Systems Modelling & Simulation

Grid Computing (M)

Lecture 18

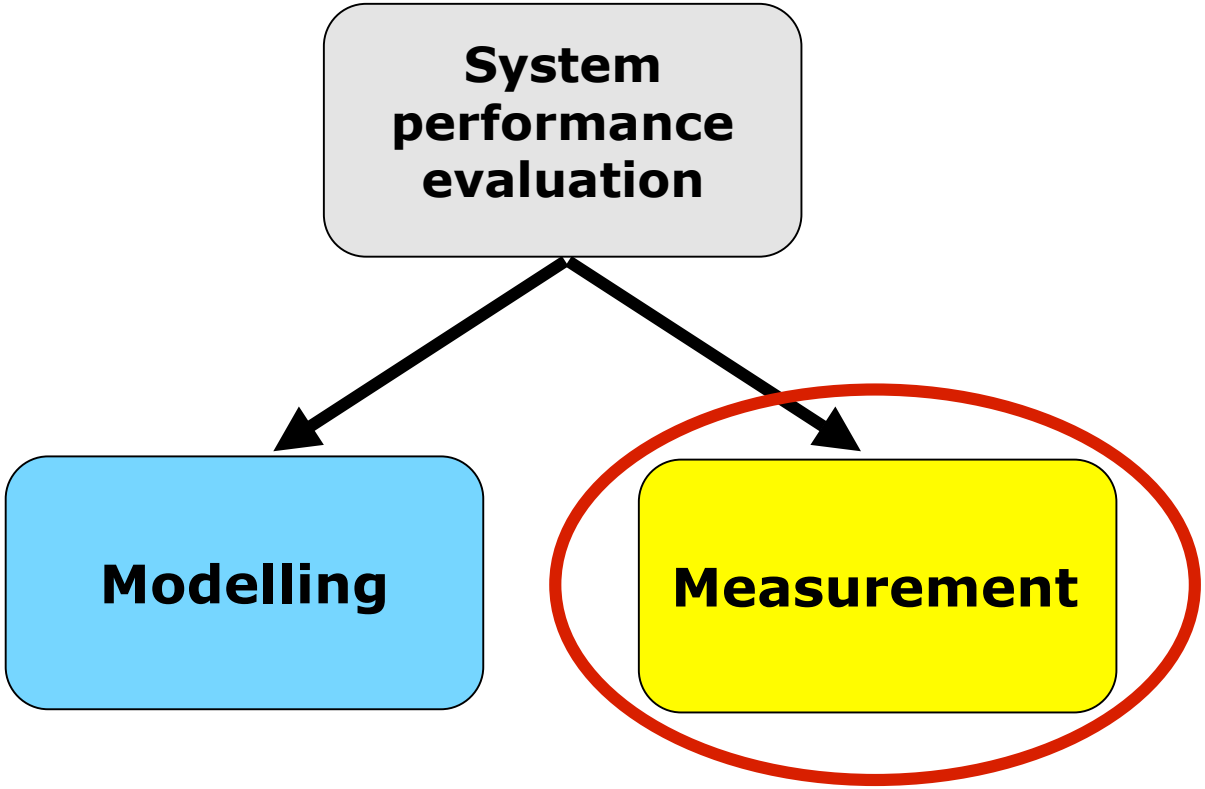
Joe Sventek (joe@dcs.gla.ac.uk)

March 2006

UNIVERSITY
of
GLASGOW



System Performance Evaluation Choices



Copyright © 2006 University of Glasgow

Recall that Measurement may be …

- **Disruptive**
  - Measuring phenomena without changing it?
    - Heisenberg Uncertainty Principle?
    - Does a tree falling in the forest make any noise if no one is there to hear?
- **Dangerous**
  - Typically interested in "failure conditions"/"stress testing"
    - How quickly will building be evacuated in a large fire?
    - At what depth will water pressure crush submarine hull?
    - At what traffic load will DCS network "collapse"?
- **Expensive**
  - Typically requires specialised equipment, lengthy measurement periods, significant people-hours etc.
- **Impractical**
  - Insurmountable technical, logistical, ethical, legal etc. barriers
- **Ad hoc**
  - How applicable is specific measurement in generic case?
- **Impossible**
  - System not yet built/implemented/deployed
    - i.e. wanting to justify system design ⇒ ***predict*** performance

Copyright © 2006 University of Glasgow

Focus on Network Performance Measurement

- Three types of measurement:
  - Passive
    - Measure the performance characteristics of traffic generated by real users
    - Depends upon which observation points have been chosen
  - Active
    - Inject traffic into the network
    - Measure the performance characteristics of the injected traffic
    - Infer characteristics of real traffic in the temporal neighbourhood of the injected traffic
    - Injected traffic may be routed differently than real traffic
  - In-line
    - Take advantage of extension headers provided by IPv6
    - Inject measurement information into extension headers
    - Guaranteed that the injected information is routed the same way as real traffic, since it is attached to real traffic
- Both active and in-line measurements can perturb the real traffic whose characteristics you are trying to determine
- Passive and Active Measurement Workshops (PAM 2000-2005)
  Internet Measurement Conference (IMC 2003-2005)

Copyright © 2006 University of Glasgow

Typical methodologies

- Off-line – analyze after the traffic has occurred
  - Perform active/passive/in-line measurements to yield packet traces
  - Analyze traces for characteristics of interest, usually looking into various statistical aggregate measures of the traffic
  - The traces are usually stored in Packet CAPture format (PCAP) to enable the use of a number of tools that have been developed in the community
- On-line – traffic analysis occurs in real time as the packets are captured
  - Libpcap/Jpcap programming support (set interface into promiscuous mode, apply host/port filters in the kernel to reduce latency and packet loss)
  - Possible use of sophisticated interface hardware to reduce interrupt load on processor (Endace DAG cards)
  - TCPDump

Copyright © 2006 University of Glasgow

Examples of support on the web

- NLANR - http://www.nlanr.net/
  - National Laboratory for Applied Network Research
  - Provides technical, engineering, and traffic analysis support of NSF High Performance Connections sites and HPNSP (high-performance network service providers) such as the NSF/MCI very high performance Backbone Network Service (vBNS)
  - The Measurement and Network Analysis team, located at UCSD/SDSC, conducts performance and flow measurements for HPC sites.
- CAIDA - http://www.caida.org/
  - The Cooperative Association for Internet Data Analysis
  - Provides tools and analyses promoting the engineering and maintenance of a robust, scalable global Internet infrastructure
- Network monitoring tools from Les Cottrell at SLAC - http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html
  - A set of Network Monitoring Tools that has been collected from system administrators in the high-energy physics community. Default starting point for grid-specific performance measurement tools.

Copyright © 2006 University of Glasgow

An Example

- Characterization
  - Use of peer-to-peer applications is becoming more prevalent in the Internet to support file sharing and other application areas.
  - P2P applications now dominate the bandwidth usage of the Internet, having overtaken web browsing applications.
  - The bandwidth consumed and (sometimes) illegal nature of P2P usage leads to a requirement from Internet Service Providers for the ability to detect the onset of P2P behavior.
  - Most applications that use the Internet exhibit particular packet patterns at the network (IP) level.

Copyright © 2006 University of Glasgow

An Example (cont)

- Hypothesis
  - P2P applications generate unique packet patterns that can be detected at the network level by monitoring traffic at a single point in the network
- Prediction
  - Packet patterns can be discovered by generating time-synchronized traces of application behavior and IP packets resulting from said application behaviour.
  - Invariant packet patterns can be discovered by comparing traces taken at each end of the path between two P2P applications.
  - These invariant patterns can be detected in real time at a single point in the network.

Copyright © 2006 University of Glasgow

The Experiment

- What is our independent variable? The choice and configuration of a P2P application whose traffic will be traced.
- What are the variables outwith our control? The other IP traffic with which our P2P traffic will compete.
- What is the meaning of invariant? The pattern is independent of delays induced by uncontrollable traffic, small percentages of packet loss, and the point in the network where the measurements are made.
- What do we mean by "detect the pattern"? We should be able to determine that the said pattern has occurred AND it was associated with P2P traffic; in particular, it is essential that we have no false positives.
- Which applications do we choose? We categorize P2P applications into equivalence classes, choosing an exemplar from each class.

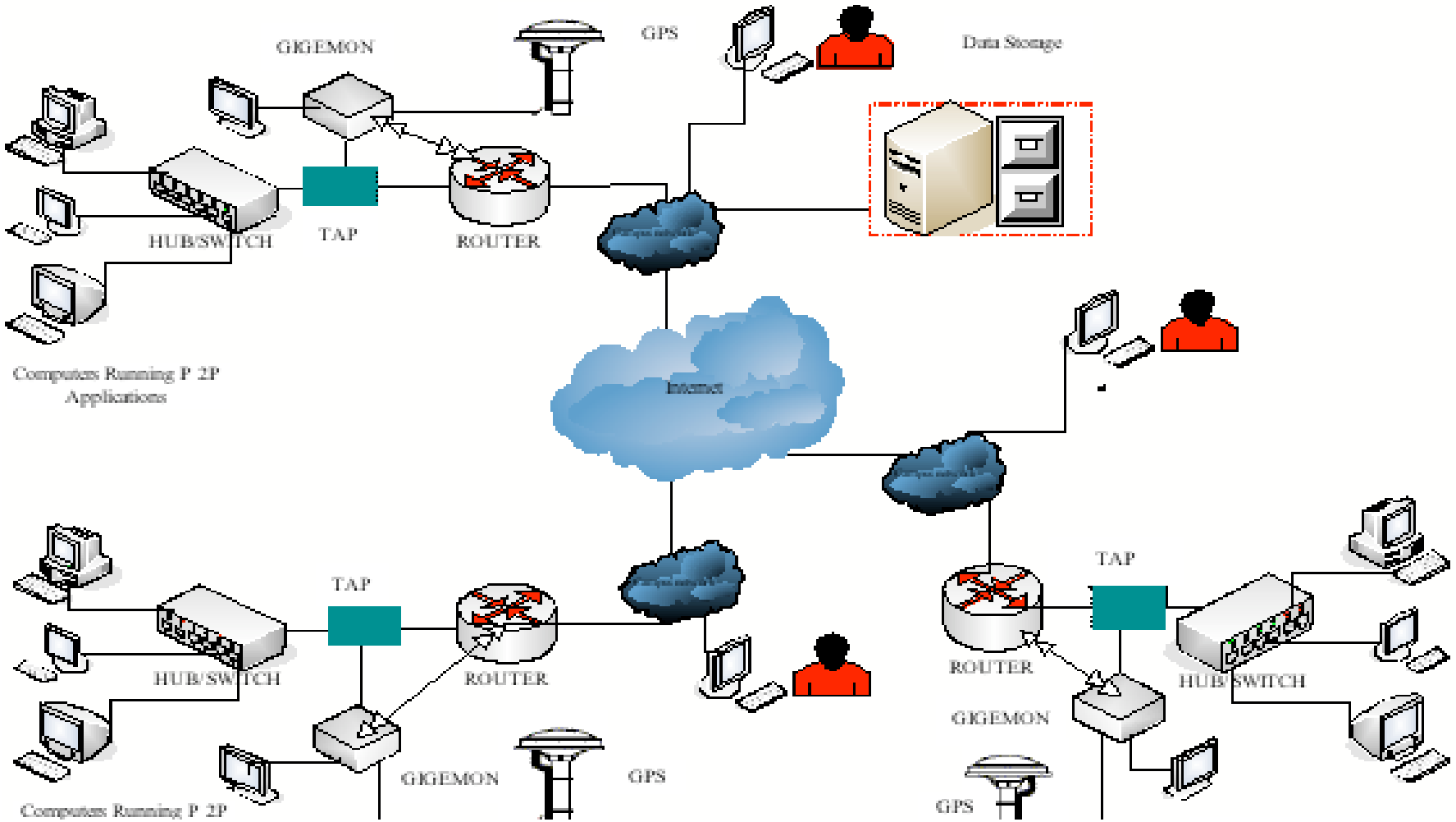
Copyright © 2006 University of Glasgow

The Experiment (cont)

- Equivalence classes of P2P applications
  - File sharing (e.g. KaZaA, Napster, Gnutella, eDonkey)
  - Streaming media
  - Collaborative working
  - Software distribution
  - Data management
- Experimental design
  - Create a closed environment in which the chosen applications can be run
  - Force all traffic between chosen applications to go through network measurement points so that the packets can be traced
  - Instrument chosen applications to log application-level communication activity.

Copyright © 2006 University of Glasgow

The Experiment (cont)



Copyright © 2006 University of Glasgow

The Experiment (cont)

- Experimental scenario
  - Decide upon the number and location of instances of a particular P2P application
  - Decide upon the application-level commands that will be issued for this experimental run
  - Stage any needed data to the host systems upon which the P2P applications will execute
  - Instruct the packet monitors (GIGEMONs) to start tracing IP packets at a particular start time
  - Instruct the host systems to start the application instances after the start time, giving the GIGEMONs enough time to reach steady state
  - Run the applications for a fixed amount of time, or until the packet monitors have exhausted their disk space
  - Archive the packet-level and application-level traces for subsequent analysis

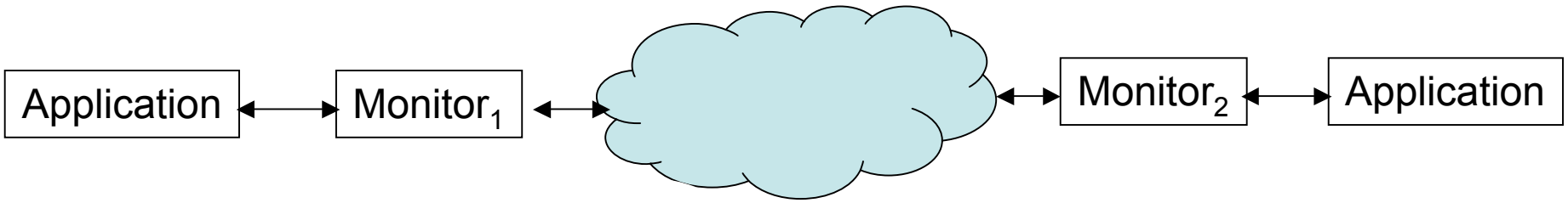
Copyright © 2006 University of Glasgow

The Experiment (cont)

- How can this structure yield invariant patterns?
  - Suppose the application uses a known file transfer protocol, such as FTP; we simply look for the control packets associated with the setup of an FTP session on the known port associated with FTP; seeing this sequence of packets is independent of the load on the network; if packet loss is encountered in the control traffic, we will see the retries
  - Suppose the application uses a known port number; seeing traffic to/from that port number is independent of the load and packet loss
  - Suppose that neither of these hold; we must perform various statistical analyses of the IP traces at both ends

Copyright © 2006 University of Glasgow

The Experiment (cont)



- Analyze trace from Monitor$_1$ to detect set of patterns, $\{p_{i1}\}$
- Analyze trace from Monitor$_2$ to detect set of patterns, $\{p_{k2}\}$
- Invariant patterns are the intersection of the two sets, $\{p_{i1}\} \cap \{p_{k2}\}$

Copyright © 2006 University of Glasgow