Quasi-Dynamic Network Model Partition Method for Accelerating Parallel Network Simulation

Hiroyuki Ohsaki
Gomez Oscar
Makoto Imase

Graduate School of Information Science & Technology
Osaka University, Japan

Contents

Background

- Increasing size and complexity of the Internet
- Demand for evaluation technique of **large-scale networks**
- Strongly required to...
  - Ensure reliability, safety, and robustness
  - Allow future network expandability
  - Assess impact of terrorism and natural disasters

Conventional Techniques for Performance Evaluation

- Analysis techniques
  - e.g., Queuing theory
    - # of states exponentially increases as # of nodes increases
- Simulation techniques
  - A huge amount of computing resources is required
- Both techniques are...
  - **Not applicable** to large-scale networks

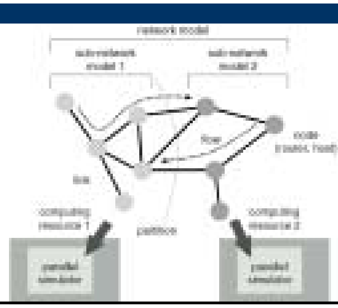
Parallel Simulation

- May allow simulation of large-scale networks
- Network simulators that support parallel simulation
  - QualNet, OPNET
    - Run on a single SMP computer
    - Not run on multiple computers
  - PDNS (Parallel Distributed NS)
    - Run on multiple computers
    - Have limited features

Research Objective

- Accelerate parallel network simulation by proposing **a network model partition method**
    - QD-PART (Quasi-Dynamic network model PARTition method)
    - Minimize communication overhead among computing resources
    - Balance loads of computing resources

Network Model Partition Overview

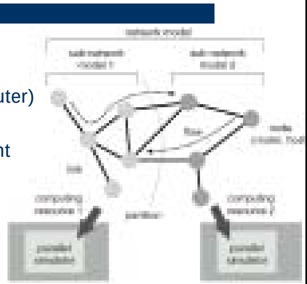


Basic Idea of QD-PART

- In many network simulation studies…
  - A network simulation is typically **repeated several times with the same parameter set**…
    for estimating the confidence interval of steady state measures
  - Partition of a network model can be gradually optimized based on **past simulation results**
    - Total simulation time
    - CPU usage of computing resources
    - Traffic intensity (i.e., # of packets transmitted)

QD-PART Algorithm: Notation

- Network model
  - G = (V,E)
  - V: node (host, router)
  - E: link
  - w(i,j): edge weight



QD-PART Algorithm: Step 1

- 1. Make initial partition
  - Assume all links have **the same traffic intensity**

$$w(i,j) = \frac{1}{\tau(i,j)^{\alpha}}$$

control parameter (α); propagation delay ($\tau(i,j)$)

  - Apply a graph partition algorithm METIS [7]
    - Results in **N sub-graphs** G1…GN
  - Perform parallel simulation and measure statistics
- 2. Make second partition based on traffic intensity
- 3. Improve partition using measured CPU usage

QD-PART Algorithm: Step 2

- 1. Make initial partition
- 2. Make second partition based on traffic intensity
  - Take account of the **measured traffic intensity**

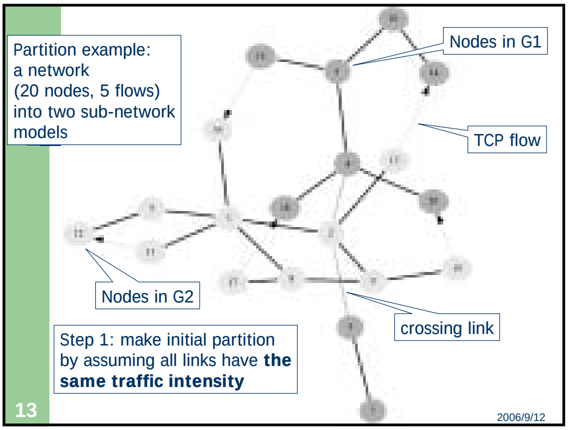
$$w(i,j) = \frac{I_t(i,j)}{\tau(i,j)^{\alpha}}$$

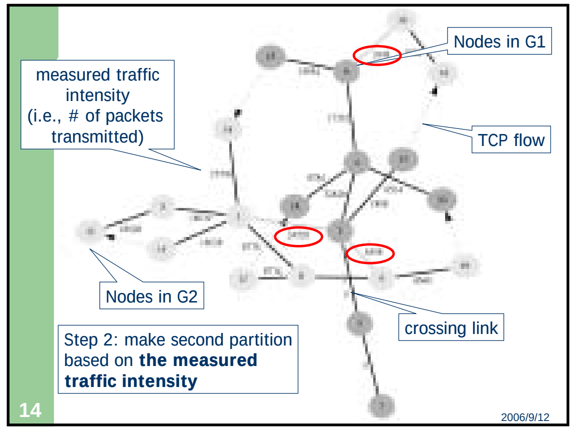
traffic intensity (e.g., # of packets transmitted)

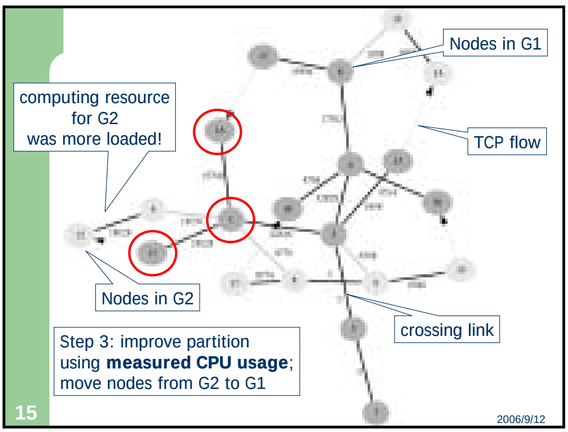
  - Apply a graph partition algorithm METIS [7]
    - Results in **N sub-graphs** G1…GN
  - Perform parallel simulation and measure statistics
- 3. Improve partition using measured CPU usage

QD-PART Algorithm: Step 3

- 1. Make initial partition
- 2. Make second partition based on traffic intensity
- 3. Improve partition using **measured CPU usage**
  - Move boundary nodes…
    from the **most loaded** computing resource
    to the **least loaded** computing resource
  - Perform parallel simulation and measure statistics
  - If the total simulation time is reduced…
    - Repeat step 3







Experiment Setup

- 2 computing resources
  (partition into two sub-network models)
  - Intel Xeon 2.4GHz with 1,024MB memory
  - Linux 2.4.30
  - PDNS version 2.27-v1a
  - 1G Ethernet

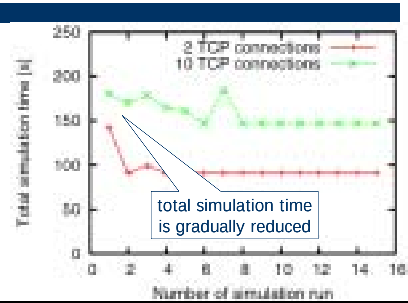
Simulation Model

- Network model
  - Number of nodes: 20
  - Number of links: 20
  - Link bandwidth: 1 or 0-1 [Mbit/s]
  - Link propagation delay: 1 or 0-1 [ms]
- Workload
  - # of persistent TCP flows: 2 or 10

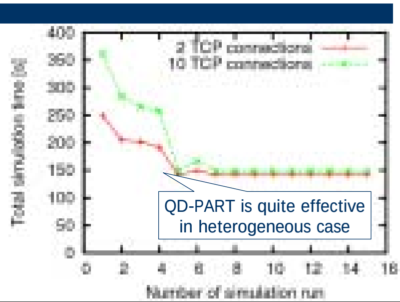
homogeneous case

heterogeneous case

Total Simulation Time vs. # of Simulation Run (Homogeneous Case)



Total Simulation Time vs. # of Simulation Run (Heterogeneous Case)



Conclusion

- Proposed a network model partition method QD-PART
  - To accelerate parallel network simulation
- QD-PART…
  - Utilizes the fact that a network simulation is typically **repeated several times**
  - Re-partitions the network model based on **past simulation results**
  - **Significantly reduces** the total simulation time

Future Works

- Through **performance evaluation** of QD-PART
  - Other types of network models
  - More computing resources
- Extend QD-PART to support **Grid environment**
  - Heterogeneous computing resources
  - Heterogeneous networking resources