Observability and Controllability for QOS Over Wide-Area Networks

Nagi Rao

(Nageswara S. V. Rao)

Computer Science and Mathematics Division

Oak Ridge National Laboratory

Oak Ridge, TN 37831

raons@ornl.gov

http://saturn.epm.ornl.gov/~nrao

QoS and Overlay Networks Session

DARPA Network Modeling and Simulation PI Meeting

April 2-4, 2001

San Diego, CA

What does it take to get end-to-end performance ?

Application ⟷ Network

Must use configuration and state of network

Need 1. Support from network: to observe and control

2. Implementation tools

Control Theory 101: to control a system, we need

1. Observabiity: ability to estimate model based on measurements
2. Controllability: ability to control the system trajectory

Very Roughly speaking, for Internet

Obsevarbility enables us to know the state of the network

ICMP, SNMP, BGP is a first step

Controllability provides ability to choose paths and traffic rates

- Current Internet provides

  Some observability and very little controllability

Observability:

Ability to infer system parameters and variables needed for QoS:

- Example: for end-to-end delay minimization: need to know bandwidths, delays, connectivity
- Two types of state variables
  - Configuration : connectivity in wired n/w
  - State: router delays, loss

Measurements are the key to estimating state – we simply cannot predict state

- we will not know the distributions precisely

- we will not have a complete differential equations for the Internet

Challenges and needs:

Approach to optimally and non-intrusively instrument the network

need right information with minimum cost

- traditional measurement systems mostly provide configuration data

- ICMP has limitations in presence of firewalls, ping disable, ICMP rate control, misleading traceroute responses

Systematic analysis and justification: cost minimization and canonicity of information must be explicit

Controllability:

Ability to control the system for QoS:

- Example: for end-to-end delay minimization: realize multiple paths
- Two types of control
  - Source control: TCP auto-tuning, parallel TCP streams, rate control
  - Remote control: routing paths, remote flow rates and priorities

Very little remote control is currently supported in Internet

- Set of users can collaborate to provide remote control to each other
- Several useful tasks can be performed with source level control, parallel TCP, NetLets, web100

Challenges and needs:

Approach to compute optimal paths and flows, and implement over the network

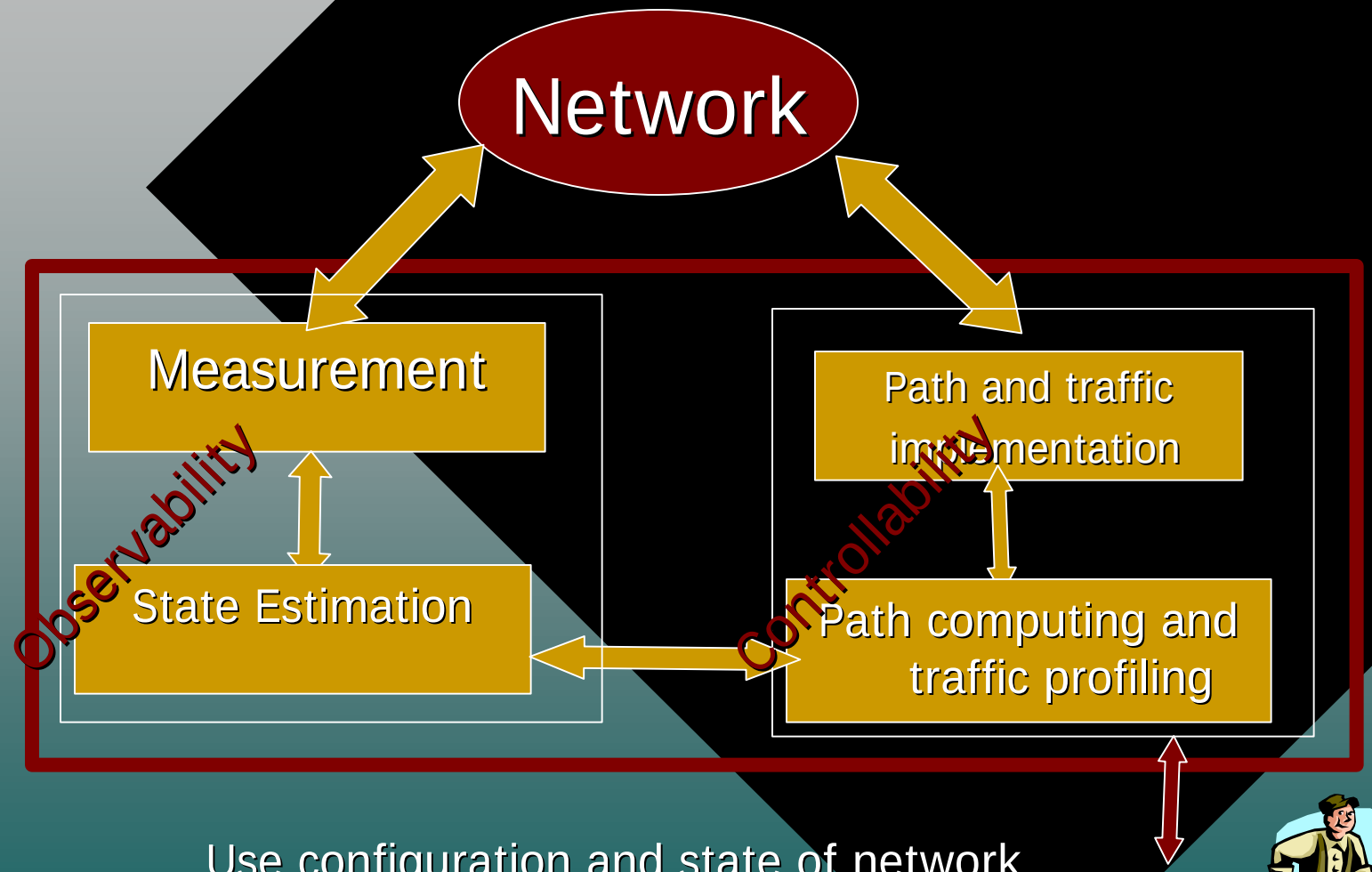
- Biggest challenge is the remote realization of computed routes and flows
- Collaborate with router companies to support control instrumentation
- Collaborate with ISP for support of control

Wide Spectrum of Network Control Support for QoS

Internet	Diffserv/IntServe, MPLS	Active Network
Very little	moderate	quite strong

Overlay Daemons:

Implemented on top of or inside OS and TCP/IP stack



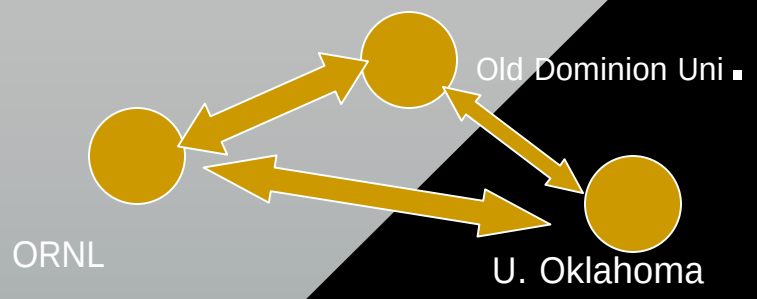
Use configuration and state of network

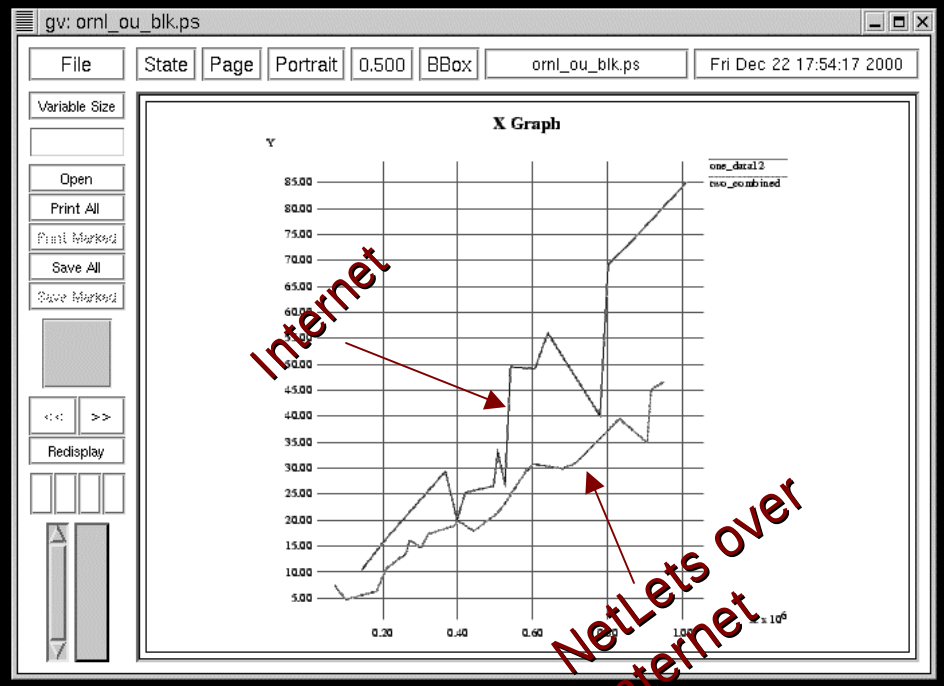
to provide the best performance from network

Why I think system like this is feasible ?

- I designed and implemented limited versions (theory, user interface, socket coding, etc.)
  - End-to-end delays over Internet using two-paths
    - Was able minimize using explicit multiple paths
    - Showed the analytical justification
    - Have first implementation on Internet – linux/unix
    - Applications: distributed and grid computing
  - Adhoc dynamic wireless network – No infrastructure needed
    - Automatically setup the network with IEEE 802.11 cards
    - Tracks connectivity changes uses other nodes as routers
    - Developed connectivity-through-time analysis
    - Working implementation – MS windows, linux point-of-access
    - Applications: remote robot team explorations
      formation of networks for emergencies

NetLets: Internet Measurements





Target: ORNL-OU: End-to-end delay minimization:

Solution: two-paths: ORNL-OU, ORNL-ODU_OU

NetLets: "optimize" end-to-end delay using multiple paths

observability	controllability
Delay measurements	Routing via other Netlets
Regression estimation	Quickest path computation

Observability:

Simple delay measurements and regressions are sufficient

Measurements must be actual delays not ICMP responses

Controllability:

Multiple paths are computed and used via other daemons

Can be much better with router support

Performance Guarantees: End-to-End delay

Θ_v Regression functions based on "Vapnik-Chevonenkis properties"

Given only measurements of sufficient (finite) size

Performance guarantee:

$$P\left\{\left[T(\hat{P}_R, R) - T(P_R^*, R)\right] > e\right\} < d$$

irrespective of the joint delay distributions

Informally, end-to-end delay of computed path is within specified tolerance of optimal with a specified probability

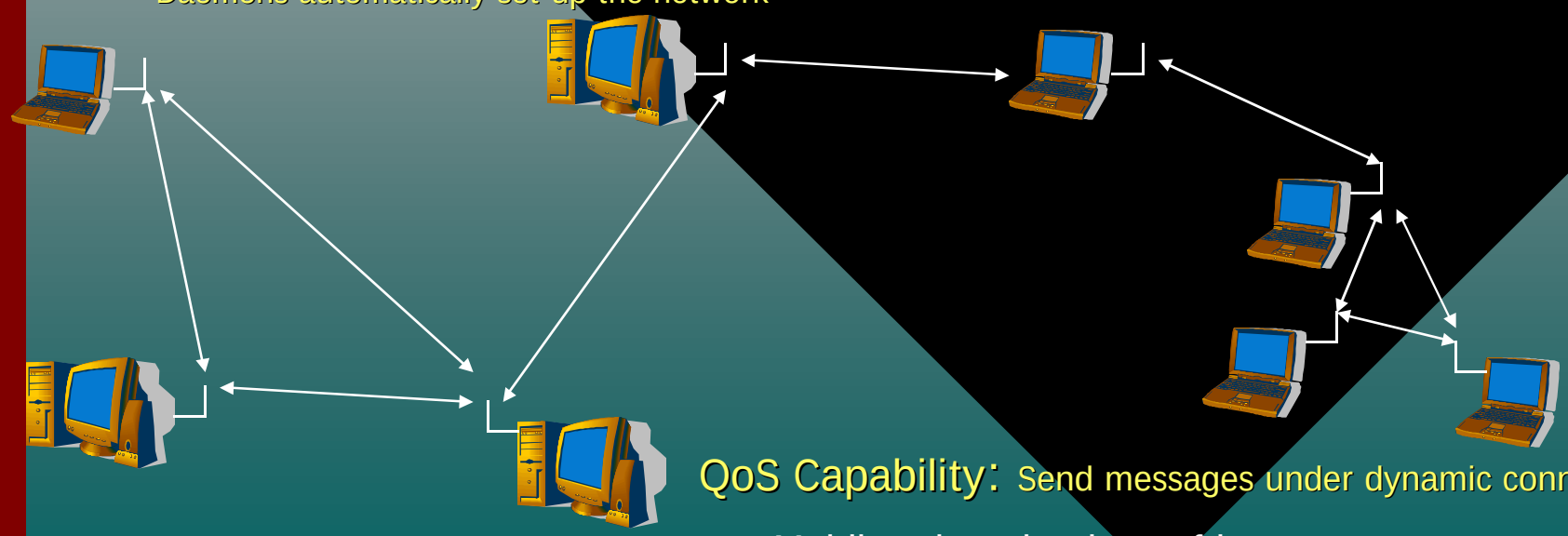
Analysis helped implementation:

1. Appropriate measurements and their optimization
2. Performance savings are real

Existing Capability: Adhoc Networks

- Uses only 802.11 pc-cards of Turbowave, Inc (Sponsor)
- Nothing else is needed – no access points – no infrastructure
- Can exchange message between any two computers using others as routers –node can be laptops,desktops running win95/98/ME, NT/2000, unix/linux
- Can adapt to laptop movements
  - can track connectivity changes
  - Implements connectivity-through-time
- Daemons automatically set-up the network

observability	controllability
Hello/iamhere UDP messages	Routing buffering
Multihop Connectivity	Route Computation



QoS Capability: Send messages under dynamic connectivity

Mobile robots in place of laptops