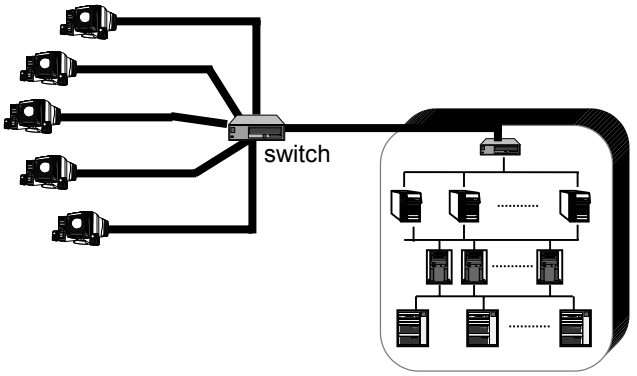
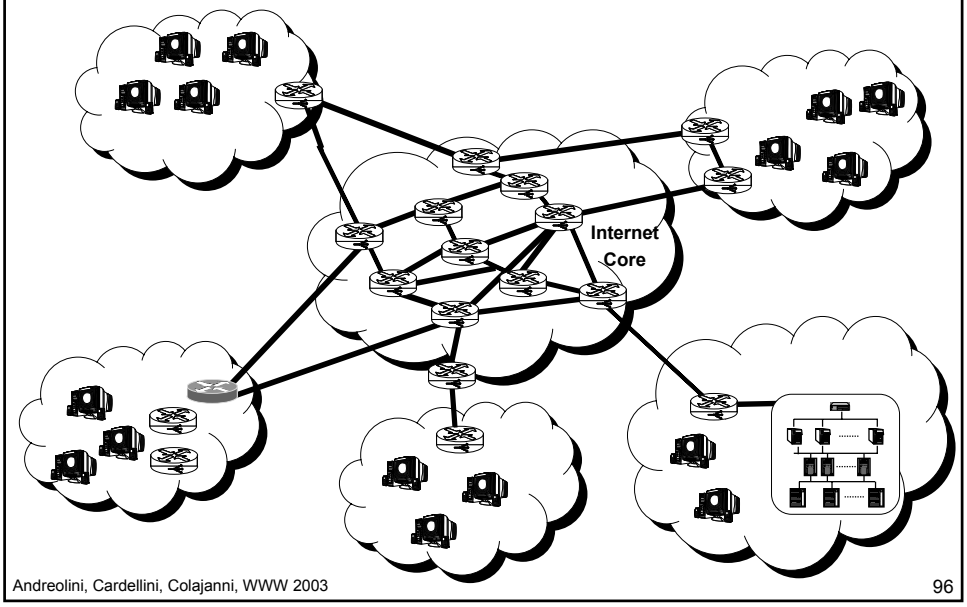
Part 4
Impacts of WAN Effects

Typical benchmarking environment



- **Problem:** real Internet does not work this way!
- **Network-related factors are not considered:** delays, transmission errors, packet losses, packet duplications and re-ordering, bandwidth limitation, caching effects, user's STOP factors, …

Web cluster in the real environment

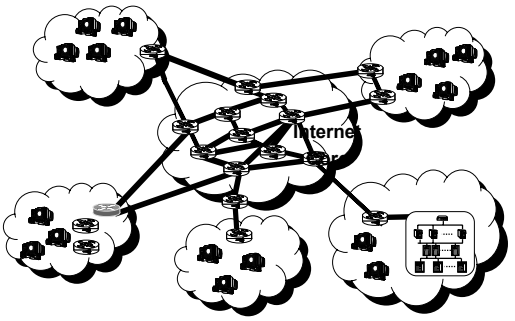


Benchmarking with WAN effects

- **Two approaches**
  - WAN environment
  - WAN emulation in a LAN environment
- **WAN environment**
  - Client machines are spread in a WAN
  - Disadvantages:
    - Not reproducible and configurable environment
    - Difficult to generate a very large workload
- **WAN emulation**
  - Centralized approach
  - Distributed approach
  - Simulation-based

⇐ Our focus

WAN environment



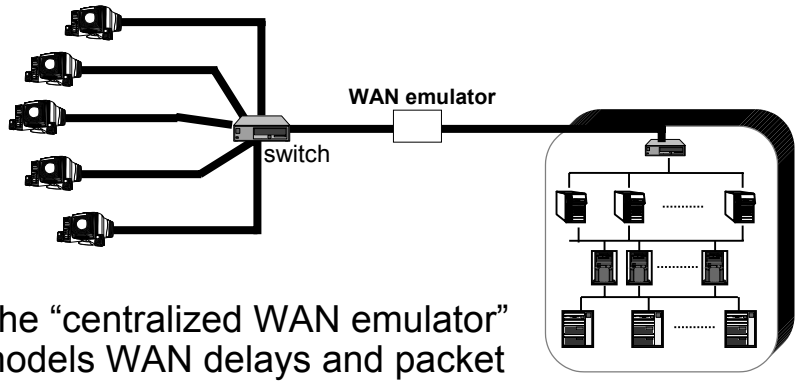
- Life server analysis captures real WAN conditions
- Network conditions are not reproducible and configurable
- Difficult to generate a very large workload
- Goal: an environment that is both realistic and reproducible
  ⇨ WAN emulation in a LAN environment

WAN emulation

- **Main goals of WAN emulation**
  - ***Realistic***: emulates WAN conditions
  - ***Reproducible***: allows iterative analysis
  - ***Configurable***: can vary many parameters
  - ***Scalable***: scales to very large workloads

- **A possible disadvantage: WAN emulation does not reproduce a specific network topology**
  - However, the effects of dynamic routes can be emulated by configuring the “packet effects” parameters

WAN emulation: Centralized approach

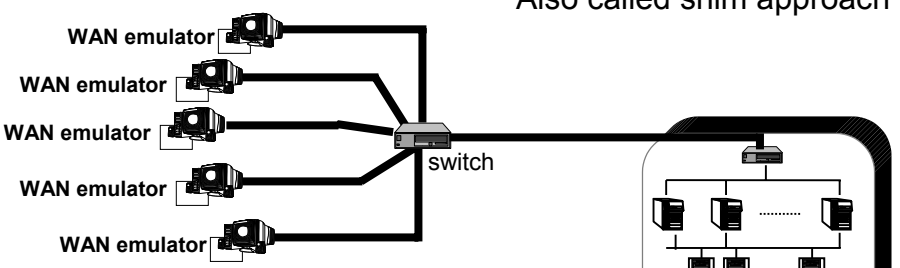


The “centralized WAN emulator” models WAN delays and packet losses (e.g., S-Client, Shunra products)

- simpler implementation
- bottleneck

WAN emulation: Distributed approach

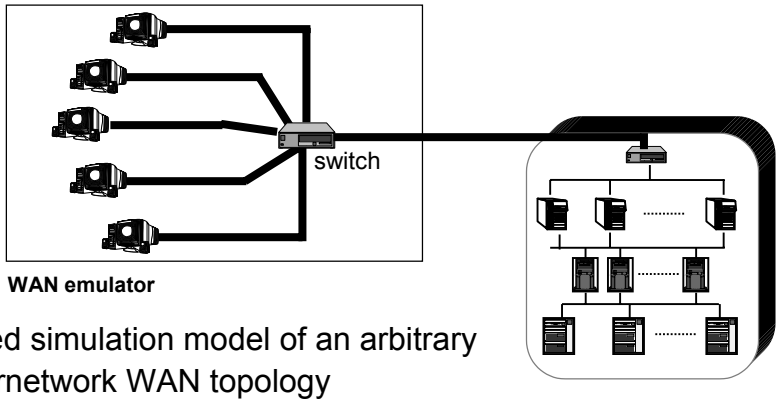
Also called shim approach



Each client acts as a WAN emulator (*e.g.*, NIST Net, DummyNet, WASP)

- scalable solution
- implementation requires modifications in the kernels of the clients

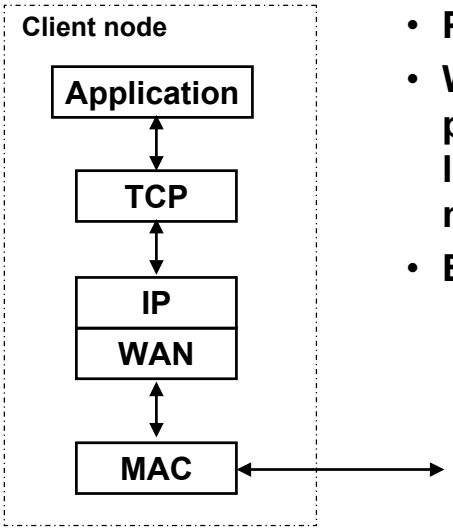
WAN emulation: Simulation-based



Detailed simulation model of an arbitrary IP internetwork WAN topology
(e.g., **IP-TNE** by Williamson et al.)

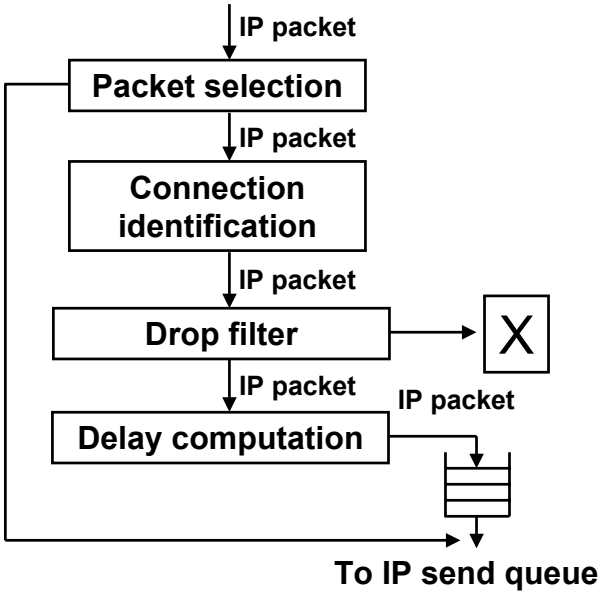
- WAN emulation through a single computer
- Gigabit Ethernet, multiprocessor machine to run the simulator

Our WAN emulator: architecture

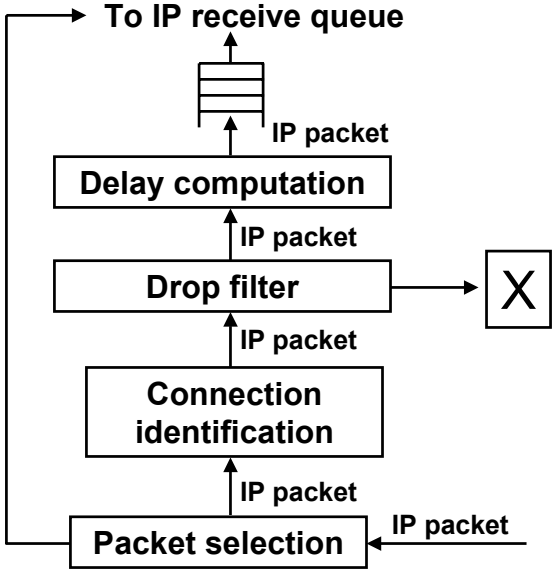


- **Patch to Linux kernel**
- **WAN emulation is performed under the IP level of each client node**
- **Emulated WAN effects:**
  - routing delay
  - packet loss
  - maximum device speed

Outbound packet flow



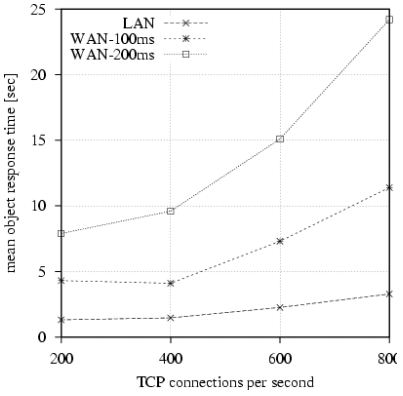
Inbound packet flow

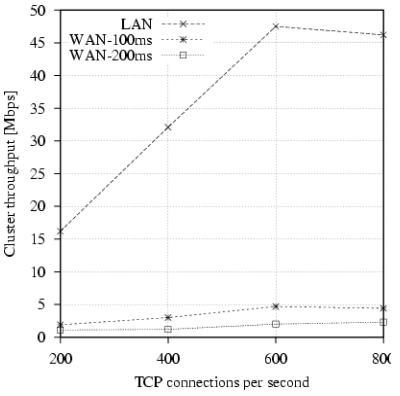


Testbed

- **Distributed approach to WAN emulation**
  - Each client runs httperf
- **Heavy-tailed workload**
  - User sessions, user think times
  - Web publishing-like workload
- **Impact of the following WAN effects**
  - Routing delay (without packet loss)
  - Packet loss (without routing delay)
- **Impact of different metrics**
  - Mean object response time
  - 90/95-percentile of object response time

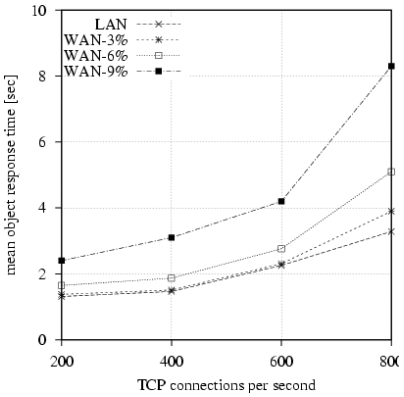
Example: *Routing delay*

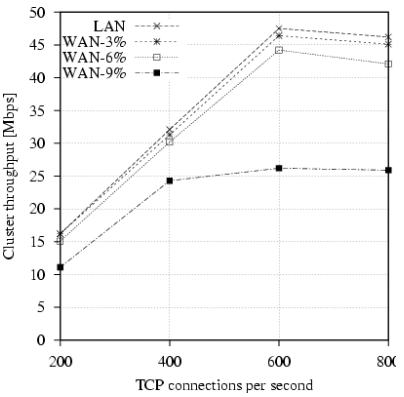




- Increasing routing delay
  - Aggregate cluster throughput decreases
  - Mean object response time increases

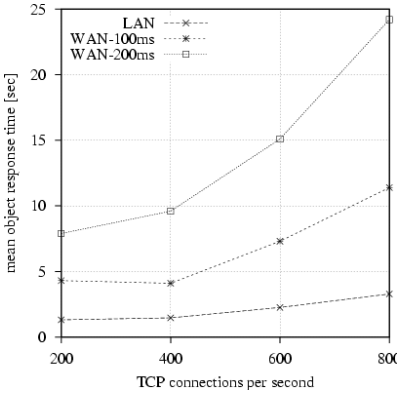
Example: *Packet loss*

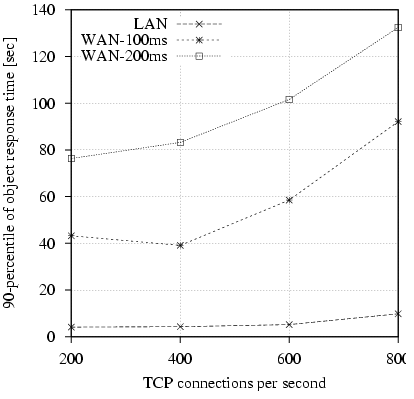




- Increasing packet loss
  - Aggregate cluster throughput decreases
  - Mean object response time increases
- Performance difference is less evident than with routing delay

Example: *Impact of different metrics*

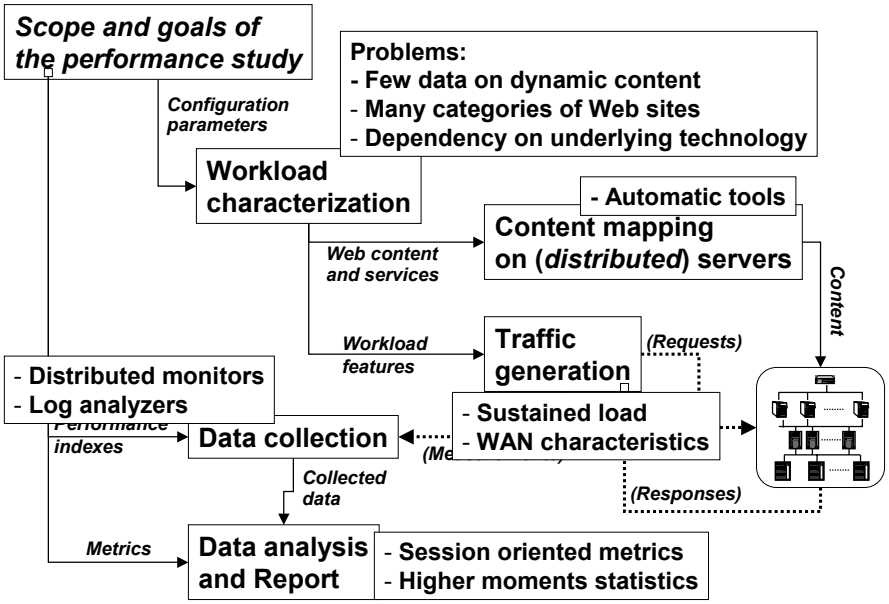




- Mean vs. 90-percentile of object response time in a delayed-routing scenario
- Due to the heavy-tailed workload, 90-percentiles of response times are an order of magnitude greater than the mean

Part 5
Summary and Research Perspectives

Main open problems



State of benchmarking tools

- **Static content:** ***Adult***
- **Dynamic content:** ***Teenager***
- **Dynamic content for distributed systems:** ***Infant***
- **Other Web-based services:** ***Unborn***

Research perspectives

1) **Web multi-clusters**
2) **Quality of Web-based services**
3) **Universal Web access**
4) **Web services**

(1) Web multi-clusters

- ***Web site realized on an architecture of geographically distributed Web clusters***
- **Web site addresses**
  - One hostname (e.g., "www.site.com")
  - One IP address for each Web cluster

First level scheduling

Authoritative DNS **or** other entity **during the lookup phase**

Second level scheduling

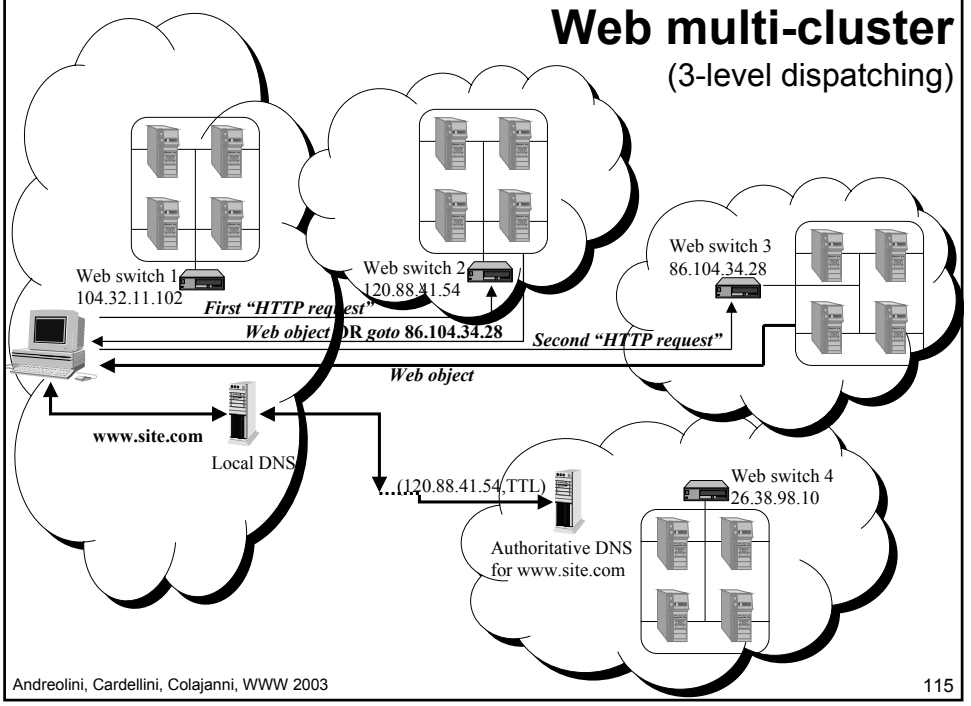
Web switch **of the Web cluster selects one server**

Third level scheduling

Each Web server **may redirect the received request to another server**

Web multi-cluster

(3-level dispatching)



Web client characterization for Web multi-clusters (*wish list*)

- **Support to *request routing mechanisms***
  - Not required for benchmarking of Web clusters
  - Required for geographically distributed Web systems
    - DNS address lookup
    - HTTP redirection
- **Capability of emulating IP-address resolution mechanism at the client**

(2) QoS in Web-based services

- ***Second generation* of Web sites**
  - communication channel for critical information
  - fundamental technology for information systems of the most advanced companies and organizations
  - business-oriented media
- **Some requirement for the *third generation***
  - differentiation of users and services
  - support to heterogeneous applications and user expectation
  - support to multi-class Service Level Agreements (**SLAs**)
  - differentiated pricing for content hosting and service providing

Quality of Web-based Services (QoWS)

High performance systems

≠

Systems for Quality of Service

QoS principles and mechanisms have been deeply investigated in the computer network area, but

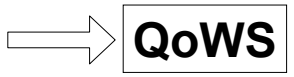
– QoS principles are not immediately applicable to the server side of the Web system

– **Network QoS** and **Server QoS** principles must be combined to provide an end-to-end QoS for Web-based services

An approach to enhance Web clusters with QoWS functionality

Step 1: *Start from main QoS principles*

– Classification of services
– Performance isolation
– High resource utilization
– Request admission
  - declaration of resource requirements
  - access control

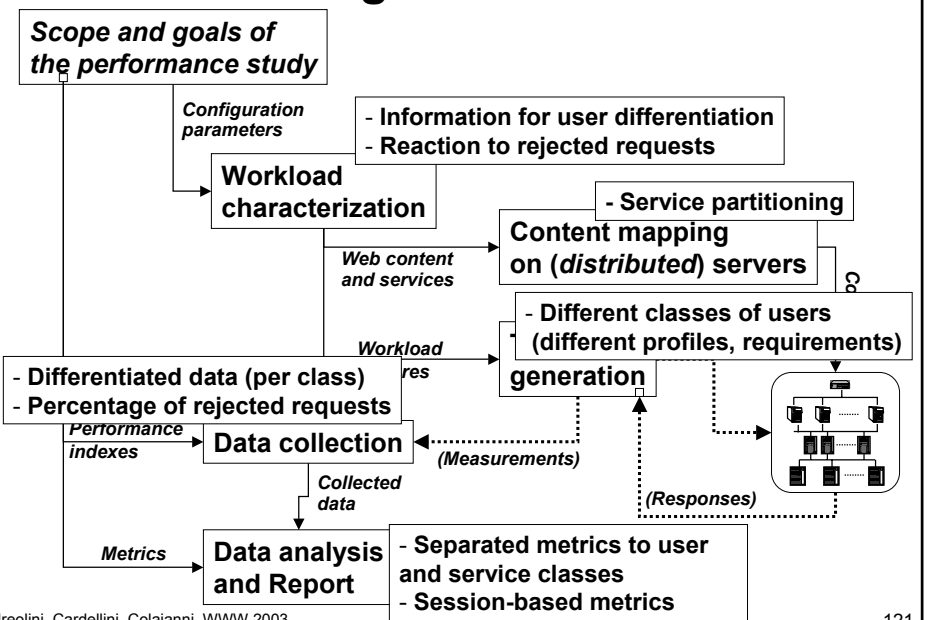


Step 2: *Find out the Web cluster components that can implement QoWS principles and mechanisms*

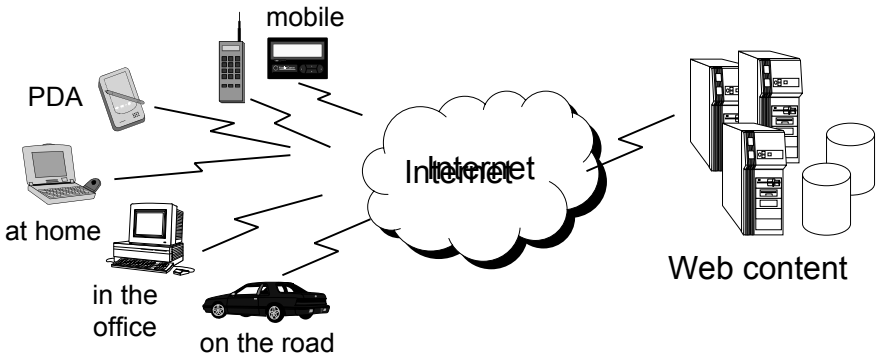
Measurement tools for QoWS

- **Lack of any benchmarking proposal**
- **Lack of load generator tools suitable for testing QoWS-enhanced systems**
  - The large majority of results about QoS and service differentiation have been carried out through simulation studies
- Main features that should be provided to test QoWS-enhanced systems

Benchmarking features for QoWS



(3) Universal Web access



- **Pervasive computing environment**
  - different client device constraints
    - connection bandwidth, processing power, storage, display, memory, and format handling capabilities

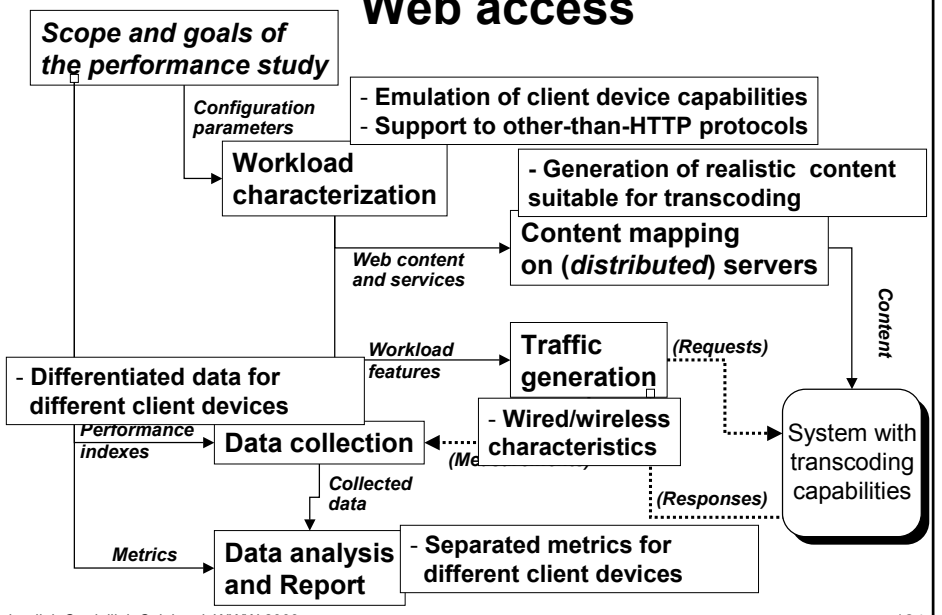
Web content adaptation

- **Techniques to adapt Web content to different device capabilities**
- **Transcoding: process of converting a multimedia object from one form to another form**
  - within media types (e.g., from JPEG to GIF format)
  - between media types (e.g., from image to text)
- **Transcoding may be deployed at:**
  - clients
  - content servers
  - intermediate proxies between the client and the content servers

No academic tool provides features suitable for testing Web systems enabled with transcoding capabilities

Just some load testing products with limited client device emulation (e.g., *Segue* for test of *IBM Transcoder Publisher*)

Benchmarking features for universal Web access



Thanks. Any question ?

weblab.ing.unimo.it

www.ce.uniroma2.it