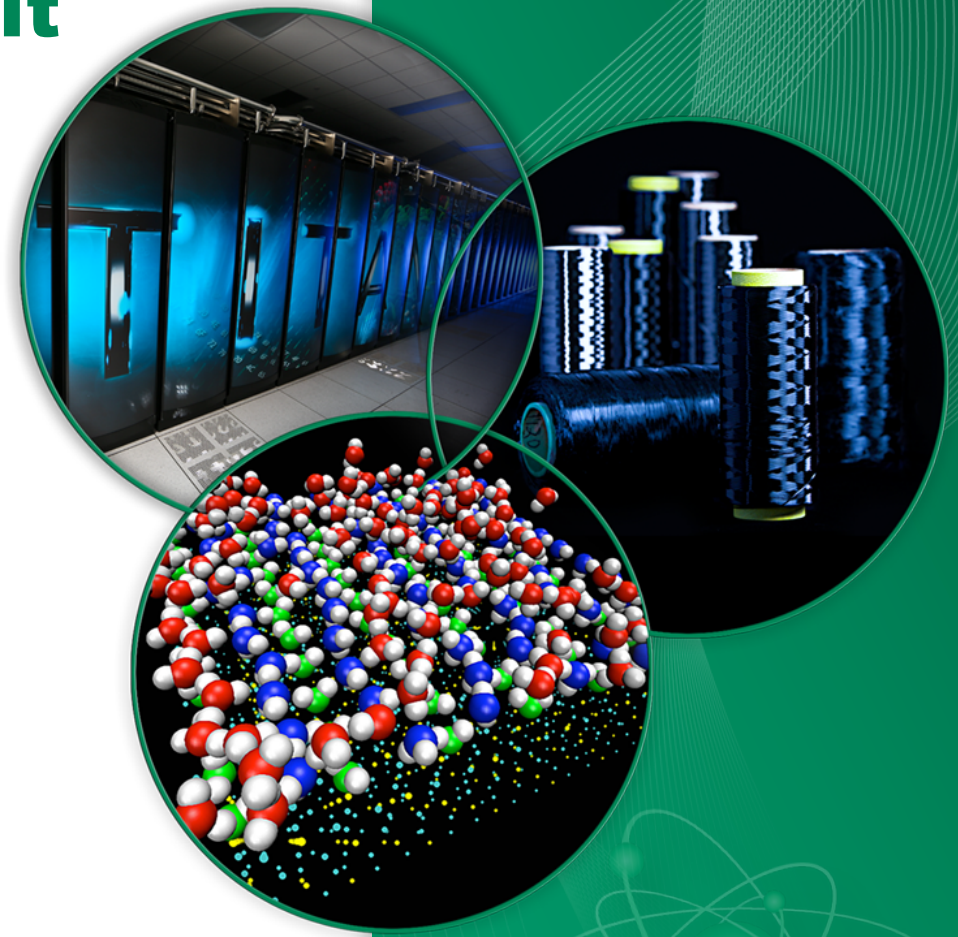
# An Adaptive End-to-End Approach for Terabit Data Movement Optimization



ORNL: Galen Shipman, Scott Atchley, Youngjae Kim, Geoffroy Vallee

UTK: George Bosilca

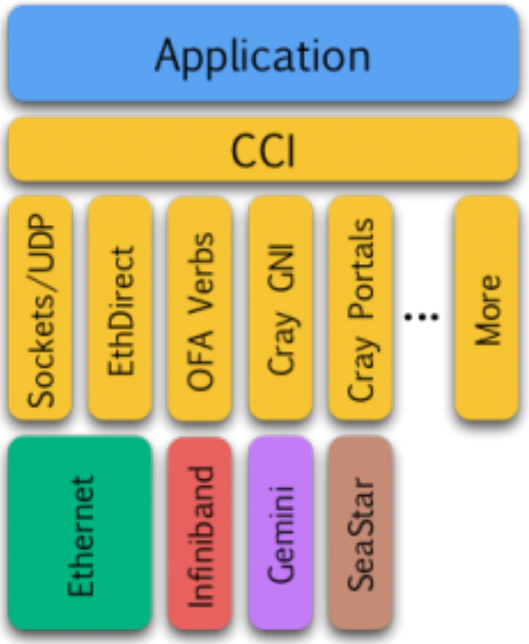
# Motivation

- Scientific discovery requires large collaborations
  - More complex workflows
  - Geographically distributed
- Data explosion
  - DOE laboratories expect an exponential increase of scientific data
  - Require enhanced systems and infrastructure for end-to-end high-speed data transfer between user facilities
- Challenges: How can we have efficient end-to-end data transfer?
  - Parallel file system access
  - Terabit networks

We propose to provide an adaptive end-to-end solution for terabit data movement between DOE facilities

# High Performance Communication Substrate

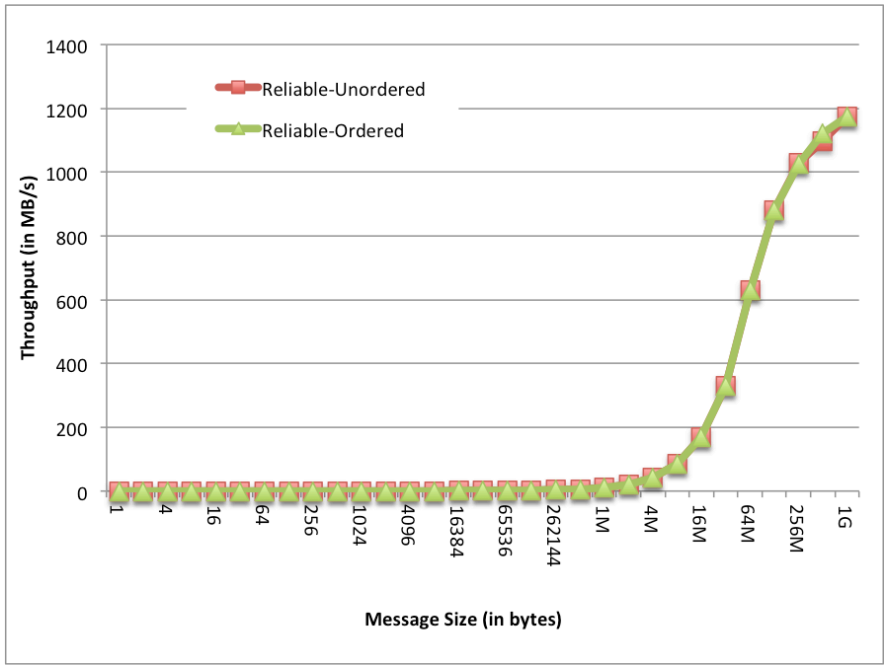
- Common Communication Interface (CCI)
  - Simple, portable and scalable network abstraction layer
  - Support all major networking technologies (Infiniband, native Ethernet, TCP, UDP, Cray Gemini)
  - Support both one sided and send/receive operations
  - Event based model
  - Target near-line rate performance
- Contributions
  - Extensions and tuning for the WAN
  - Routed communications
  - Integration with Open-MPI for MPI support (led by UTK)



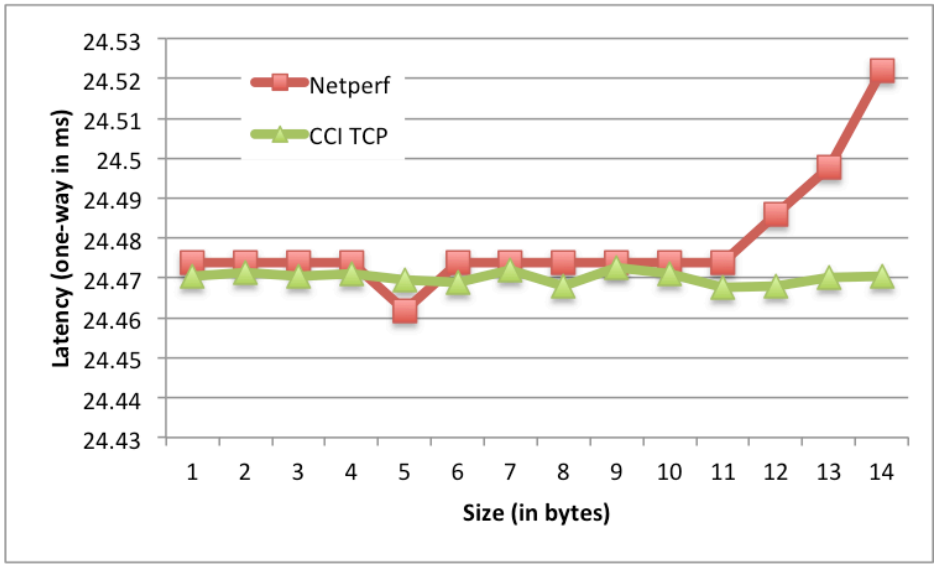
# CCI – WAN Performance Overview

- Testbed: ESnet ANI 100Gb network
- Results over 10Gb network between ANL and NERSC

*RMA TCP Bandwidth*

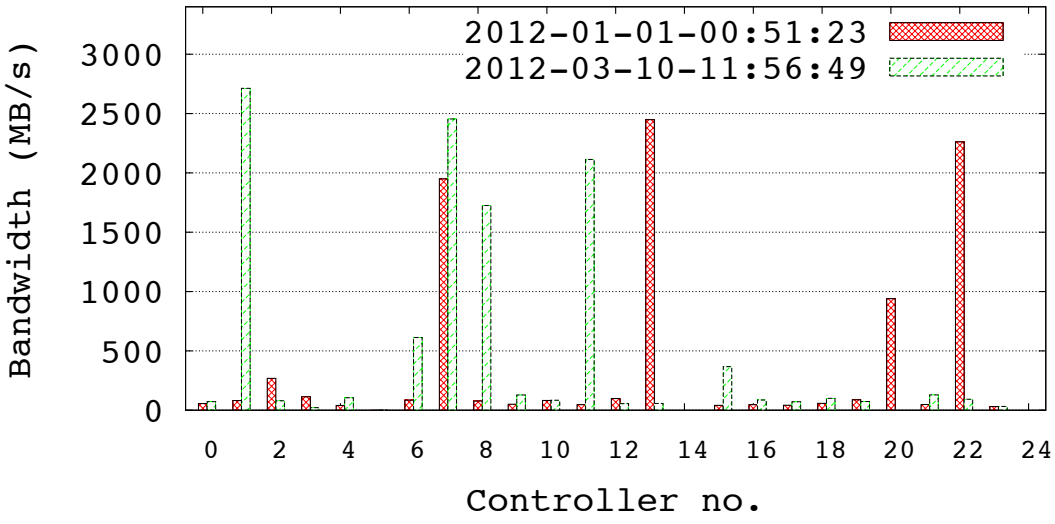



*TCP Latency*



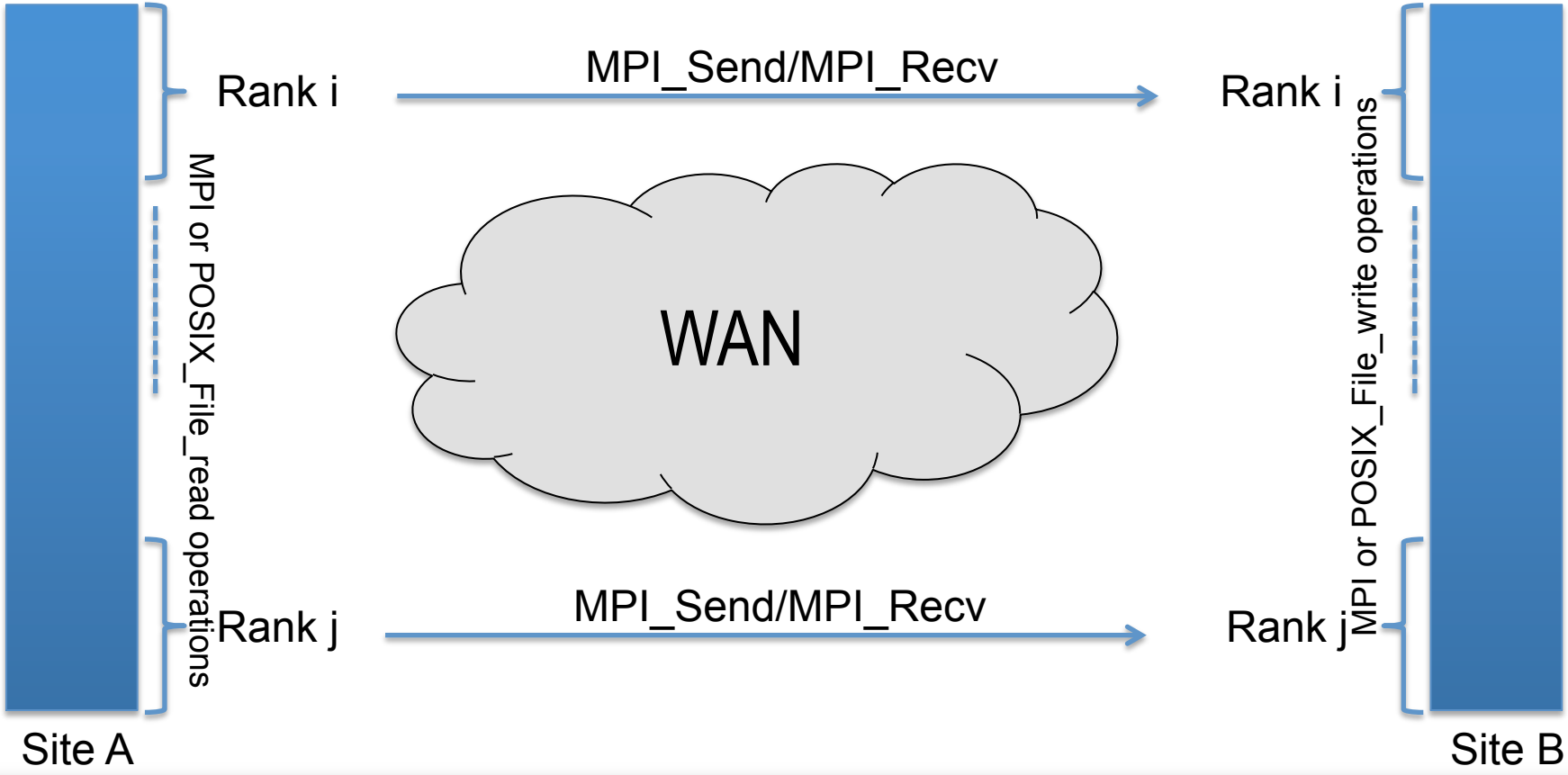
# End-to-End Data Movement

- Challenges
  - High performance end-to-end data movement requires
    - Efficient access to the parallel file system (both on the source and sink)
    - High performance network communications
    - Coordination between file system access and network to avoid bottlenecks
  - Simple tools and methods for users
- Contributions
  - Basic coupling of MPI codes
  - Z-scheduler
  - ZS scheduler



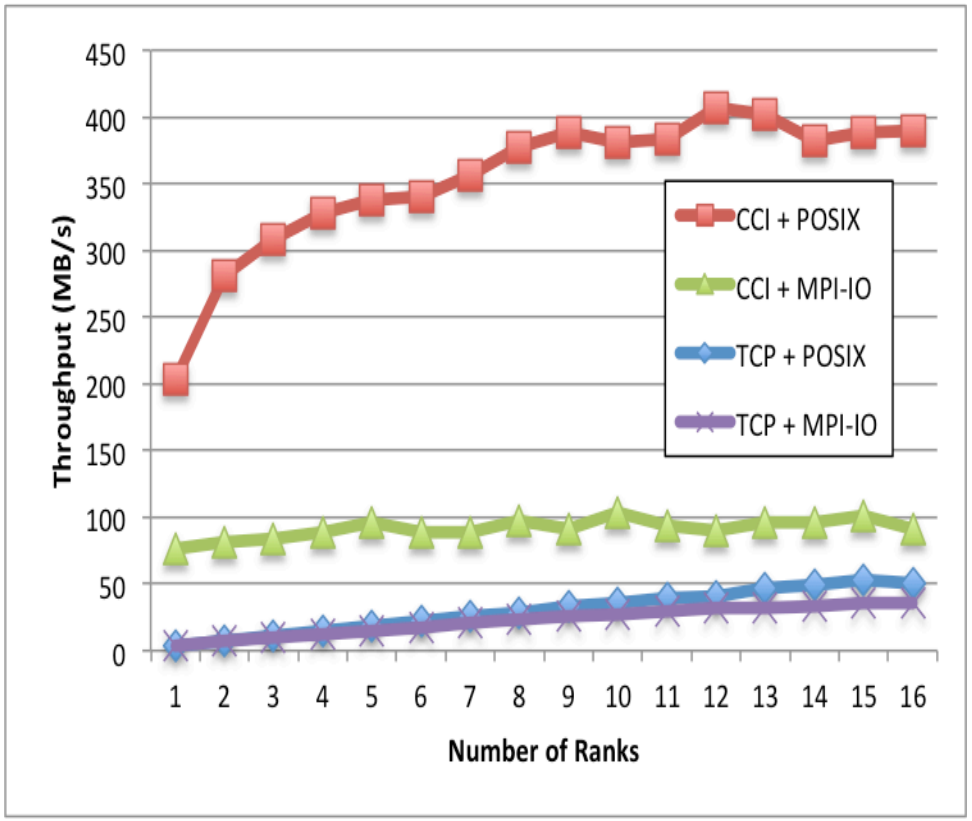
# Use Case – Code Coupling Over WAN

- Code coupling use case (MPI connect/accept)
- Familiar model for users
- No access coordination to PFS and network resources



# Use Case – Performance

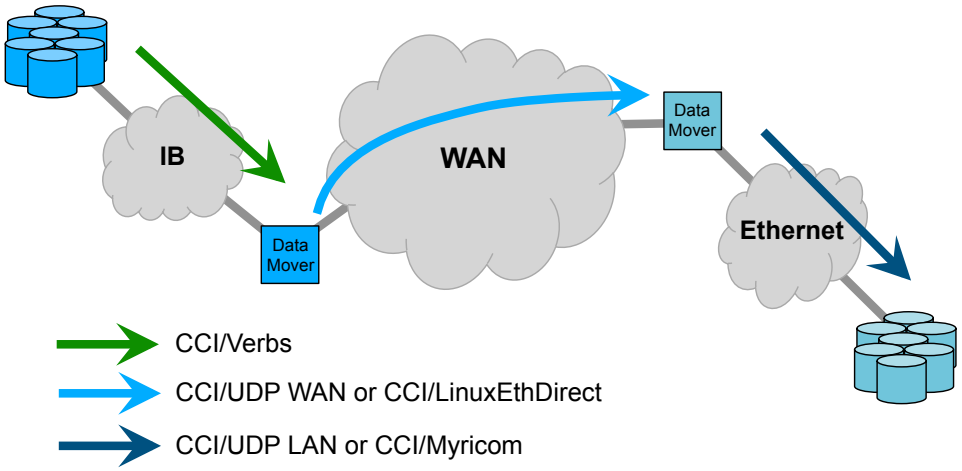
- Transfer of a 1GB file over ESnet ANI between ANL and NERSC (10Gb ethernet)
  - Support both POSIX and MPI-I/O for file access
  - Parallel file access with parallel connections

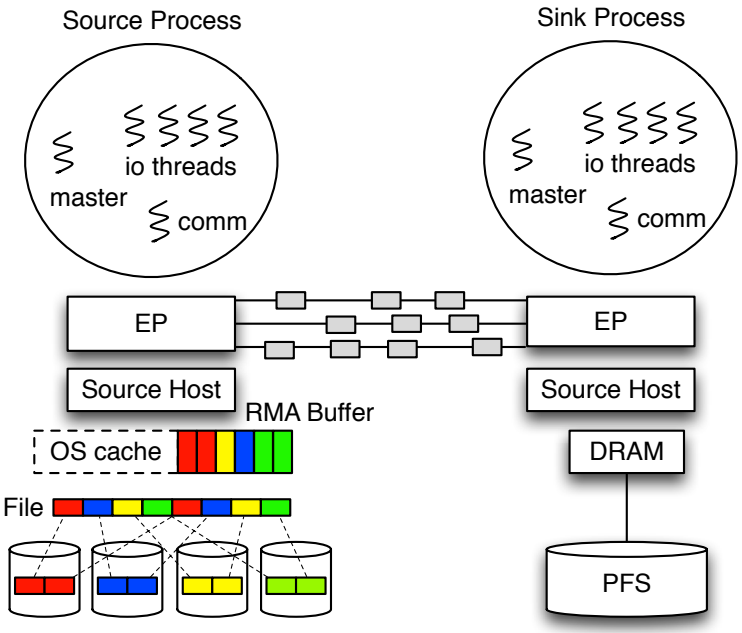


# ZS Scheduler

## Main characteristics

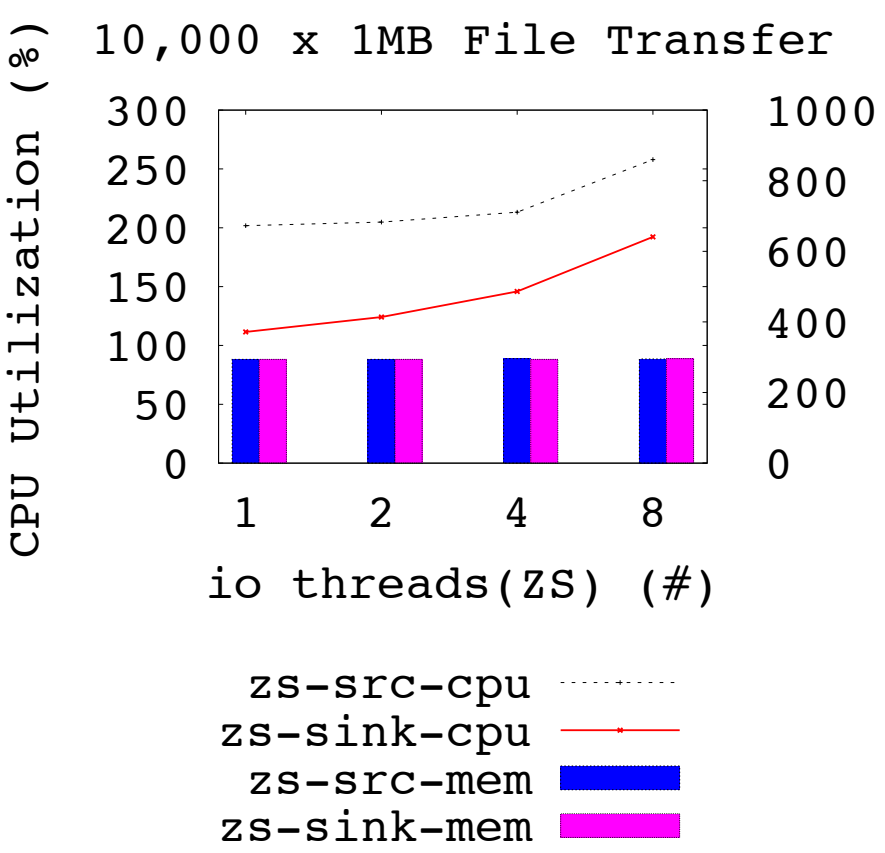
- Congestion aware I/O optimization techniques
- File layout-awareness for data access
- Solve I/O impedance between storage and network using CCI
- Enable source-based buffering technique using flash storage

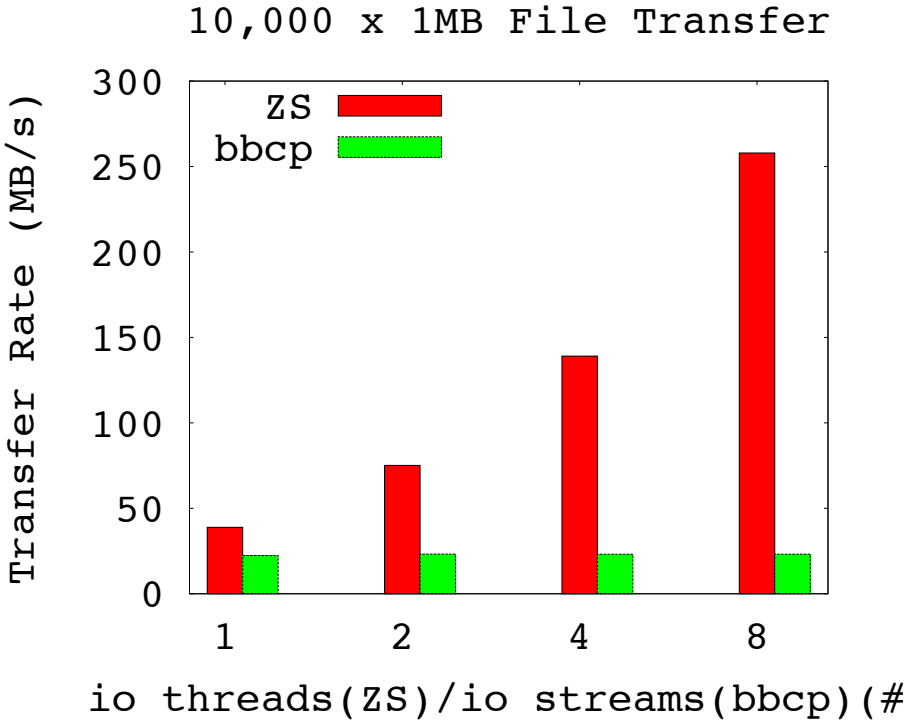


# ZS Scheduler – Performance (2)

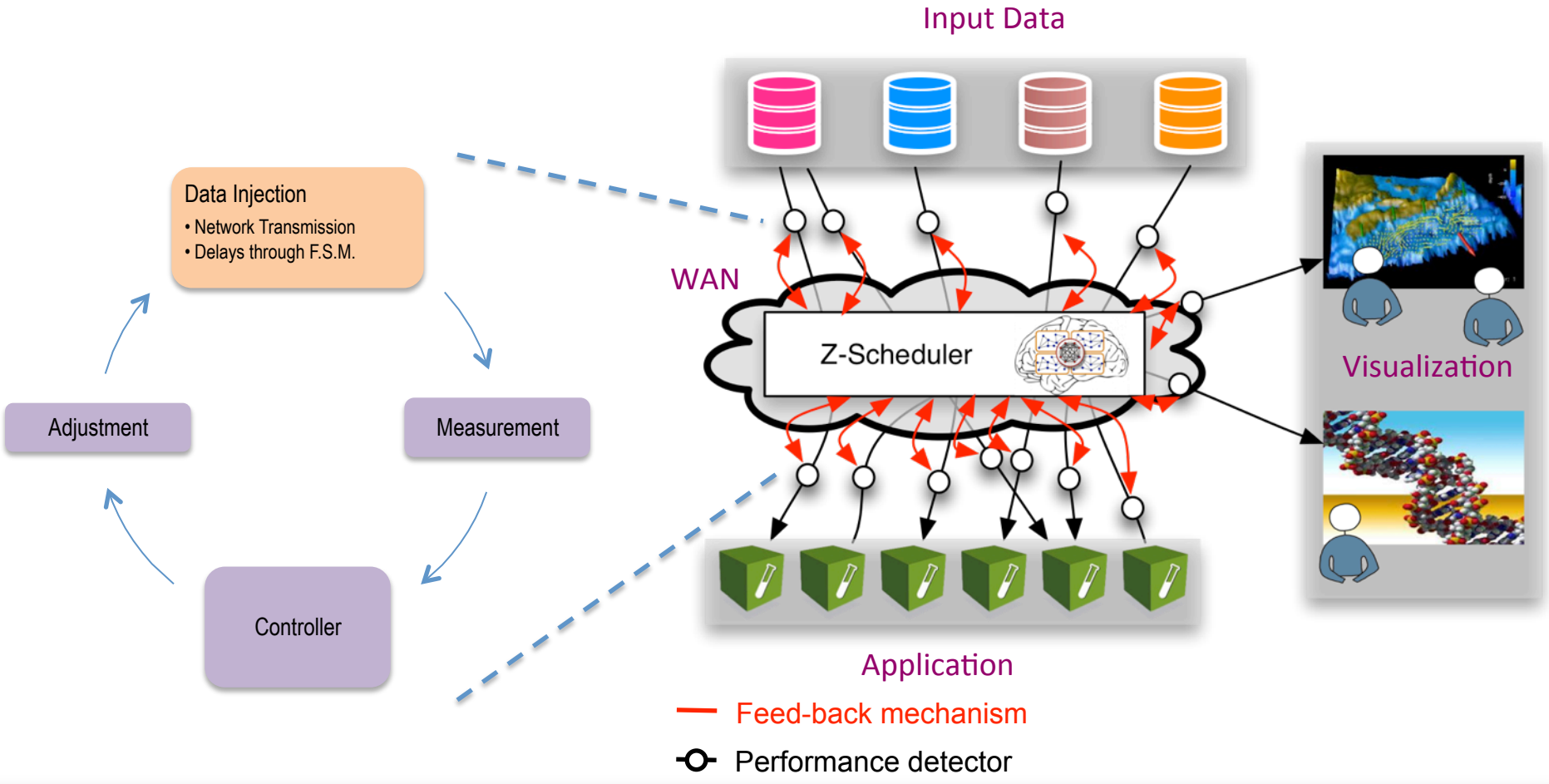
- Comparison of ZS and BBCP for end-to-end data transfer



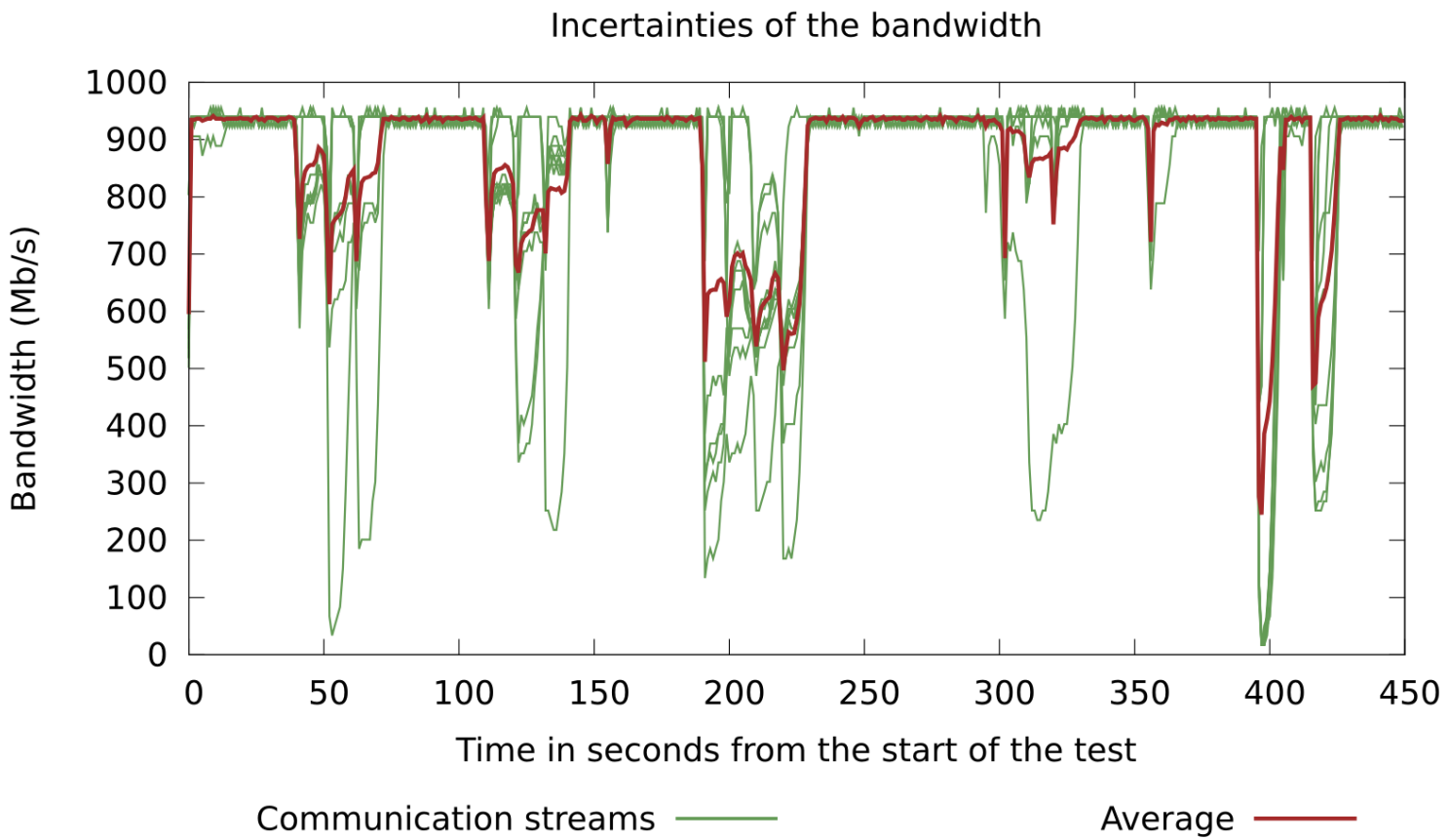



# Z-Scheduler – Overview

- Online prediction of the network capacity and dynamically adapt the each peer bandwidth
- Orchestrate the data transfer to match the access pattern of the application



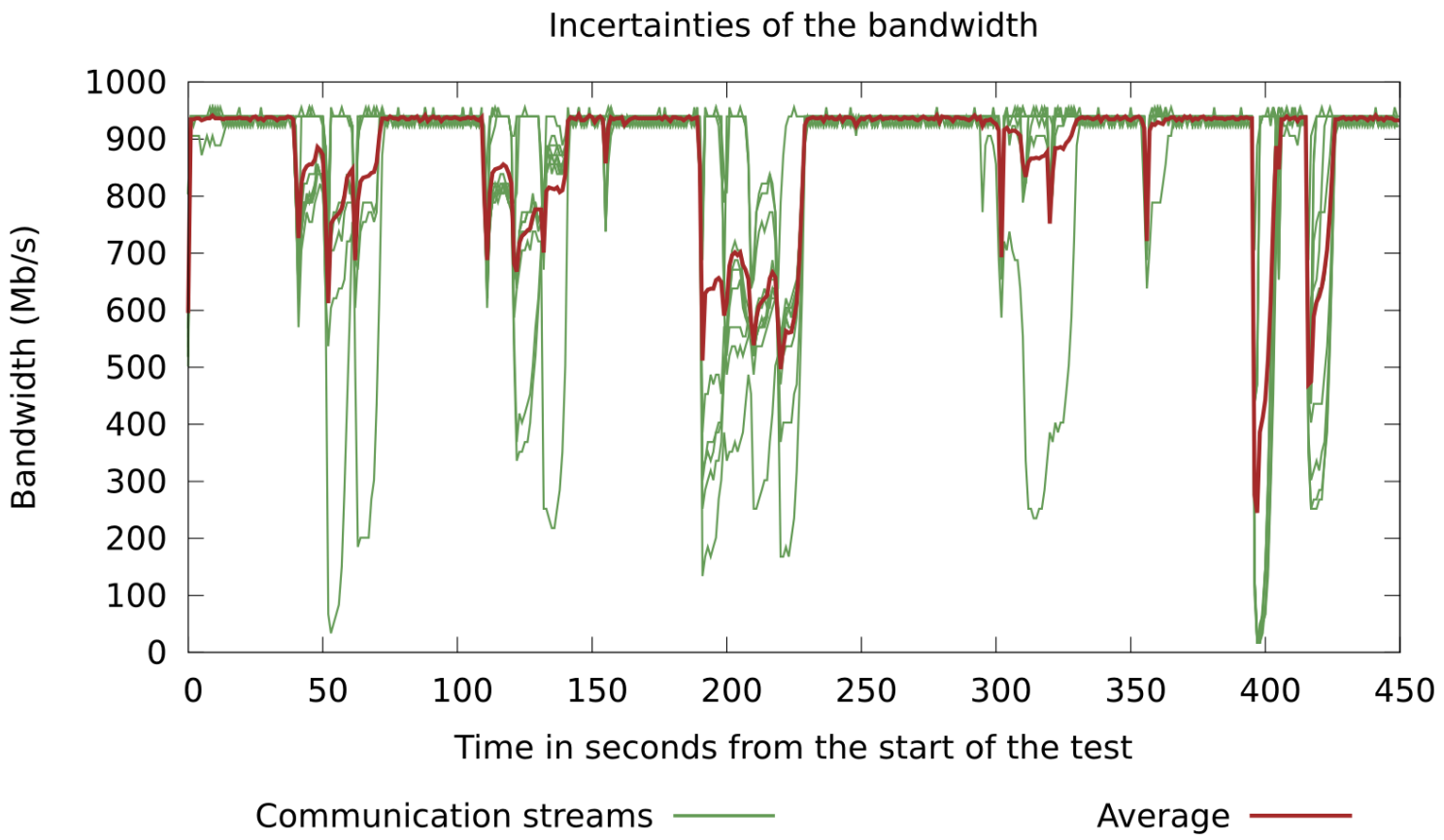
# WAN: TCP bandwidth under stress




Intense bandwidth:

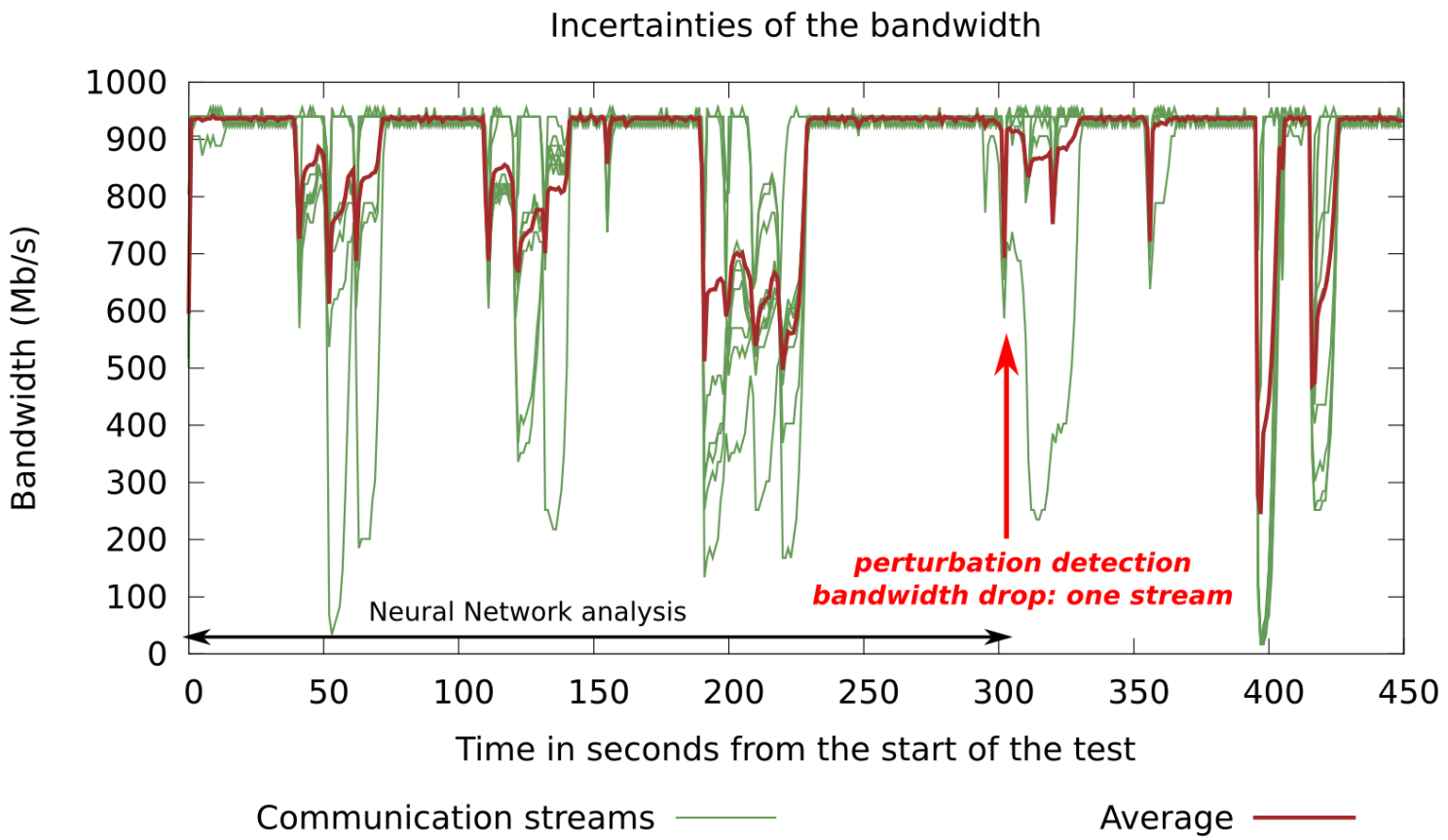
- data stream bandwidth can be largely by stress
- no fairness guaranteed

# WAN: TCP bandwidth under stress



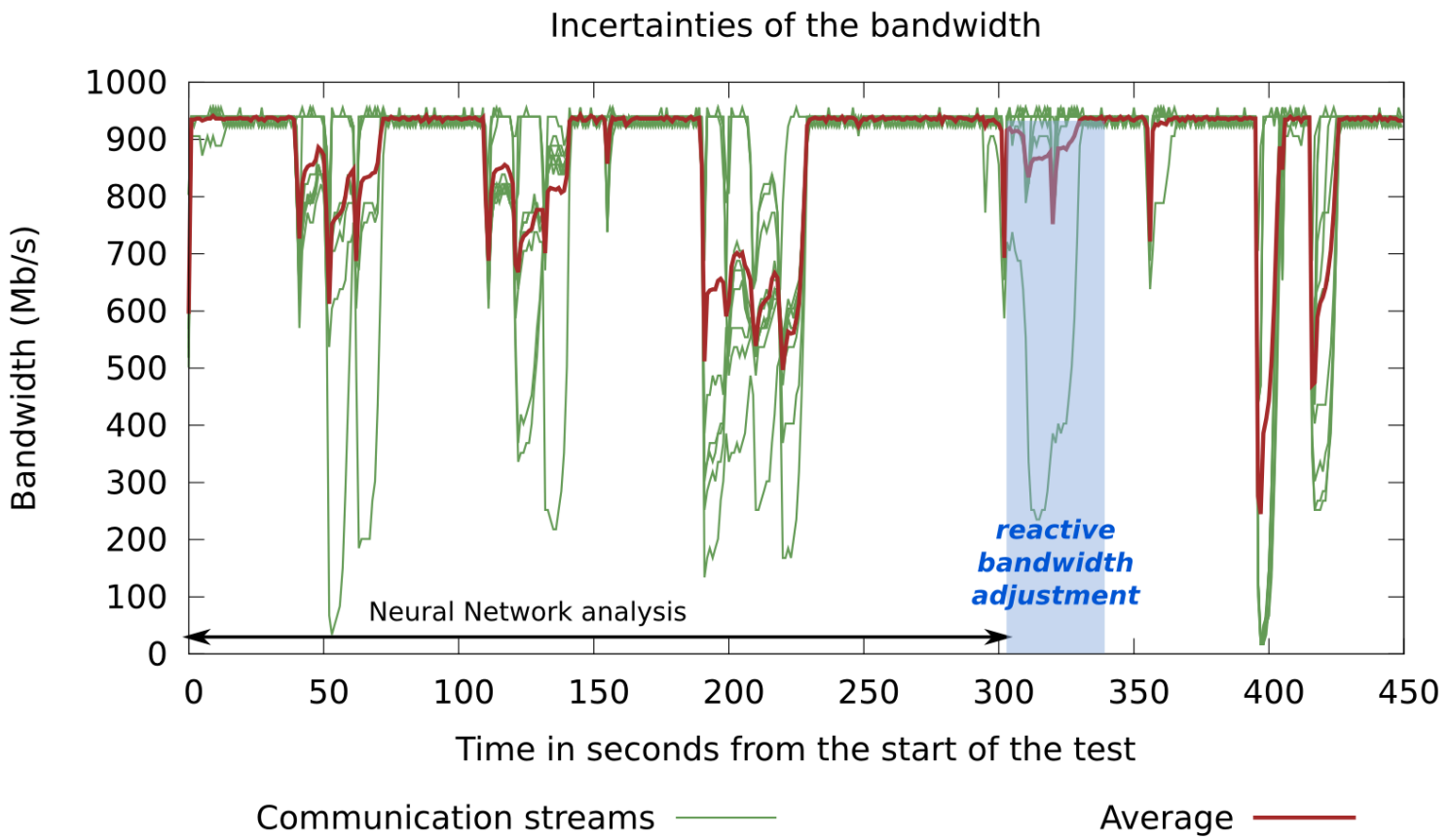

Performance feed-back mechanism to learn about specific classes of perturbations and their impact

# WAN: TCP bandwidth under stress




New perturbations predicted and classified
Training continue during the entire transfer
Traces used to predict characteristics of the traffic based on temporal properties.

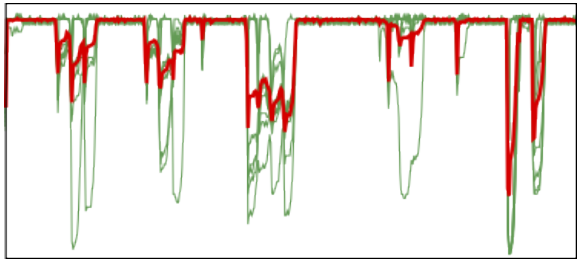
# WAN: TCP bandwidth under stress




Dynamically adjust the bandwidth to satisfy the requirement determined by the Z-scheduler.

# Z-Scheduler – Neural Network

- The quality of the bandwidth adjustments is directly linked to the performance of the underlying construction and update of the neural networks
- Systems of interconnected “neurons”
  - Can compute values from inputs by feeding information through the network
    - Machine learning
    - Pattern recognition



→ Model difficult nonlinear systems

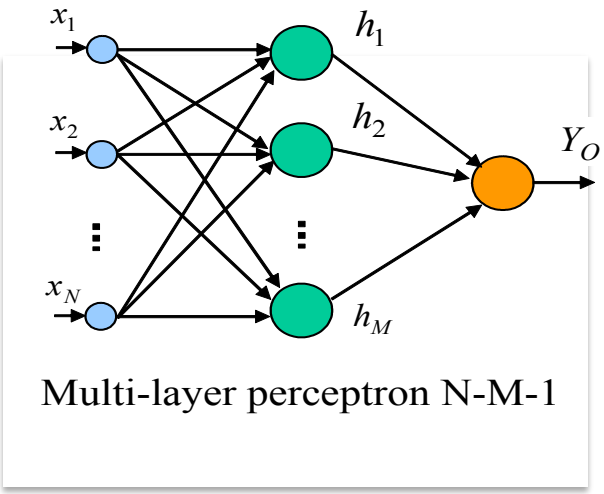
→ Alternative to solve complex problems

- Image processing, pattern recognition, robotics, etc

→ Solvers of intelligent tasks

- Prediction, classification, recognition, optimization, etc

→ Good generalized properties, self-adaptation



Multi-layer perceptron N-M-1

# What question does your research motivate you to now ask?

- What about the exascale and interactions with instruments?
  - How can we handle streams from different sources without interfering computation on exascale systems?
  - How can we handle short-lived data streams?
  - How can we co-schedule resources across different administration domains and ensure end-to-end performance?
- Investigation opportunities
  - Extend the concept of unique namespace to ease co-scheduling across multiple administration domains
  - Investigate scheduling algorithms to guarantee end-to-end performance and data integrity
  - Investigate multi-user co-scheduling to avoid bottleneck pushing/ pulling data to/from exascale storage