling

The rise of AI-centric and data-intensive applications has fundamentally reshaped the demands placed on network infrastructure. Today's large-scale data centers must accommodate a new class of traffic patterns characterized by **high throughput, low latency, and extreme sensitivity to congestion**. PFC is essential for high-performance workloads such as RDMA (Remote Direct Memory Access), AI model training, and storage protocols. These evolving workloads make **Priority-based Flow Control (PFC)** essential to ensure reliable, efficient, and lossless data transmission across the network.

Accurate simulati**on of congestion control (PFCs) cannot be left as an afterthought—it is a first-order design constraint.** Mischaracterizing the network behavior can lead to inaccurate performance predictions, flawed capacity planning, and inefficient protocol tuning.

Here are some key features of modern data centers and how simulating PFCs is relevant for deep understanding:

- **Advanced Transports:** Many of these, such as **RDMA over Converged Ethernet (RoCE)** are highly sensitive to packet loss. Simulating with PFC enables lossless ethernet for specific traffic classes so engineers can validate protocol behaviors under load and ensure that their transports will perform efficiently at scale to meet network latency and throughput targets.
- **AI/LLM Workloads**: Both LLM training and inference often involve traffic patterns that can create congestion hotspots in the network. These communication bottlenecks can significantly affect performance and convergence times. PFC helps ensure consistent performance by preventing head-of-line blocking and packet drops in congested environments, at each level and for separate traffic classes, instead of globally pausing all traffic. It is critical to include flow control in the simulation in order to represent the real-world behavior.
- **Multi-Tenant Isolation and QoS:** Data centers run multiple tenants and applications on the same network and simulating with PFCs, with the correct traffic prioritization, per class, helps engineers ensure that critical traffic (such as storage or real-time inference) will be protected.
- **Disaggregated infrastructure:** With disaggregated **storage, compute and accelerators**, traffic between components becomes more performance-critical and PFC is needed to avoid packet lost at key congestion points. When designing resilient, **scala**ble network topologies, accurately simulating with PFCs is critical for accurately capturing the network behavior.

Use cases

Scala models differences in ingress admission behavior in switches, as well as how output queue congestion propagates into input-side accounting and PFC triggers. Scala users leverage this modeling precision to tackle the complex challenges of configuring PFC effectively in live environments.

In real-world deployments, Priority Flow Control (PFC) can introduce challenges such as head-of-line (HOL) blocking and the risk of congestive collapse due to backpressure in multistage network topologies. Simulation plays a critical role in evaluating whether these risks are real for a given network and workload combination. It also enables comparison with the drawbacks of operating in lossy network environments. In some cases, simulation supports hybrid approaches, such as selectively enabling PFC only between the NIC and the leaf switch, an increasingly common practice among operators.

Rather than aiming for a fully lossless network, many view PFC as a tool to reduce packet loss within acceptable margins. Scala's simulations help operators explore this tradeoff, offering a probabilistic view of thresholds and buffer headroom that supports detailed sensitivity analysis.

Some treat PFC as a backup mechanism that should only activate when the congestion control features included with their transport protocols, such as RoCE or UET, are not sufficient. Scala simulations are frequently used to assess the effectiveness of this layered approach to congestion control.

Coupling effects and "noisy neighbor" issues are of significant interest when combining PFC based jobs in the same network or superimposing lossless and lossy traffic in the same network. Scala's simulation platform offers detailed logs and analytics to determine whether such interactions occur and how they impact system performance.

Scala Computing provides Full PFC Support

To address this need, Scala Computing's ns-3 model interfaces provide native support for Priority-based Flow Control (PFC). PFC endpoint and switch behaviors vary between vendors and have a major impact on overall distributed system performance. Scala Computing captures these nuances through vendor-specific ns-3 models that closely replicate real-world hardware behavior, and through Scala-generic models that embody practical, optimized implementations.

The PFC enhancements include:

- **Interaction with higher-layer protocols such as RDMA and AI traffic flows**
- **Class-based traffic differentiation**:
    - Enable PFCs for each of the 8 individual traffic classes
- **Per-priority pause frame generation and handling**
    - Per traffic class settings such as PFC thresholds to trigger the generation of pause/resume frames, pool consumption vs headroom, and setting up a pause quanta value.
- **Realistic modeling of ingress/egress buffer behavior**

    - For the Scala Generic switch, Auto PFC Configuration Mode automatically calculates the Headroom Pool allocation and limits per port based on network link speed and cable length.
- **Set up and Monitor PFCs at every layer**
    - Charts show time series PFC propagation in the network.
    - Tables provide counts of how many PFCs were generated by each tier of the network
    - Some switches include charts to monitor Head-of-line blocking at each tier, and at the sending endpoint.

PFC Statistics

Tier	XOff RX	XOff TX	XOn RX	XOn TX	Total Time Paused (usec)	Direction (UL/DL)
Global	422040	422040	286455	286455	4920784.8	
Spine Switch	0	0	0	0	0	ul
Spine Switch	0	0	0	0	0	dl
Fabric Switch	0	0	0	0	0	ul
Fabric Switch	76013	104745	58862	65491	667751	dl
Rack Switch	104745	76013	65491	58862	808628	ul
Rack Switch	0	241282	0	162102	0	dl
NIC	241282	0	162102	0	3444404.7	ul
NIC	0	0	0	0	0	dl

This ability to simulate PFCs allows for the precise reproduction of lossless Ethernet behavior that occurs in real world PFC enabled networks, making it possible to:

- Evaluate protocol and hardware designs under realistic load
- Tune flow control thresholds and buffer sizes
- Predict system behavior under varying congestion scenarios

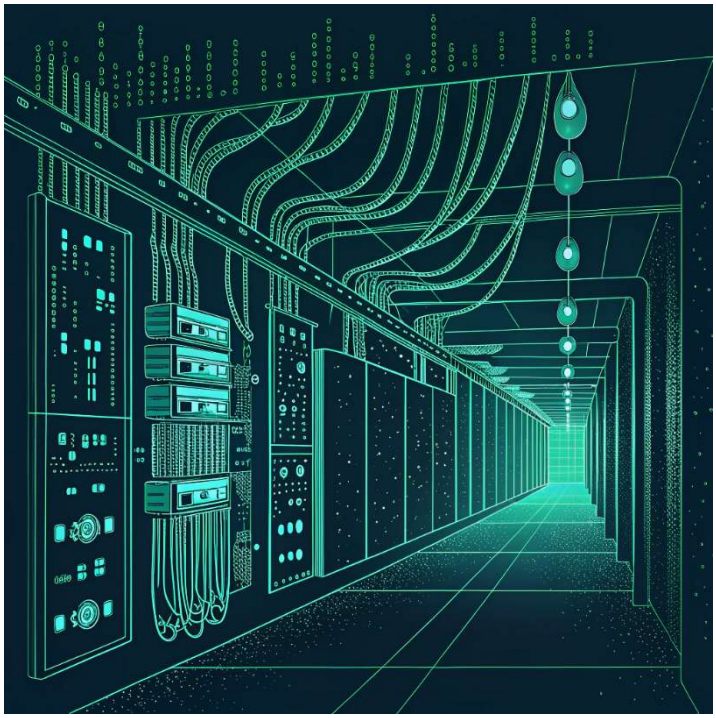
These capabilities are vital for companies designing next-gen AI data center infrastructure.

Conclusion

As data centers and AI/LLM workloads grow more complex, simulation tools must evolve to reflect the true nature of modern congestion control in order to truly optimize the network utilization. Priority-based Flow Control (PFC) is a foundational technology in lossless Ethernet and must be accounted for at every layer when simulating data center networks.

In addition to scalability and many other key capabilities, Scala Computing offers full PFC support, empowering engineers to make informed decisions based on accurate, high-fidelity, large-scale simulations. For anyone building or optimizing modern data center fabrics, simulating with all the correct flow features, including PFC, is not merely helpful, **it is essential!**

Network Simulation must support Priority-based Flow Control (PFC) to meet the demands of High-performance transports such as RoCE, AI/LLM training and inference, Disaggregated and multi-tenant architectures and Traffic-class differentiation for real-time vs. best-effort flows.



For more information about Scala's network simulation platform (with advanced PFC support) and how it can support your AI Network Simulation needs, please contact info@scalacomputing.com or visit www.scalacomputing.com.