Ubiquitous Sentient Network: Measurement-Based Network Operating System

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

Large Scale Networking Workshop

March 12-14, 2001

Vienna, VA

What does it take to integrate a node into Internet ?

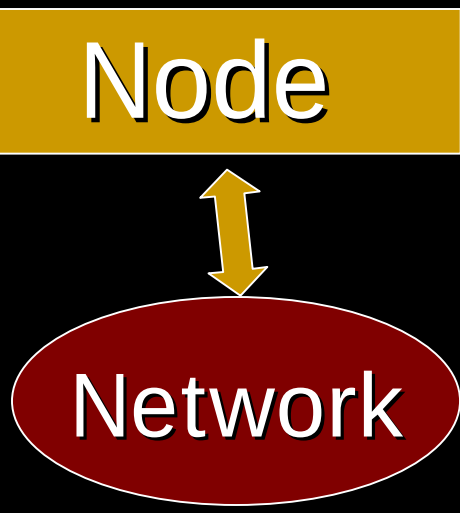
Depends on OS, end device, network support

- no uniform way
- at the mercy of ISPs

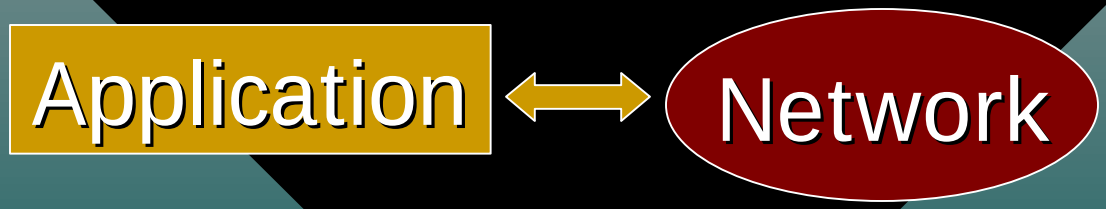
End nodes of network must be plug and play

- can't do it gracefully at present

(e.g. palm with wireless cards)



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

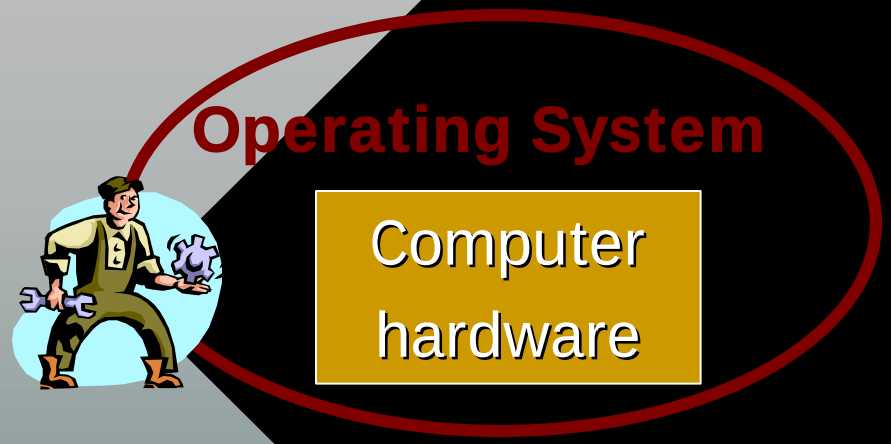


Must use configuration and state of network

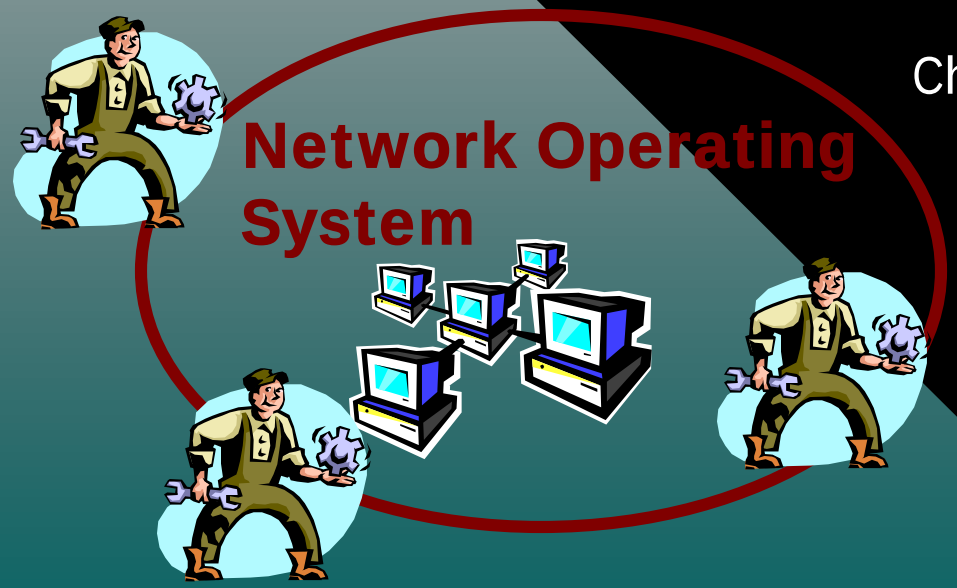
Need 1. Support from network

2. Tools for implementing the functionality

Operating system for network



Conventional OS facilitates access to computing h/w



Network Operating System provides easy access to underlying network:

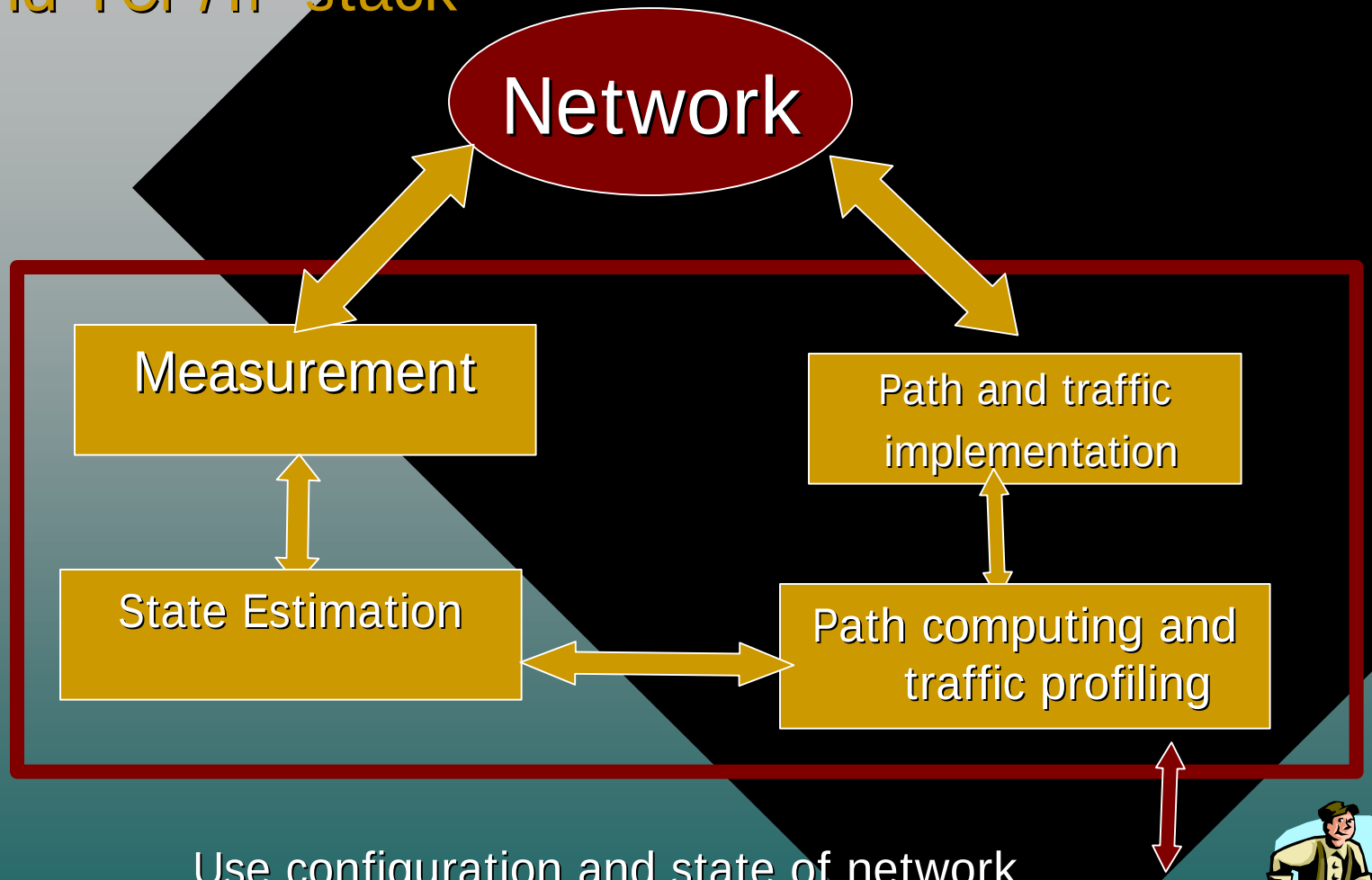
Challenges:

- distributed hardware
- network model partially known

Goal: Make network middleware as “instructions”

NOS Daemons:

Implement NOS as daemons on top of OS and TCP/IP stack



Use configuration and state of network

to provide the best performance from network

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

Components of NOS Daemons

Two basic types of nodes:

- End-nodes: laptops, palmtops, workstations
  - Complex but seem to be implementable in theory
- Internal nodes: routers, switches, hubs
  - Complex - technically and politically challenging

Daemons run on all network nodes

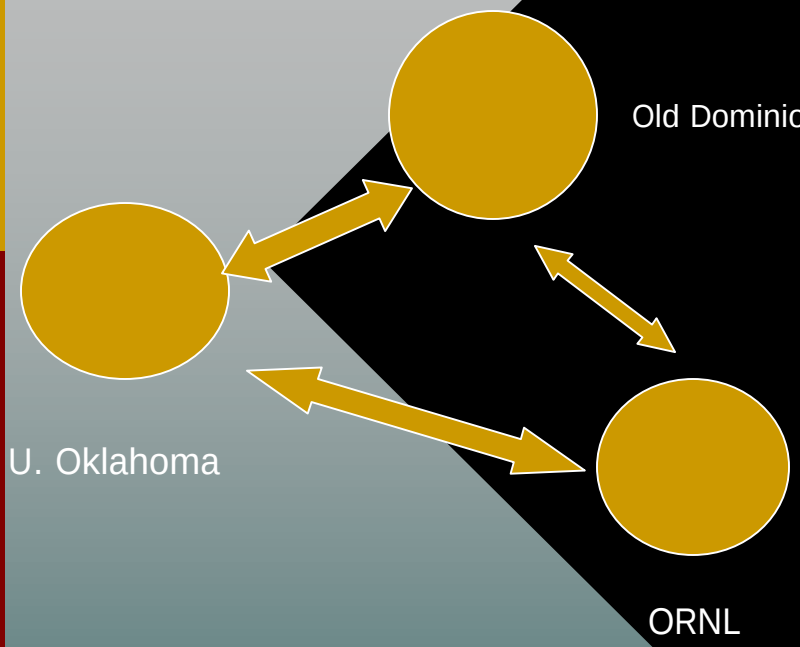
NOS Kernel: (more details in whitepaper and website)

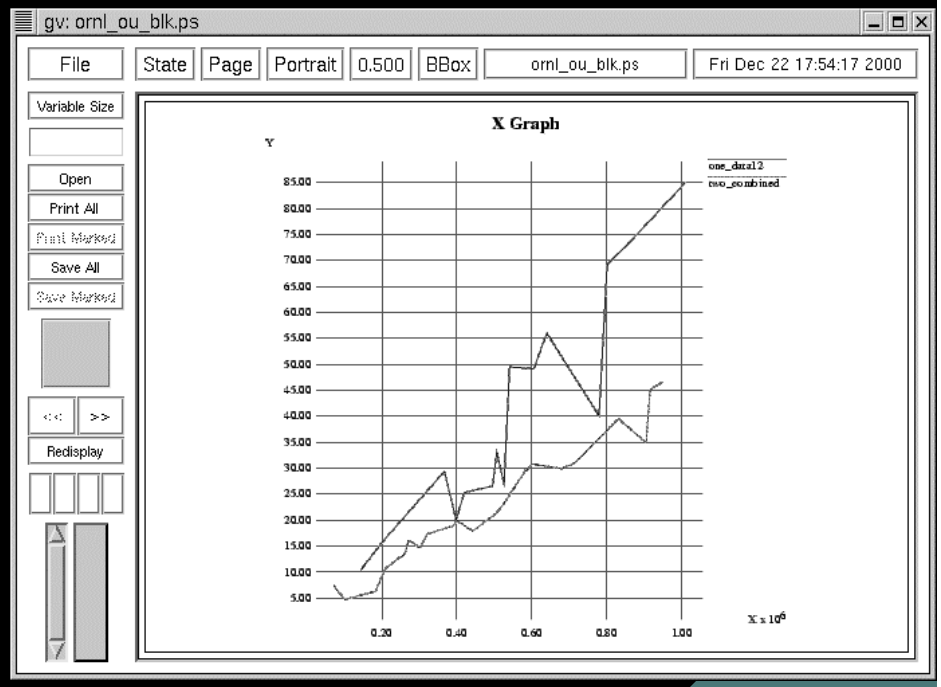
- Network Modeling and Optimization sub-kernel
  - Perform the network task best possible way
- Sub-kernel for end devices
  - Enables plug-and-play of nodes

Why I think system like this is feasible ?

- I designed and implemented limited versions of NOS (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

Internet Measurements





DARPA Funded

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

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

NOS automatically “optimizes” end-to-end delay using multiple paths

Performance Guarantees: End-to-End delay

Θ_v Regression functions based on "Vapnik-Chevonenkis properties"

Given only measurements of sufficient (finite) size

Performance guarantee:

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

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:

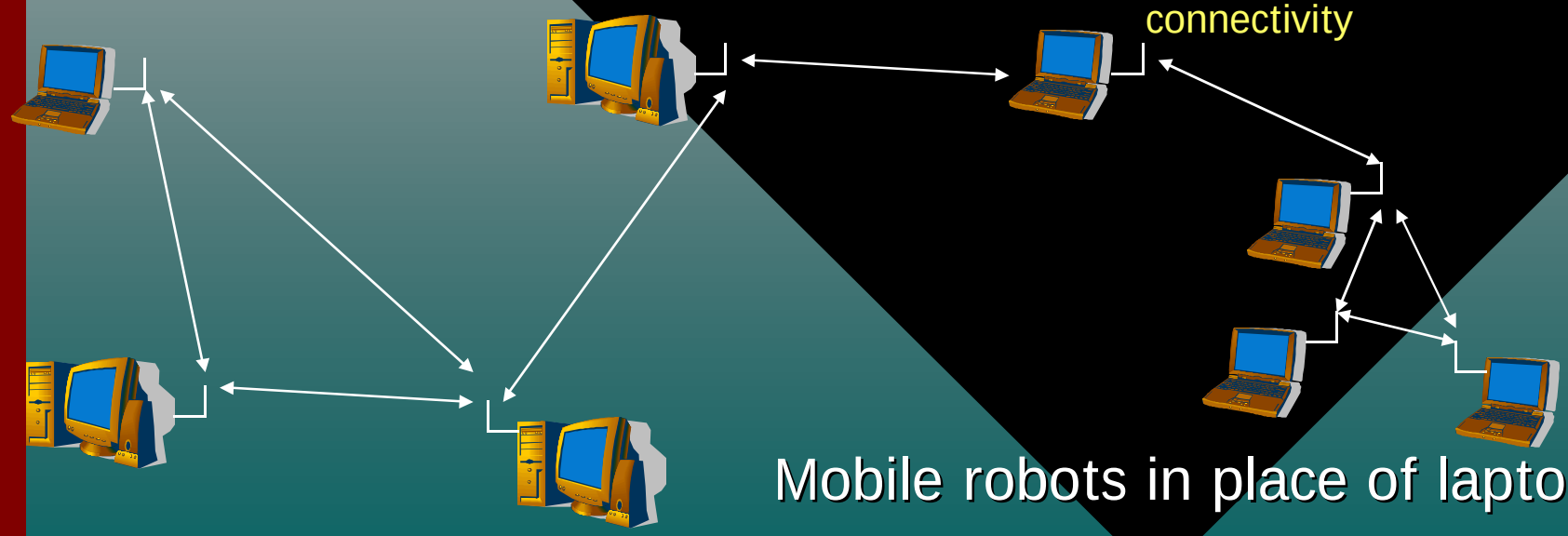
1. Appropriate measurements and their optimization
2. Performance savings are real

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

NOS capability:

Send messages under dynamic connectivity



Mobile robots in place of laptops

Details:

White paper

raons@ornl.gov

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