Large-scale experiments on a cluster

Liang Wang
Supervisor: Prof. Jussi Kangasharju
Dept. of Computer Science
University of Helsinki, Finland

Large-scale experiments

- Motivation
  - Modern systems are large and distributed.
  - Need to evaluate robustness, adaptability and performance.
- Three (four) options
  - Simulator
  - Internet
  - Cluster
  - (Analytical)

Why on the cluster

- With cluster, we can
  - easily control all the participants and access all the data;
  - make large-scale experiments reproducible;
  - simulate different real-life scenarios by using different parameters;
- It looks beautiful, however,
  - cluster is always “smaller” than the experiment scale we want.
  - design and deploy experiment is non-trivial.

Ukko cluster

- Introduction
  - computing infrastructure for the research and education purpose in the Dept. of Computer Science, Univ. of Helsinki.
  - everyone in the department can access it.
- Specification
  - 240 Dell PoweEdge M610 nodes, connected with 10-Gb link;
  - Each node has 32GB of RAM and 2 Intel Xeon E5540 2.53GHz CPUs
  - Each CPU has 4 cores, there can be 16 concurrent threads due to hyper-threading.
  - (Part of our work was done on HIIT cluster)

Our work & aims

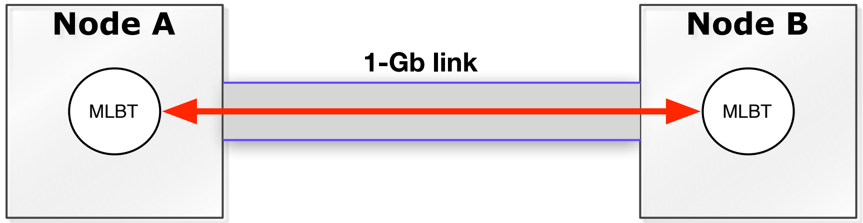
- Aims in the long-run
  - In a nutshell, measure & evaluate large-scale distributed systems in a systematic and consistent manner.
- Currently, we ...
  - focus on P2P system (BitTorrent) evaluation in cluster environment.
  - develop simple but flexible tools to deploy the experiments and automate the whole process(deploying, collecting data, simple analyzing).
  - figure out various restrictions on the large-scale experiments on Ukko cluster
  - study how to design reasonable experiments.
  - try to gain experience for future evaluation for other systems.

BitTorrent experiment

- Why it is worth study
  - The dominant file-sharing protocol in the world - real-world data can be used to validate the results from the cluster experiments.
  - A good starting-point - there is abundant literature can be referred to.
  - A typical complex system - peer-level behaviors are simple and easy to understand, the system's overall behaviors are complicated.
- Experiment target
  - Instrumented clients are widely-used in research area. There are several ready-made ones, but not full-fledged. We use our own BitTorrent client, based on official version.
  - Evaluate different implementations, mainly focus on Mainline Ver4.

Some practical issues

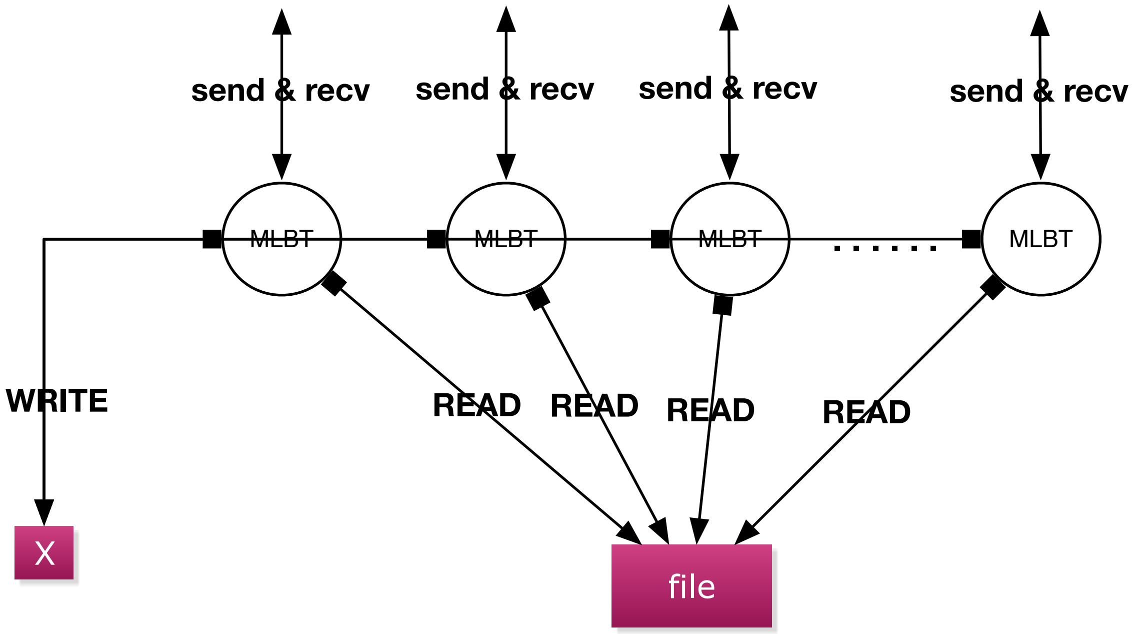
- Bypass I/O
  - I/O operations to the hard disk are bypassed. Not only because of the limited storage capacity, it is the first bottleneck of the performance.
  - With the simplest experiment setting, one seeder, one leecher, and no limits on the transmission rate,



I/O bypassed?	stable transmission rate	CPU resources on I/O wait
No	70MB/s	over 85%
Yes	115MB/s	almost 0%

Some practical issues (contd.)

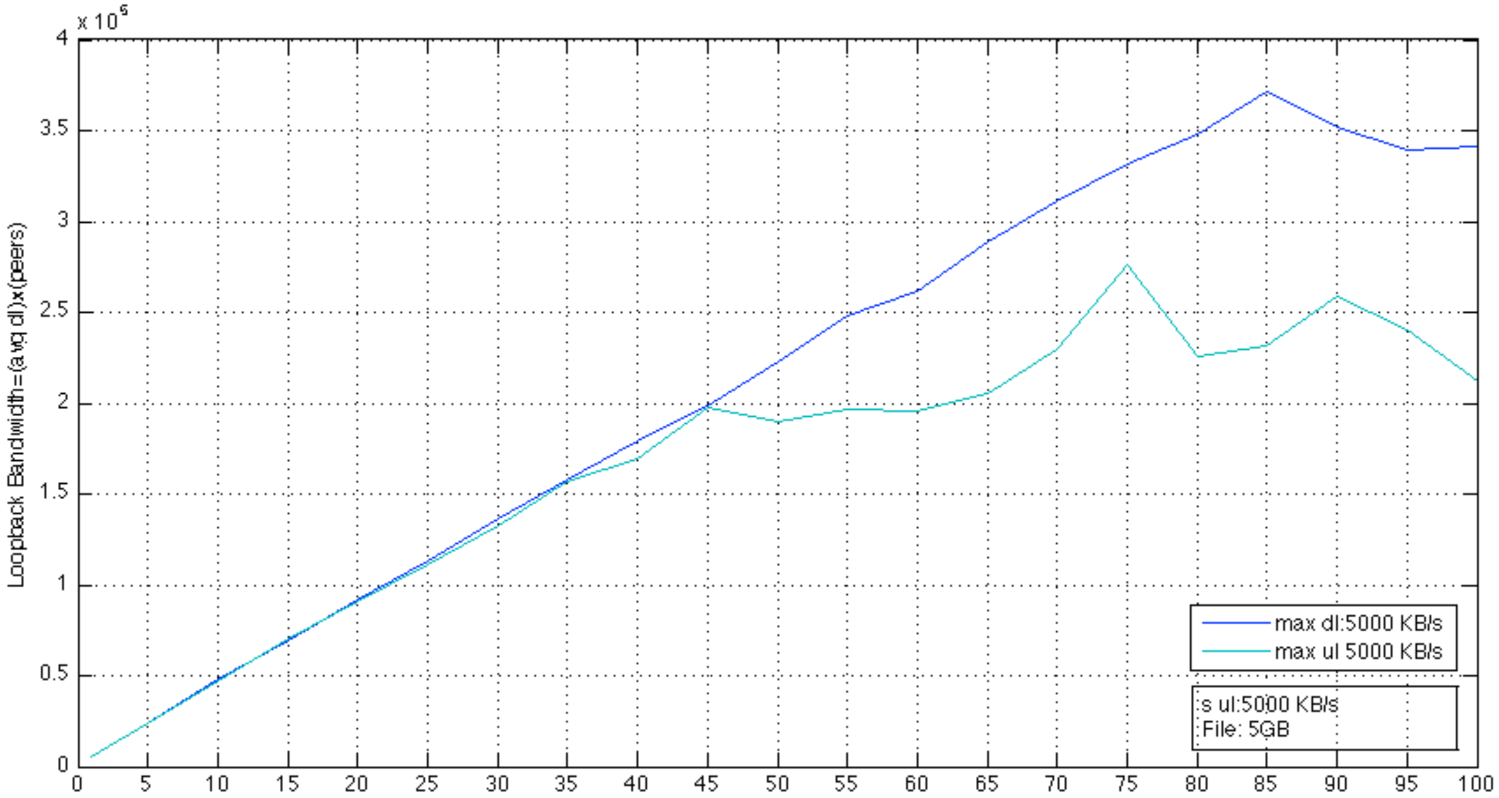
- Running multiple instances on one node
  - Reason: maximize the utilization; enlarge the experiment scale with limited resources.
  - Method: application-layer isolation, no hypervisor is used. Pros & Cons?
  - Lots of nasty issues needs to take care -- e.g. I/O overheads, storage issue, system parameters.
  - Bypass the write operations, redirect the read operations.

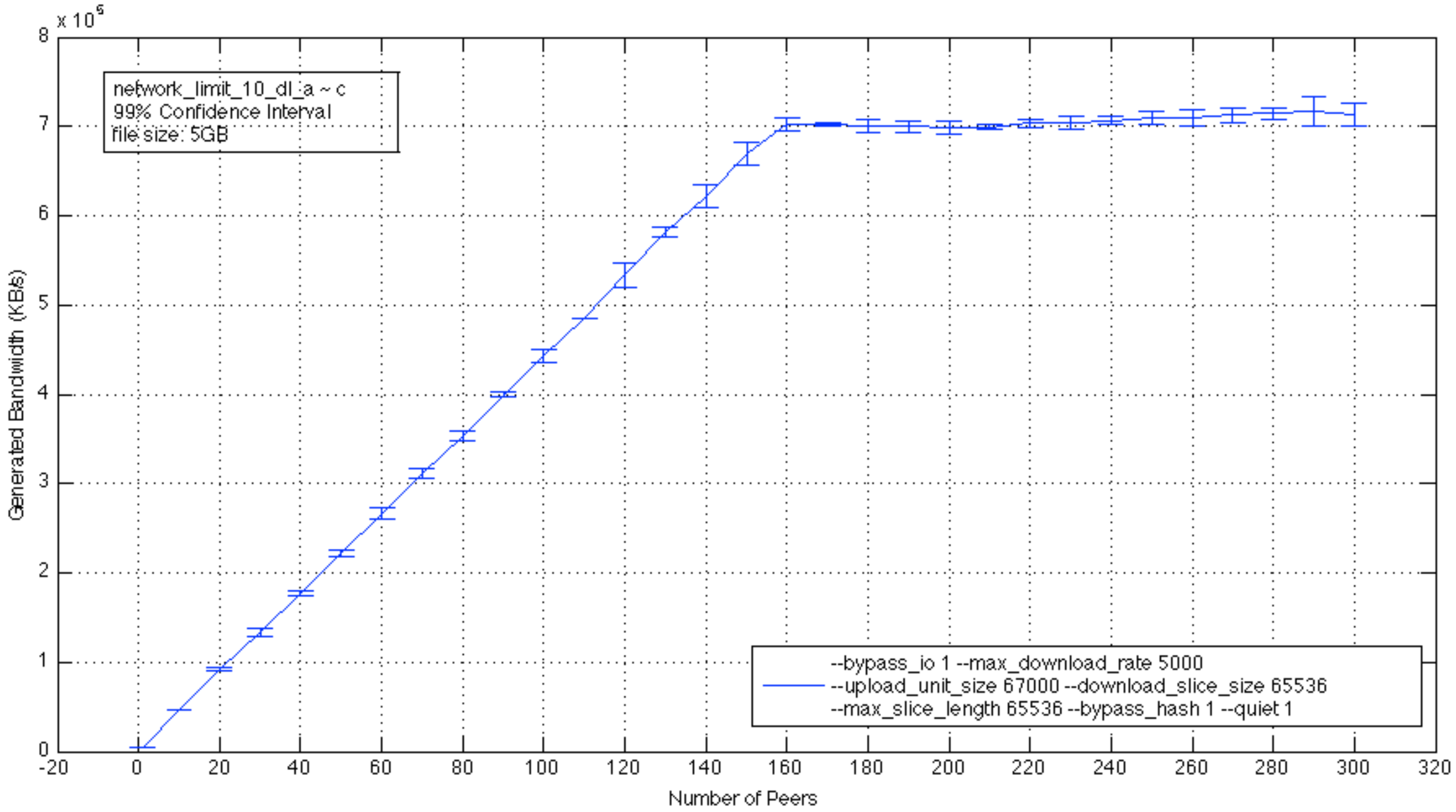


Some practical issues (contd.)

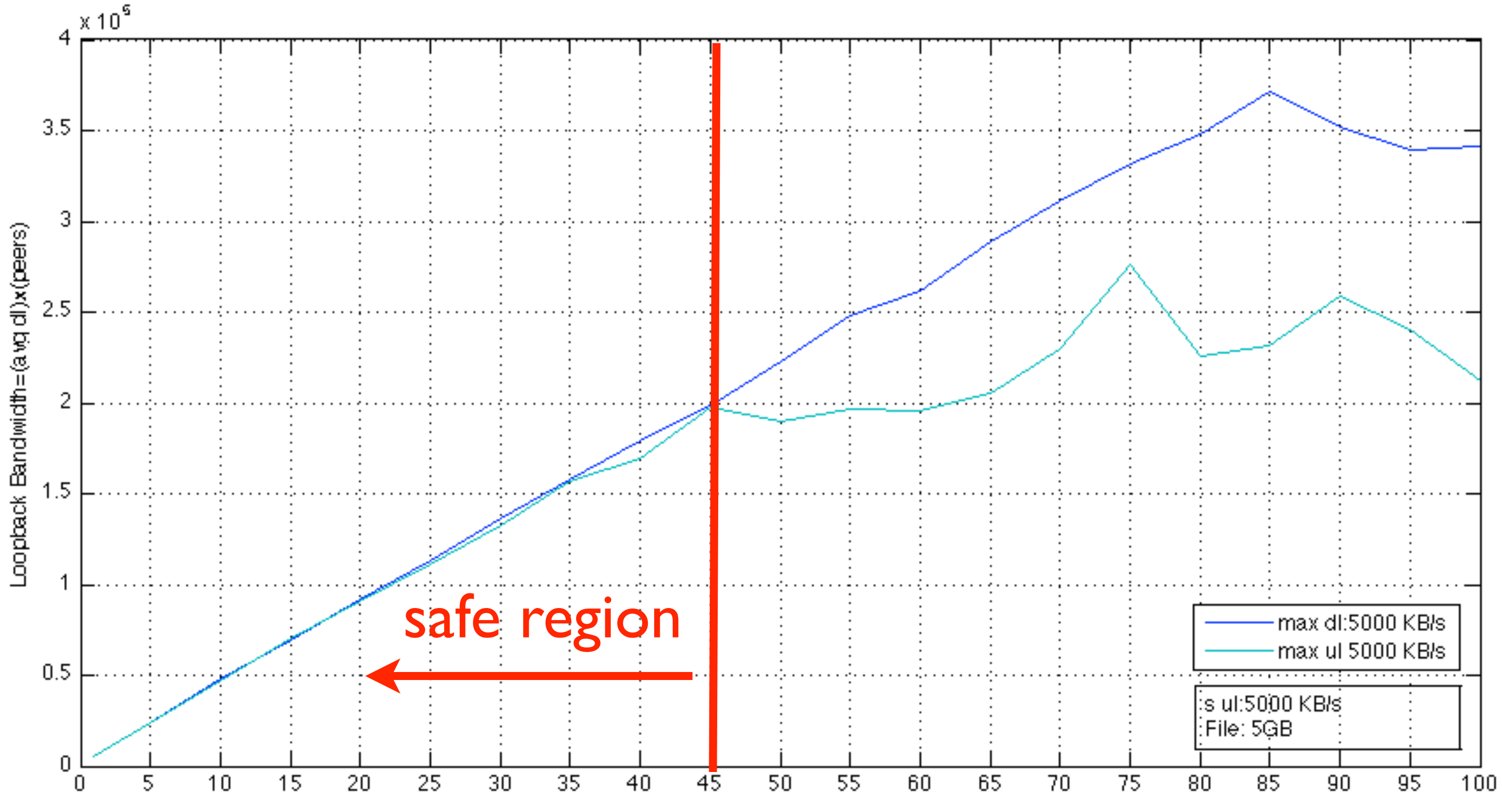
- Tune the parameters
  - the default parameters may work well on a home connection with low bandwidth. But some of them are not suitable on a high performance cluster.
  - Sending buffer(reduce write operations to network interface), slice size (reduce read operations). Control the number of concurrent uploads, which is calculated from the upload rate.
- Other Restrictions
  - For example, ip_local_port_range = 32768 ~ 61000 (28232 available)
  - CPU, memory, max sockets, max opened file, max processes, etc.

Some practical issues (contd.)

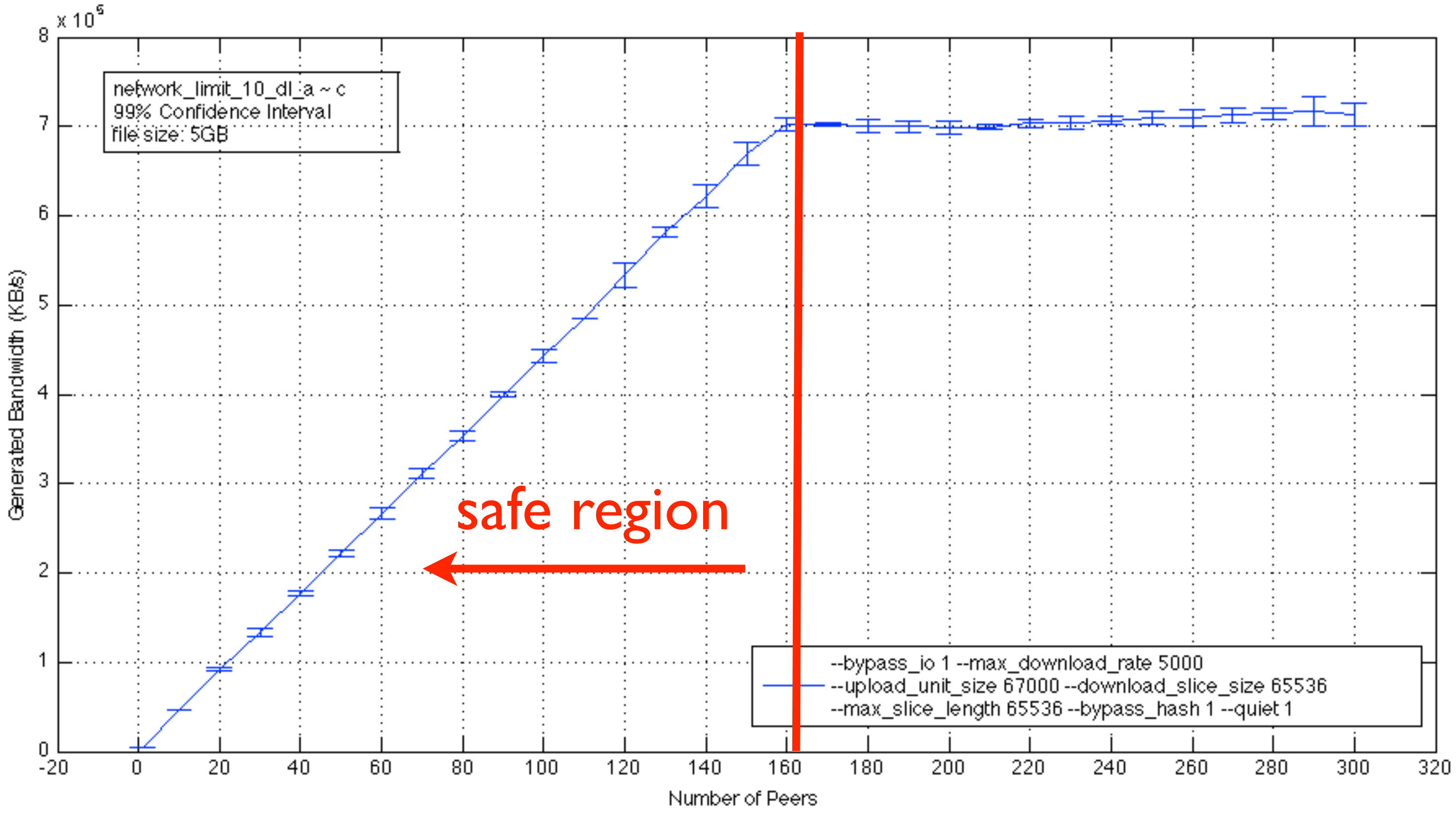




Some practical issues (contd.)



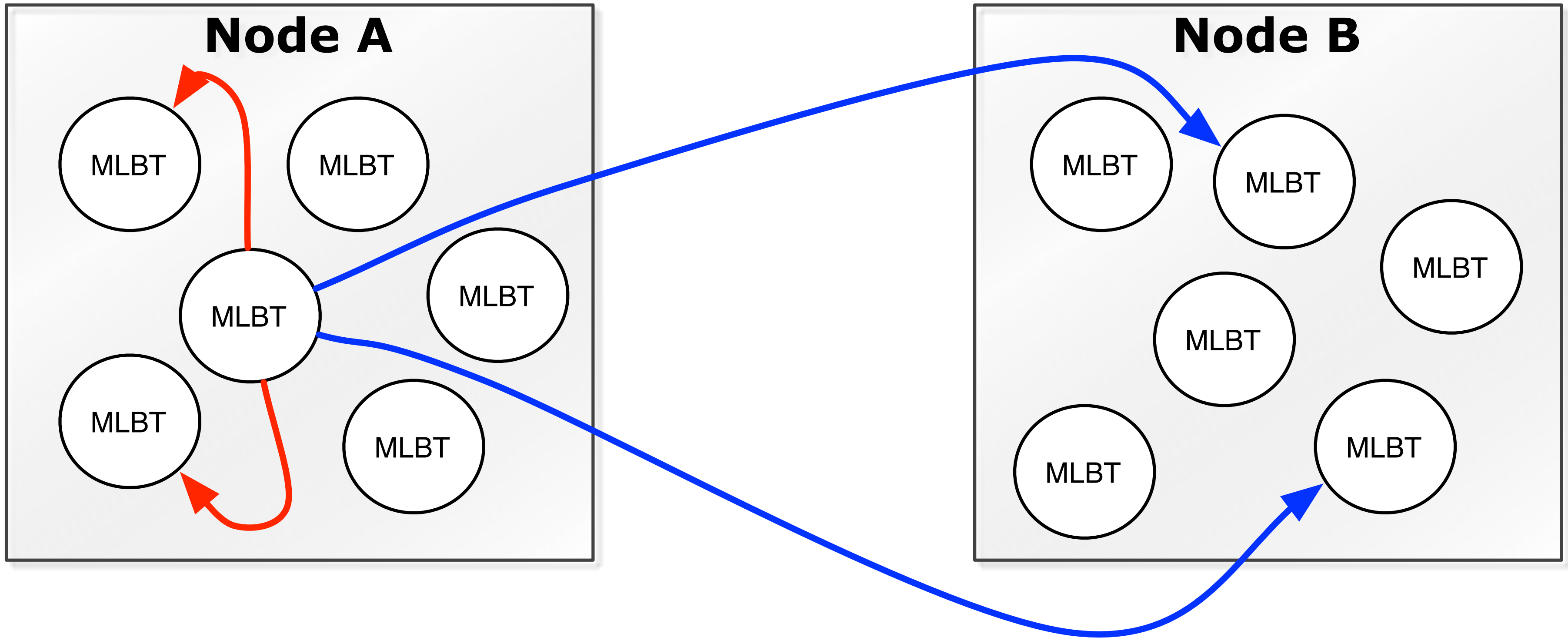
safe region



safe region

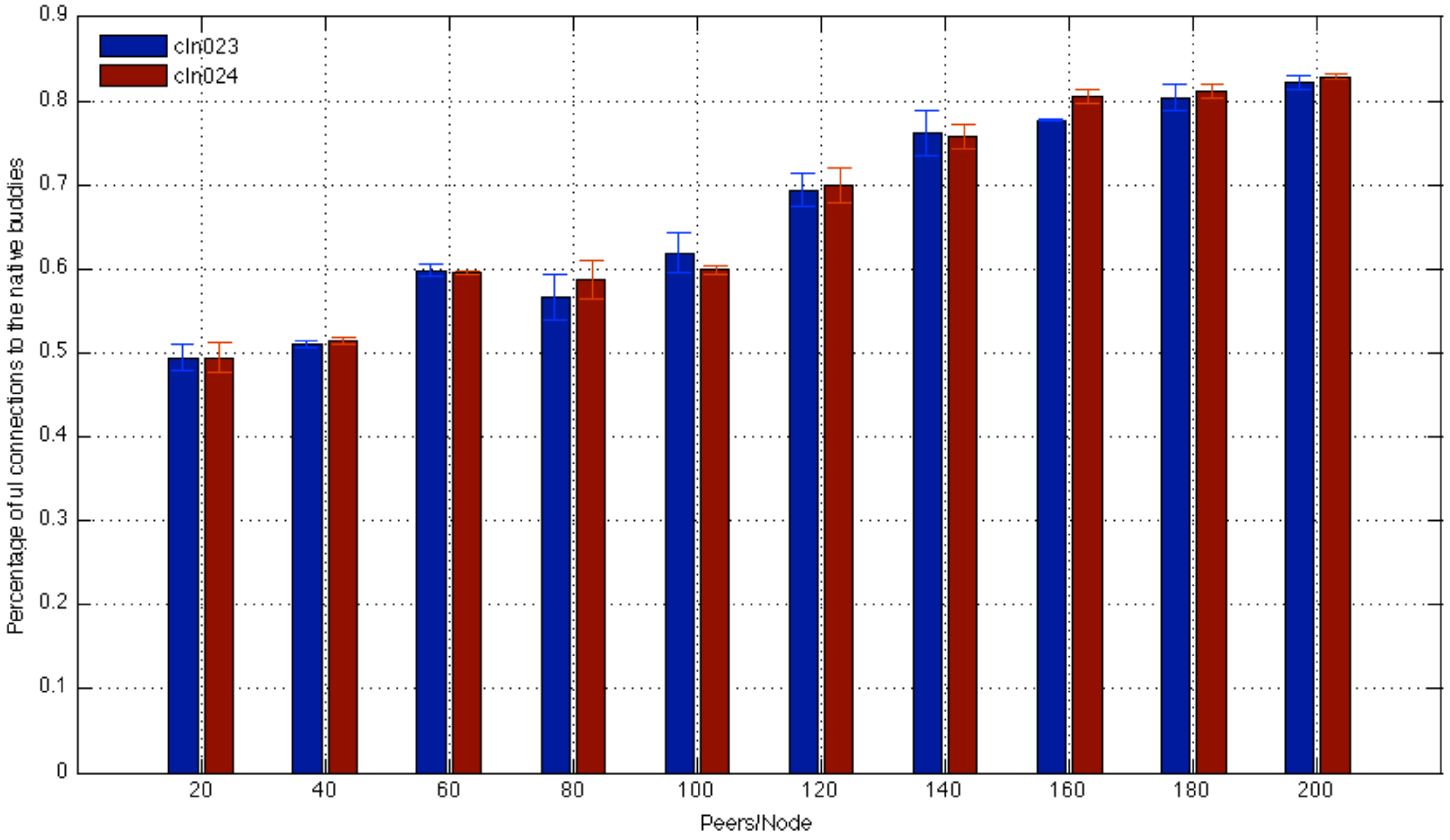
Two-node experiment

- Homogeneous experiment, all MLBT with same configurations
- Two types of experiments, upload-constrained & download-constrained
- Two types of outgoing connections, connections to the native peers & connections to the foreign peers



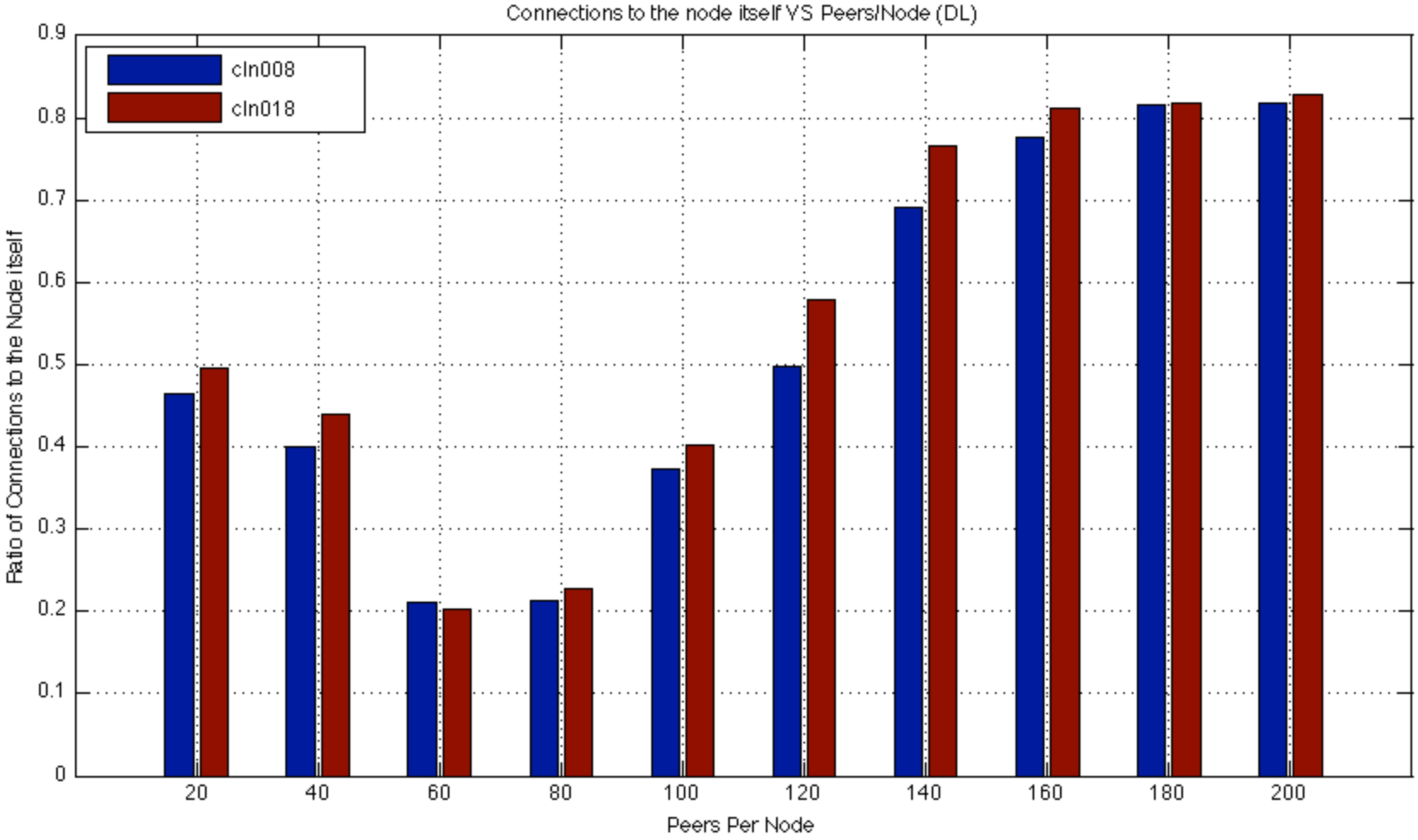
Change in BT’s behaviors

- Two-node experiment: upload-constrained



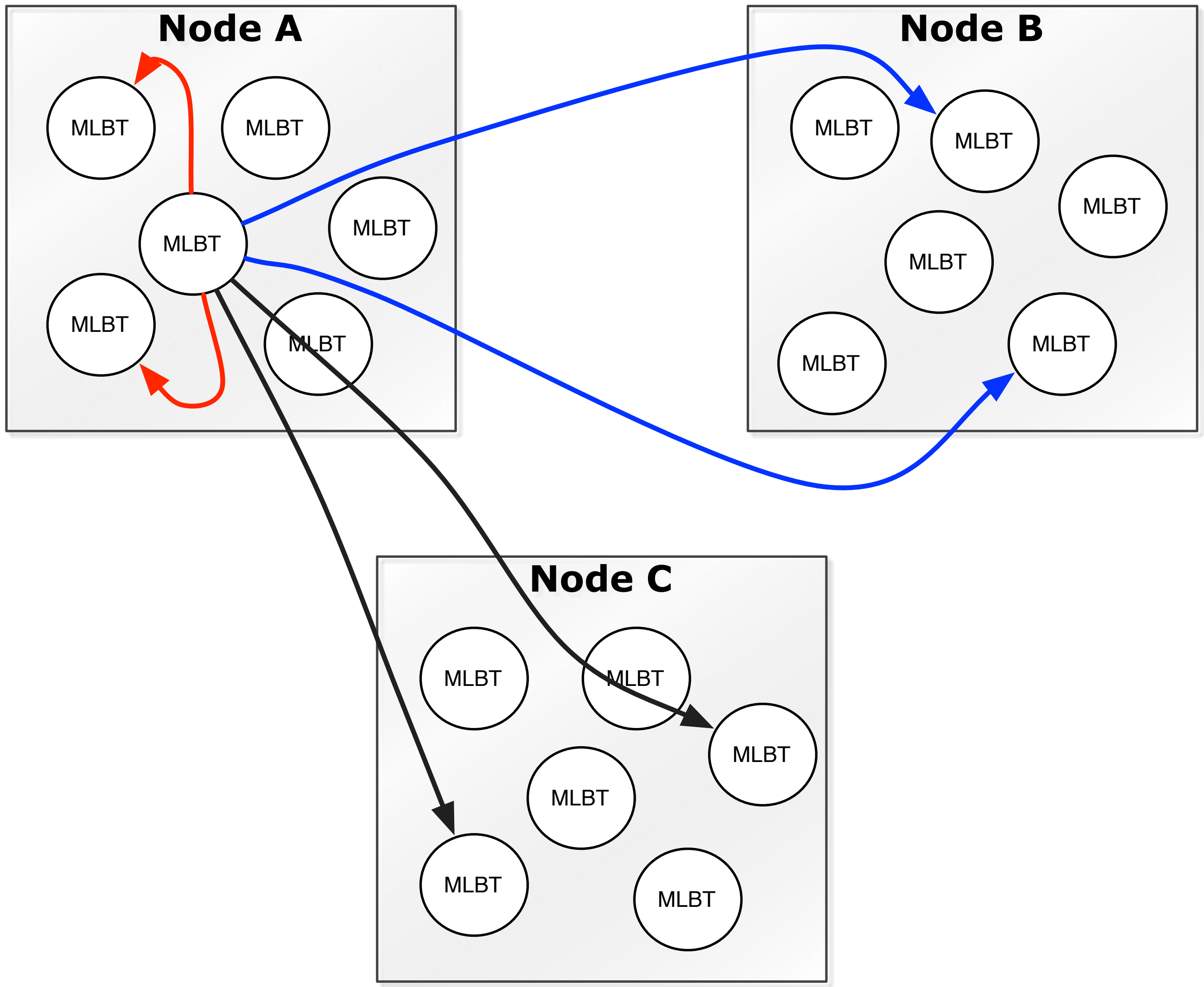
Change in BT’s behaviors

- Two-node experiment: download-constrained



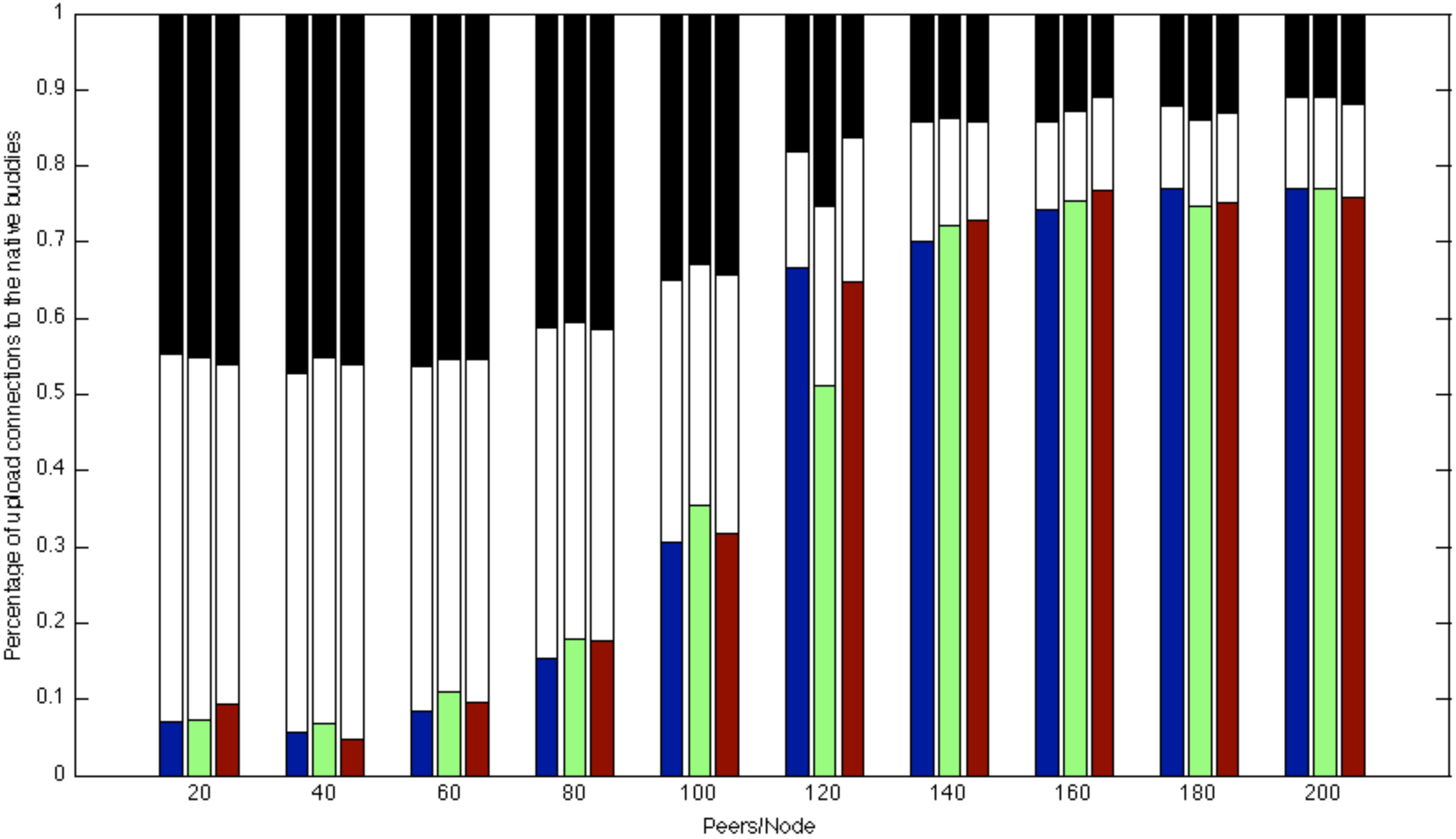
How about three nodes?

- Homogeneous experiment, all MLBT with same configurations



Change in BT's behaviors

- How about 3 nodes? (download-constrained)



Conclusion

- To experiment on a cluster, we must consider
  - Experiment target. (protocols and implementations)
  - Platform configurations and limitations. (depends on the underlying os)
  - Network configurations and topology.
  - Many things can be the bottlenecks, so the experiment should be carefully designed!

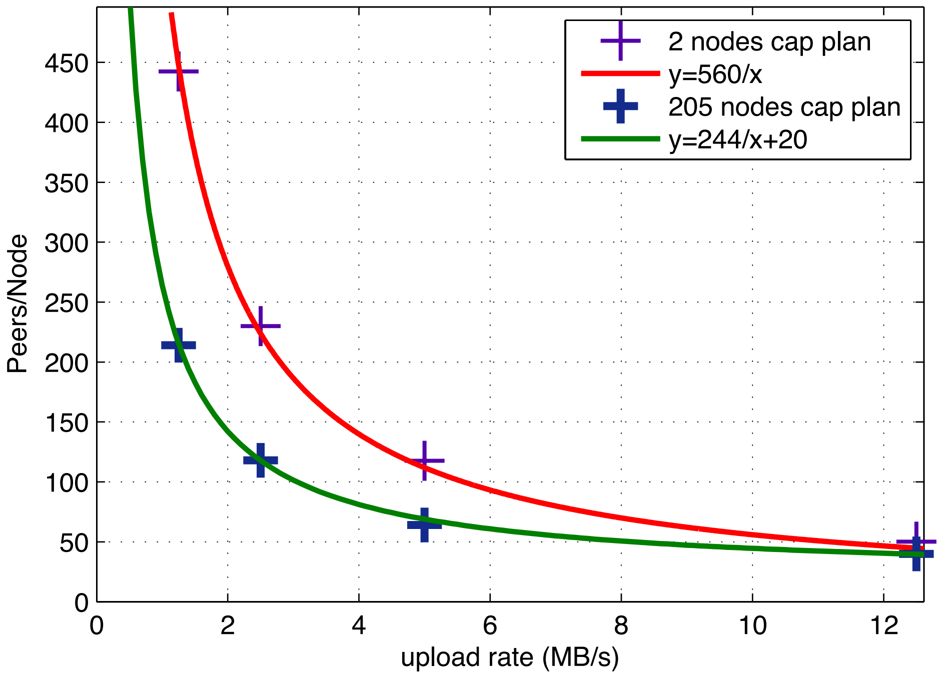
Conclusion (contd.)

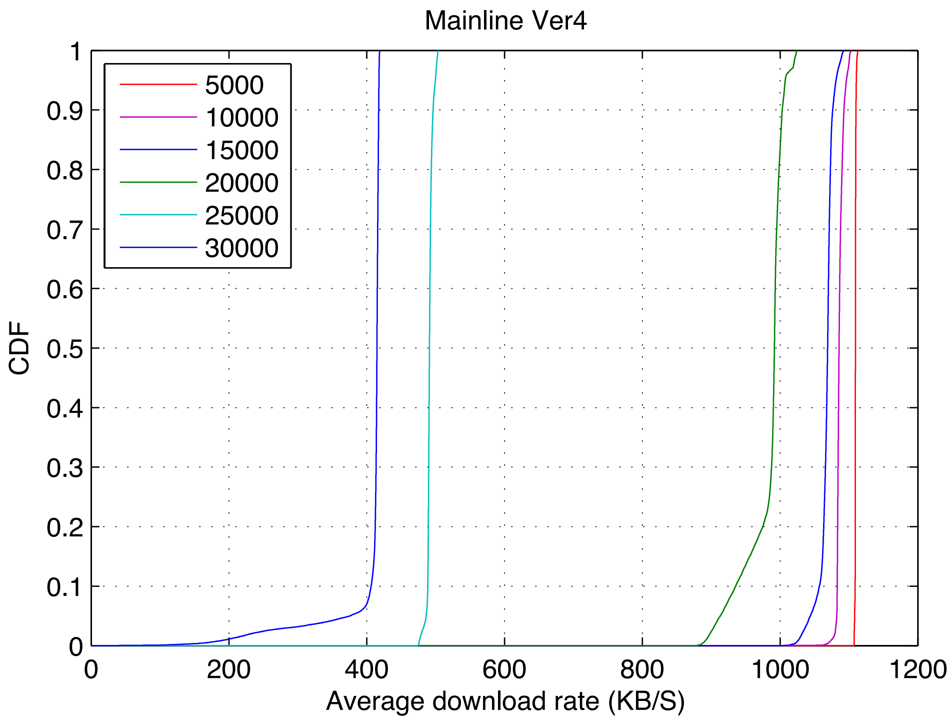
- Any other conclusions here?
  - It seems experimenting on a cluster is “dangerous”, too many underlying details, too many hackings, too many restrictions can mess up an exp.
  - Don’t forget the benefits from the cluster!
- It is feasible, but we need to be very careful.
  - Always, or at least try to know every underlying details.
  - Always design rational experiment.
  - Always play in the safe area.

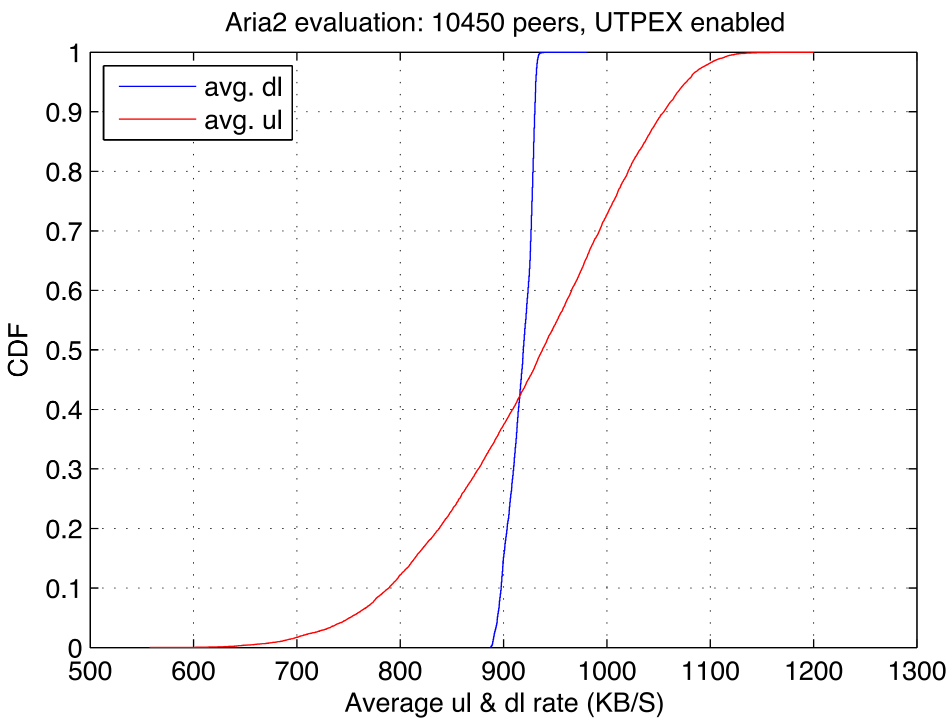
Thank you!

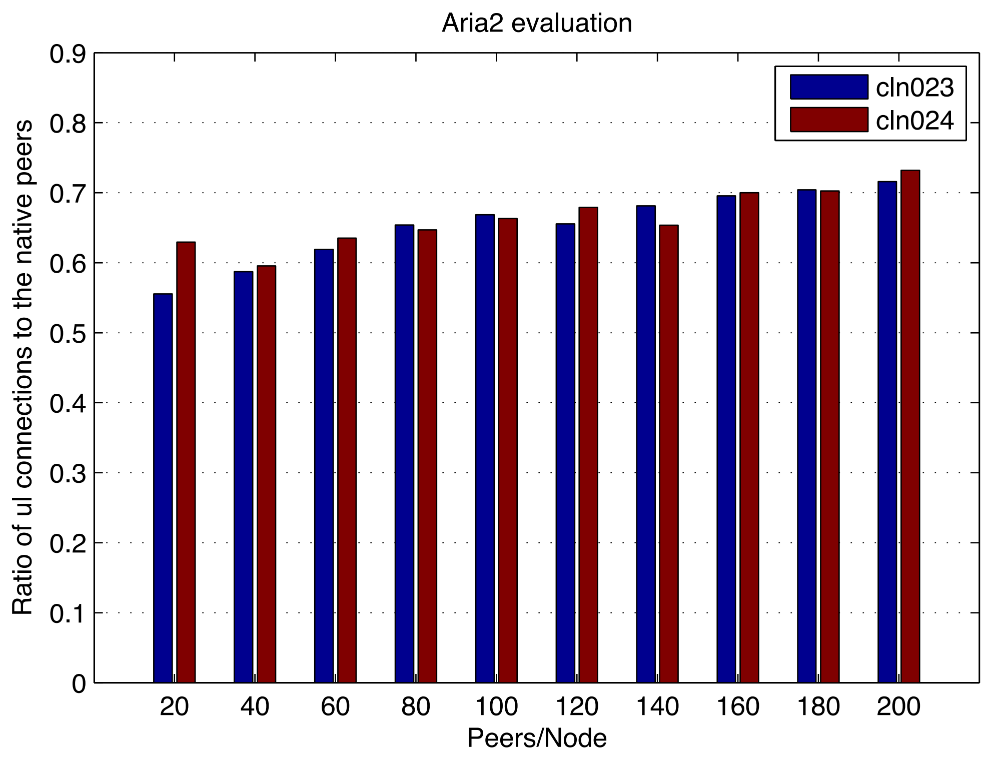
Liang Wang, Dept. of Computer Science

Extra figure of exp on Ukko


2 nodes cap plan
y=560/x
205 nodes cap plan
y=244/x+20
Peers/Node
upload rate (MB/s)


Mainline Ver4
CDF
Average download rate (KB/S)


Aria2 evaluation: 10450 peers, UTPEX enabled
avg. dl
avg. ul
CDF
Average ul & dl rate (KB/S)


Aria2 evaluation
cln023
cln024
Ratio of ul connections to the native peers
Peers/Node