Software Systems Research – Portfolio Review

Dr. Nelson R. Manohar Alers

nelsonmanohar@yahoo.com
nelsonmanohar.sphosting.com/research

Outline of the Talk

- Background [10%]
- Computer-Supported Collaboration [25%]
- Dynamically Customized Web Touring [25%]
- Multimedia Computing Networking [35%]
- Wrap-Up [5%]

Educational Background

- **B.S. in Computer Engineering**
  - **The University of Puerto Rico at Mayaguez**
- **M.S. in Computer Engineering**
  - **The University of Wisconsin at Madison**
  - Computer Architecture and Organization (Advisor Dr. Yu Hen Hu)
- **M.S.E. in Industrial Engineering**
  - **The University of Michigan at Ann Arbor**
  - Statistical Quality Control/Information Systems (Advisor Dr. Dan Teichroew)
  - From January 1991 to December 1992*
- **Ph.D. in Computer Science and Engineering**
  - **The University of Michigan at Ann Arbor**
  - Software Systems Research (Advisor: Dr. Atul Prakash)
  - From January 1991* to May 1997

Work Experience

- **Senior Technical Associate**
  - **@ AT&T Bell Laboratories, Naperville, Illinois**
  - From June 1986 to August 1986
- **Member of the Technical Staff I**
  - **@ AT&T Bell Laboratories, Naperville (Indian Hill), Illinois**
  - From June 1987 to March 1991*
- **Member of the Technical Staff I**
  - **@ AT&T Bell Labs field work at La Telefonica's Spain AIN**
  - From September 1990* to December 1990*
- **Member of the Technical Staff**
  - **@ Bell Communications Research, Piscataway, New Jersey**
  - From June 1992 to August 1992
- **Research Staff Member**
  - **@ IBM Thomas J. Watson Research Center**
  - From May 1997 to December 2001*

Research Traversal

- Computer Architecture
  - University of Wisconsin - Madison (87-88)
- Advanced Intelligent Networks
  - AT&T Bell Labs (S'86, S'87, 88-91)
- Statistical Process Control & Systems Engineering
  - University of Michigan IOE Department (91-92)
- Distributed Computing & Distributed Systems
  - Bellcore (S'92);
  - University of Michigan EECS Department (92-93)
- Collaborative Systems
  - University of Michigan EECS Department (93-94)
- Collaborative Multimedia Systems
  - University of Michigan EECS Department (94-97)
- Multimedia Computing Networking
  - IBM T. J. Watson Research Center (97-01)

Selected Publications

- **"Applying Statistical Process Control to the Adaptive Rate Control Problem",**
  - Manohar, Nelson R.; Willebeek-Lemair, Marc H.; Prakash, Atul, in Proceedings of Multimedia Computing and Networking Conference, pp. 45-60, San Jose, CA, January 1998.
- **"Dealing with Synchronization and Timing Variability in the Playback of Interactive Session Recordings",**
  - Nelson R. Manohar and Atul Prakash, in Proceedings of the Third ACM Int'l Multimedia Conference, pp. 45-56. San Francisco, CA, November 1995.
- **"The Session Capture and Replay Paradigm for Asynchronous Collaboration",**
  - Nelson R. Manohar and Atul Prakash, in Proceedings of the Fourth ECSCW Conference, pp. 149-164. Stockholm, Sweden, September 1995.
- **"A Framework for Programmable Multimedia Overlay Networks",**
  - N. R. Manohar, A. Mehra, M. H. Willebeek-LeMair and M. Naghshineh, in IBM Journal of Research and Development, Special Issue on Digital Video, 43(4), July/August 1999.
- **"A Flexible Architecture for Heterogeneous Replayable Workspaces",**
  - Nelson R. Manohar and Atul Prakash, in Proceedings of the Third IEEE Int'l Conference on Multimedia Computing and Systems, pp. 274-278, Hiroshima,

Other Publications

- **"Streaming and Synchronization of Re-executable Content",**
  - N. Manohar and A. Prakash, *unpublished*, 1998.
- **"Design Issues on the Support of Tools and Media on Replayable Workspaces",**
  - N. Manohar and A. Prakash, CSE-TR-304-96, Dept. of EECS, Univ. of Michigan, September 1996.
- **"Design Considerations in Building a Distributed Collaboratory",**
  - A. Prakash, F. Jahanian, R. Hall, N. Manohar, A. Mathur, C. Rasmussen, H. Shim, T. Weymouth, G. Wu, D. Atkins, R. Clauer, and G. Olson, School of Information, Univ. of Michigan, Feb. 1995.
- **"Statistical Quality Control and Software Productivity."**
  - N. Manohar, Quals Report, (research work under Dr. Daniel Teichroew), Dept. of IOE, Univ. of Michigan, May 1992.
- **"The DCIS6 Finite State Machine Tables",**
  - Nelson R. Manohar-Alers, AT&T Bell Laboratories, Int'l 5ESS Features Development Department, Internal Memorandum, August 1989.
- **"The Computer Architecture of VLSI Digital Signal Processors",**
  - MSEE Thesis/Report, Nelson R. Manohar-Alers, Department of ECE, Graduate Engineering Library, University of Wisconsin-Madison, August 1988.

Intellectual Property (IP) Activity

- **IP Training:**
  - Trained with **IBM** **Master Inventors** Mr. Leon Lumelsky and Dr. Philip S. Yu

- **IP Performance:**
  - **Principal inventor** (and principal inventor-in-training) on seven patents.
  - **Eight USPTO patent filings**, **seven** successfully granted.
  - **Two IBM Invention Plateaus** achieved.

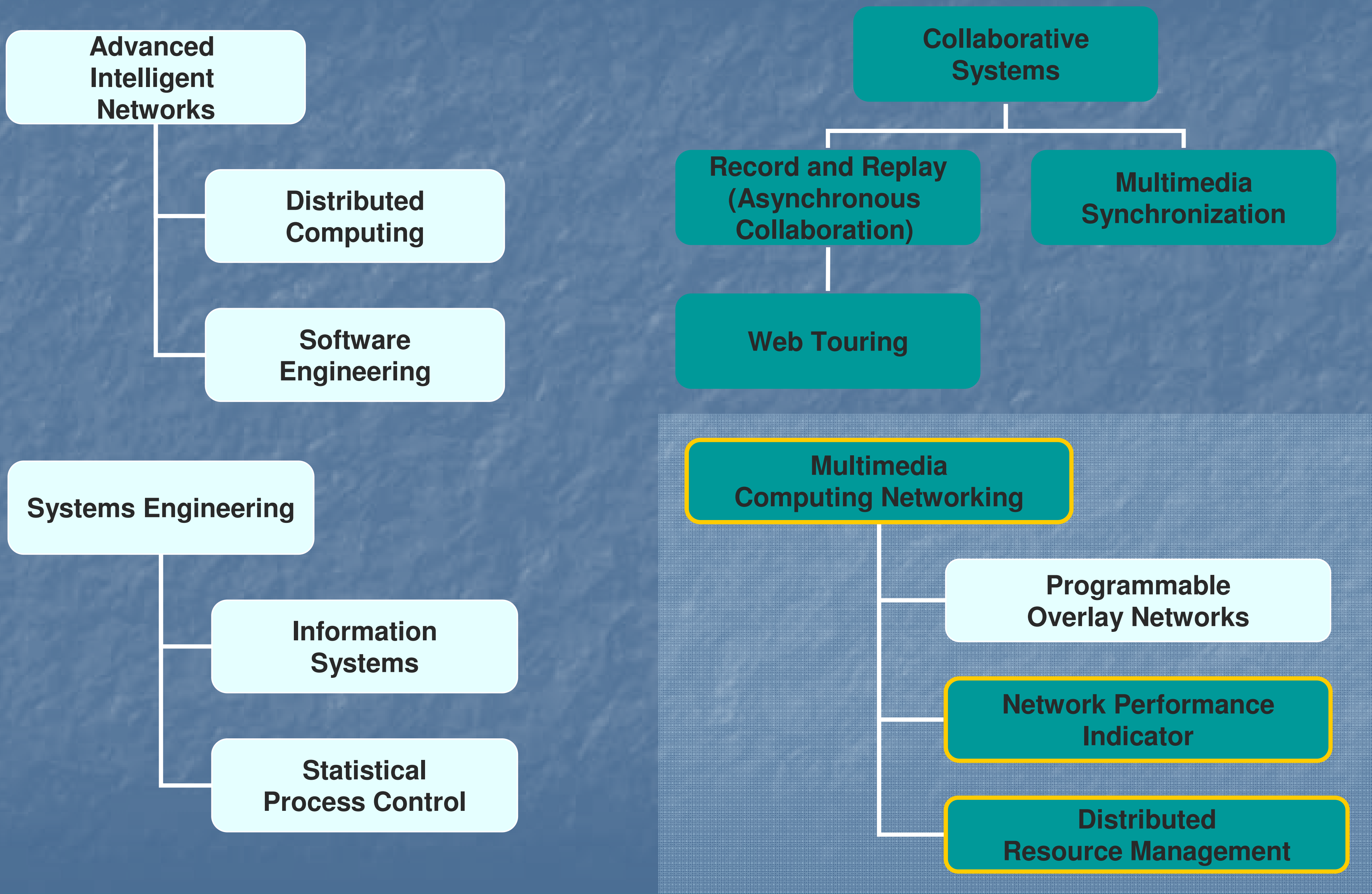
Selected Patents

- **6,572,662: dynamic customized web tours…**
  - related to: tour data mining, tour authoring, <u>like-minded</u> touring of multiple websites, token-based control of traversal projections over web-tours, touring clients,...
- **6,516,350: self-regulated resource management…**
  - related to: autonomous (self-regulated) distributed resource management integrating traditional <u>demand-shaping and capacity-shaping</u> mechanisms …
- **6,466,980: capacity shaping of distributed resources on an internet environment…**
  - related to: replication management, QoS, and <u>capacity-shaping</u> of a network's resources (e.g., capacity-follows-demand management distributed resource management policy), etc…
- **6,529,950: policy-based QoS negotiation…**
  - related to: brokering framework for distributed resource management, etc...

Complementary Patents

- **6,463,454: integrated load distribution and resource management …**
  - related to: replication and capacity policies for distributed resource management, etc…
- **6,460,082: service-oriented resource signatures…**
  - related to: low overhead resource management and measurements policy for distributed servers, resources, capacity, objects, etc...
- **6,377,996: seamless live streaming handoffs …**
  - related to: handoff of live multimedia streaming across servers, “virtual sockets”, migration transparency, etc...

Software Systems Research Map



Outline of the Talk – Revisited

- **Background [10%]**
- **Computer-Supported Collaboration [25%]**
- **Dynamically Customized Web Touring [25%]**
- **Multimedia Computing Networking [35%]**
  - **Building Robust Network Performance Indicator**
  - Distributed Resource Management for Multimedia
- **Wrap-Up [5%]**

Building a Robust Network Performance Indicator

"Applying Statistical Process Control to the Adaptive Rate Control Problem"

Work at IBM Thomas J. Watson Research Center

"Applying Statistical Process Control to the Adaptive Rate Control Problem", by Manohar, Nelson R.; Willebeek-Lemair, Marc H.; Prakash, Atul, in Proceedings of Multimedia Computing and Networking Conference, pp. 45-60, San Jose, CA, January 1998.

Motivation

- Multimedia Computing Networking
  - **handling the impact of multimedia over the network**
    - e.g., session-oriented, multimedia flows, end-to-end QoS reqs., resource management, value-asset model, etc.
  - **handling of the "impedance mismatch" that exists**
    - between multimedia applications
    - and (to-be provisioning, present or future) networks
  - **we would like the resulting mechanisms or building blocks**
    - to be robust
    - easy to implement
    - low (signaling and tracking) overheads

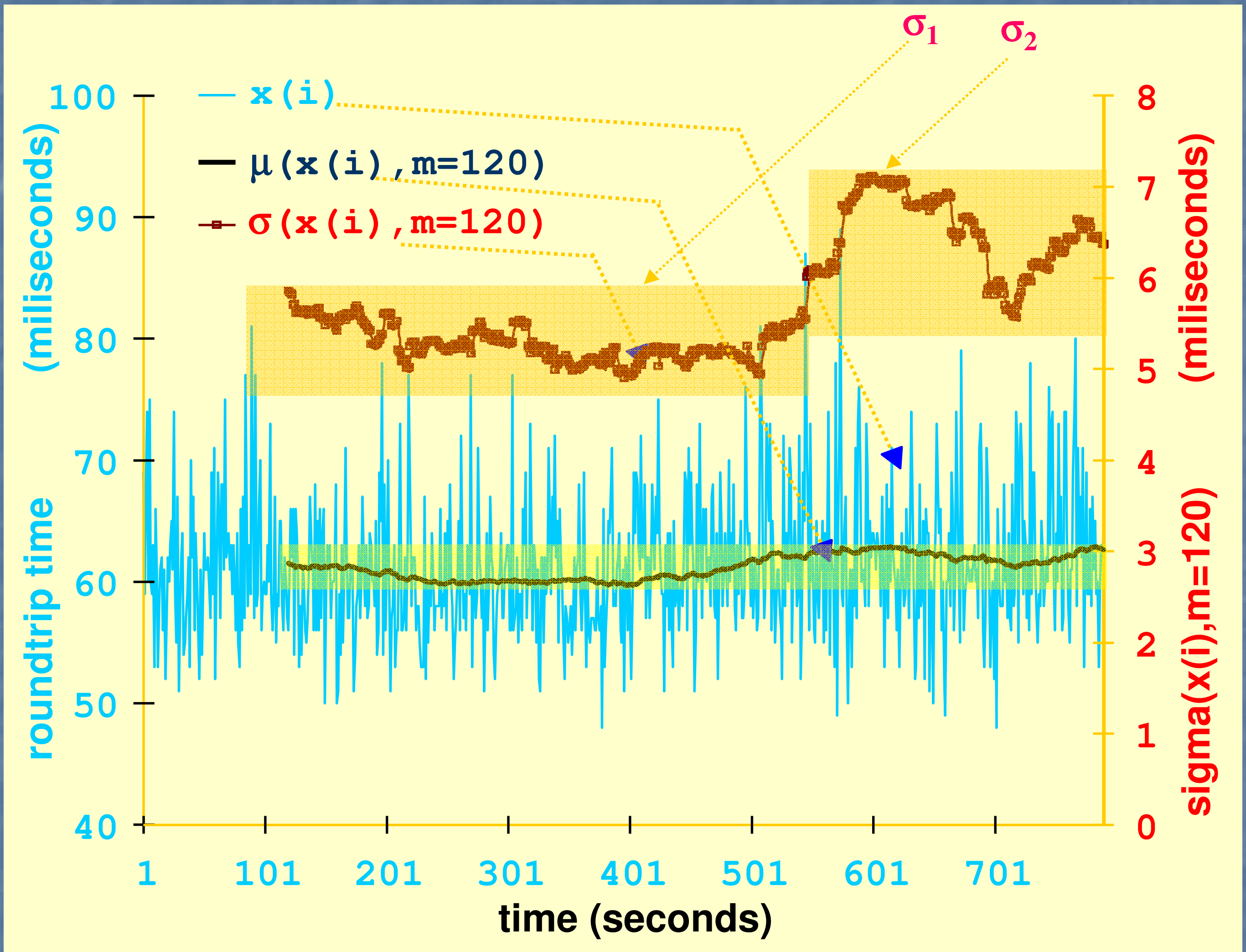
Related Work

- **mechanisms to handle "impedance mismatch"**
  - **(traditionally) from applications to the network:**
    - inducing multimedia application reqs. into networking middleware
      - for example, diffserv (TCP), multicast (routing), etc.
  - **(but also) from network to multimedia infrastructure:**
    - enhancing network capacity (internet2, vBNS, etc.)
    - **enhancing the intelligence of multimedia infrastructure to adapt to the network state (RSVP, QoS, etc.)**
- **network state measurements**
  - **(active) network probing**
    - probe-and-adapt: short-term fluctuations, unnecessary adjustments
  - **(passive) network and web traffic characterization**
    - (web) spatial/temporal stability: different time-scale components
    - ethernet/web traffic fractal: similar shape regardless of timescale

Network Performance Envelope:

Factoring Process Variability on Sessions

- **From:**
  - probe-and-adapt
    - fast indicators
    - very low setup cost
  - no confidence
- **To:**
  - process state
    - forecast-strength
    - some setup cost
  - confidence analysis

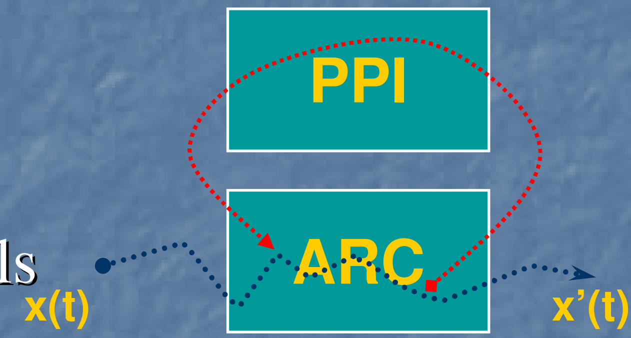


RTT network probe: one apparently stable process mean, but upon examination variability states (i.e., a process shifts).

Building Reliable Building Block:

Network Performance Indicator

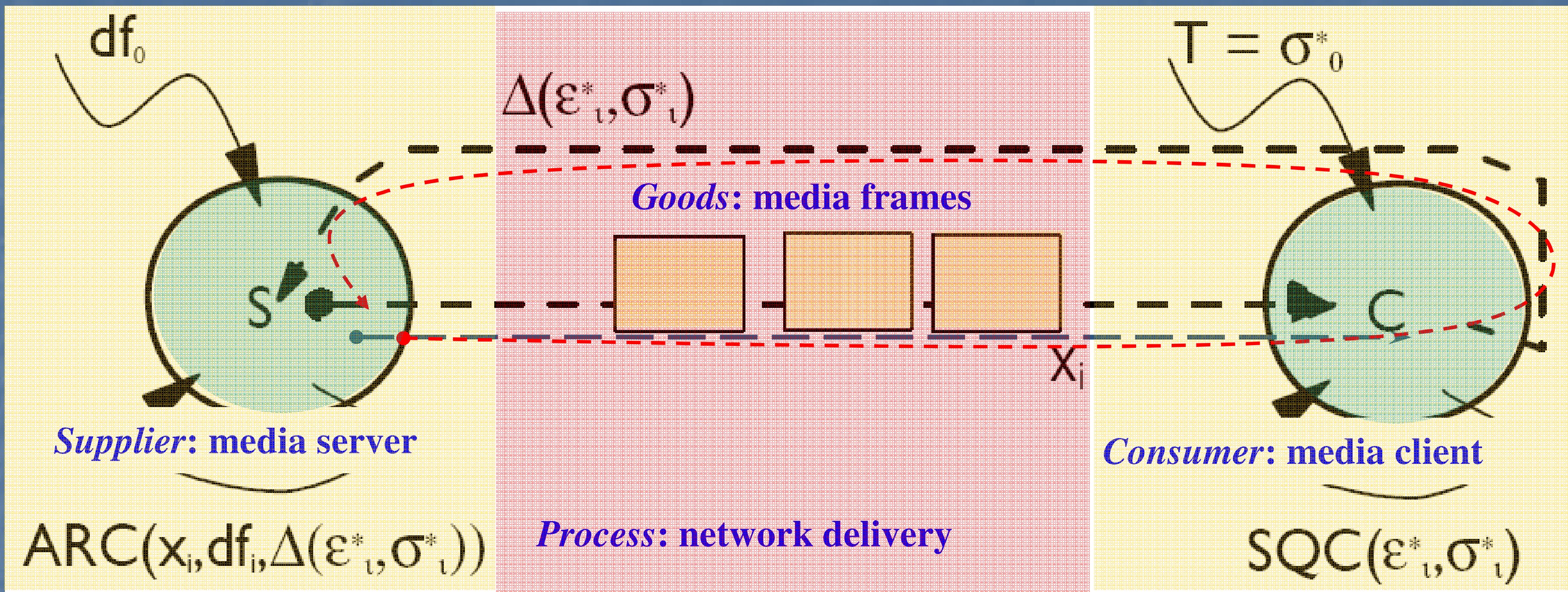
- **Reliable process performance indicators (PPI)?**
  - **robust forecasts**
    - strength of sampling and smoothing
    - capable of associating confidence to forecasts
  - **assimilation of process variability**
    - to recognize significant changes
  - **generation/tracking of process state**
    - to understand where we stand
- **Important to adaptive applications**
  - atop (i.e., guiding) prove-and-adapt protocols
  - such as Adaptive Rate Control (ARC)



- **Statistical Process Control (SPC) as PPI-kernel**
  - typically long-term horizon, industrial processes
    - run-to-run feedback control vs. online-SPC
    - centralized online-SPC vs. distributed online-SPC
  - adapt online-SPC for distributed ARC problem

Network Performance Envelope:

Statistical Process Control Formulation



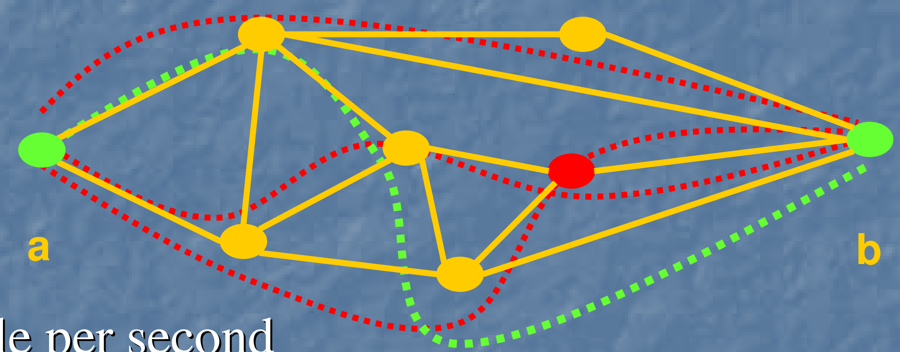
- **Supplier: adaptive rate control**
  - to match production to delivery
- **Process: network delivery**
  - *goods*: media packets, frames, buffers
- **Consumer: smoothing problem**
  - adapts delivery to presentation

- **SPC: end-to-end process performance**
  - quantizes performance measurements
  - into statistical process state – a quality indicator
  - then drives ARC() w such quality indicato

SPC
ARC

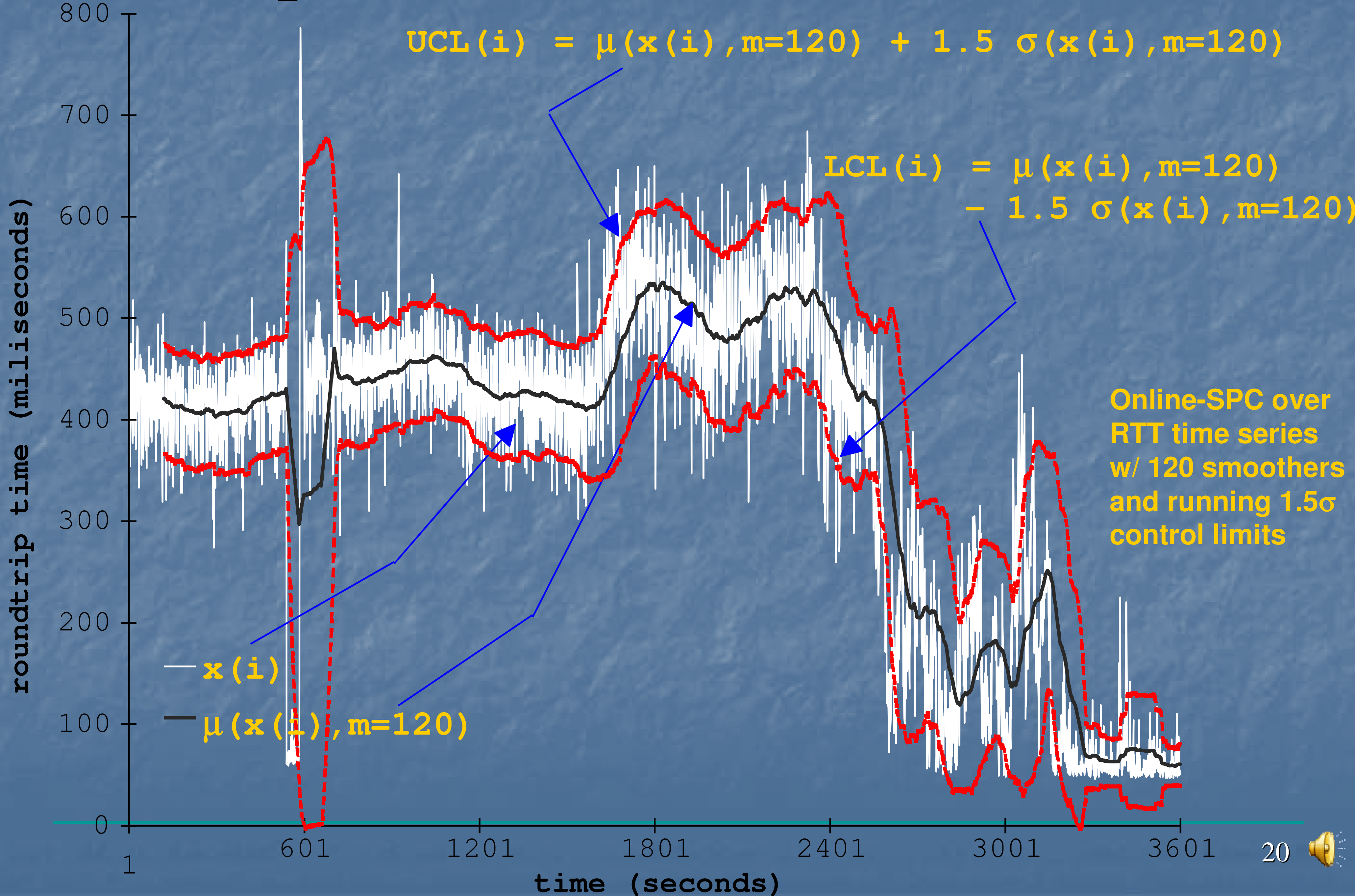
Network Measurements – Setup

- (active) network probing and monitoring
  - **network indicator**: RTT roundtrip delay (measure of congestion along path)
  - e.g., available bandwidth, **network congestion, stability, jitter**
- probing methodology (time series acquisition)
  - **end nodes**: (Univ. Michigan at A2) - (Univ. North Carolina – CH)
  - **network probe**: ping (ICMP)
  - **test duration**: 3600 seconds
  - **timescale**: seconds to minutes
  - **smoother**: 120/30 UWMA smoother
  - **sampling frequency**: 1 random sample per second
  - **sampling load**: negligible (40ms to 400ms) with respect to sampling frequency (1 second) with respect to available network bandwidth
  - **sampled distribution**: no constraining assumption due to law of large numbers, random sampling, temporal/spatial stability, and network fractality



- bottleneck spots, router instability, underlying distr.
  - timescale leads to aggregation effect on multiple path routing and corresponding sampling of underlying distribution(s) on each *i-th* sample

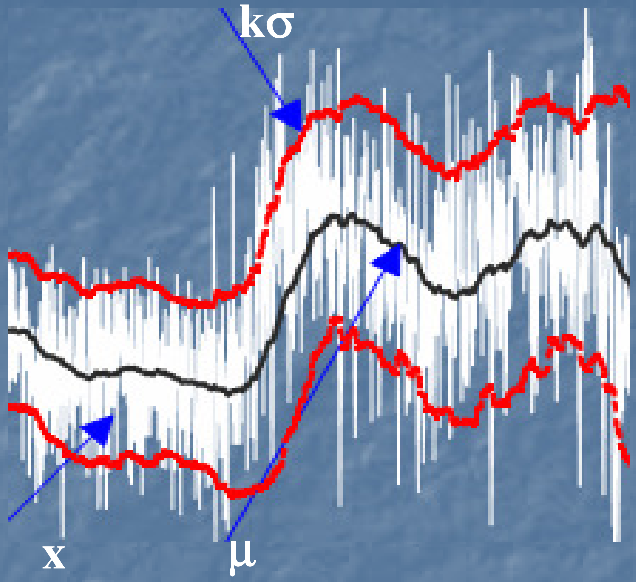
Online-SPC Floating Performance Envelope



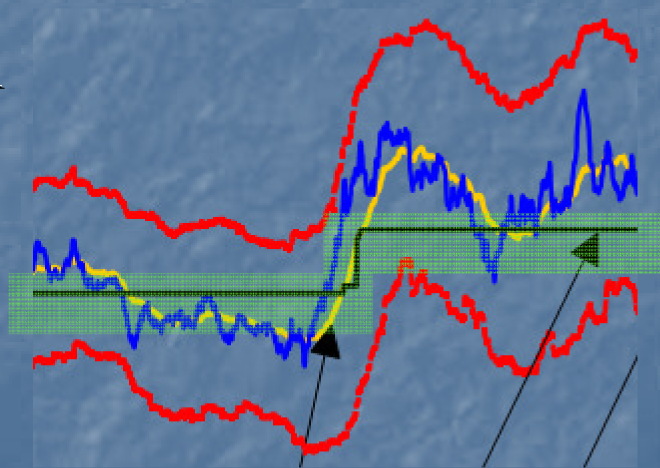
Network Performance Envelope:

Process State – Basic Idea

- **floating performance envelope**
  - tracking window over process state (μ, σ)
  - under associated statistical confidence region (kσ)
  - reacts to statistically significant changes
- **process stability indicator**
  - quantizes performance into statistical process state
- **providing context for state changes**
  - between **fast** (fractional process state) signal
  - *wrt* **slow** (full process state) signal
- **compare fast against stable signal**
  - approx. same versus significantly different
  - two types of **piece-wise linear** segments
    - **horizontal** segments (process state)
    - linear **slopes** (process changes)
  - building block for adaptive infrastructures



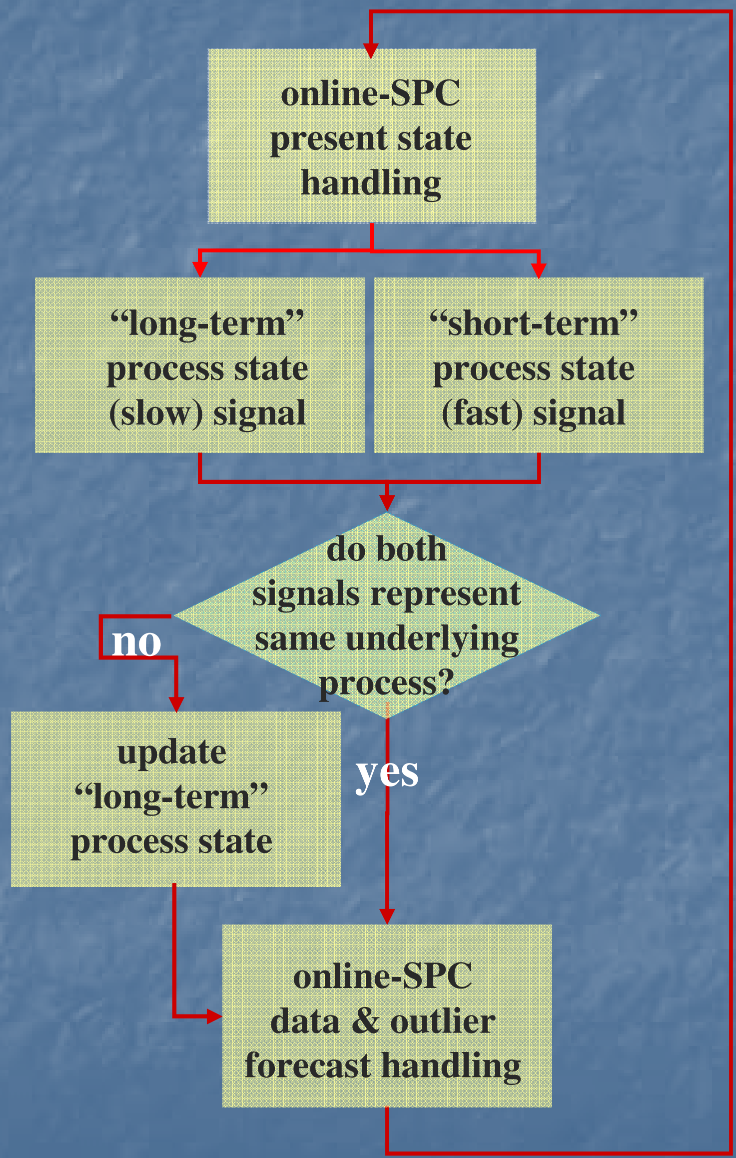
that is, just with online-SPC, but now three signals!



Network Performance Envelope:

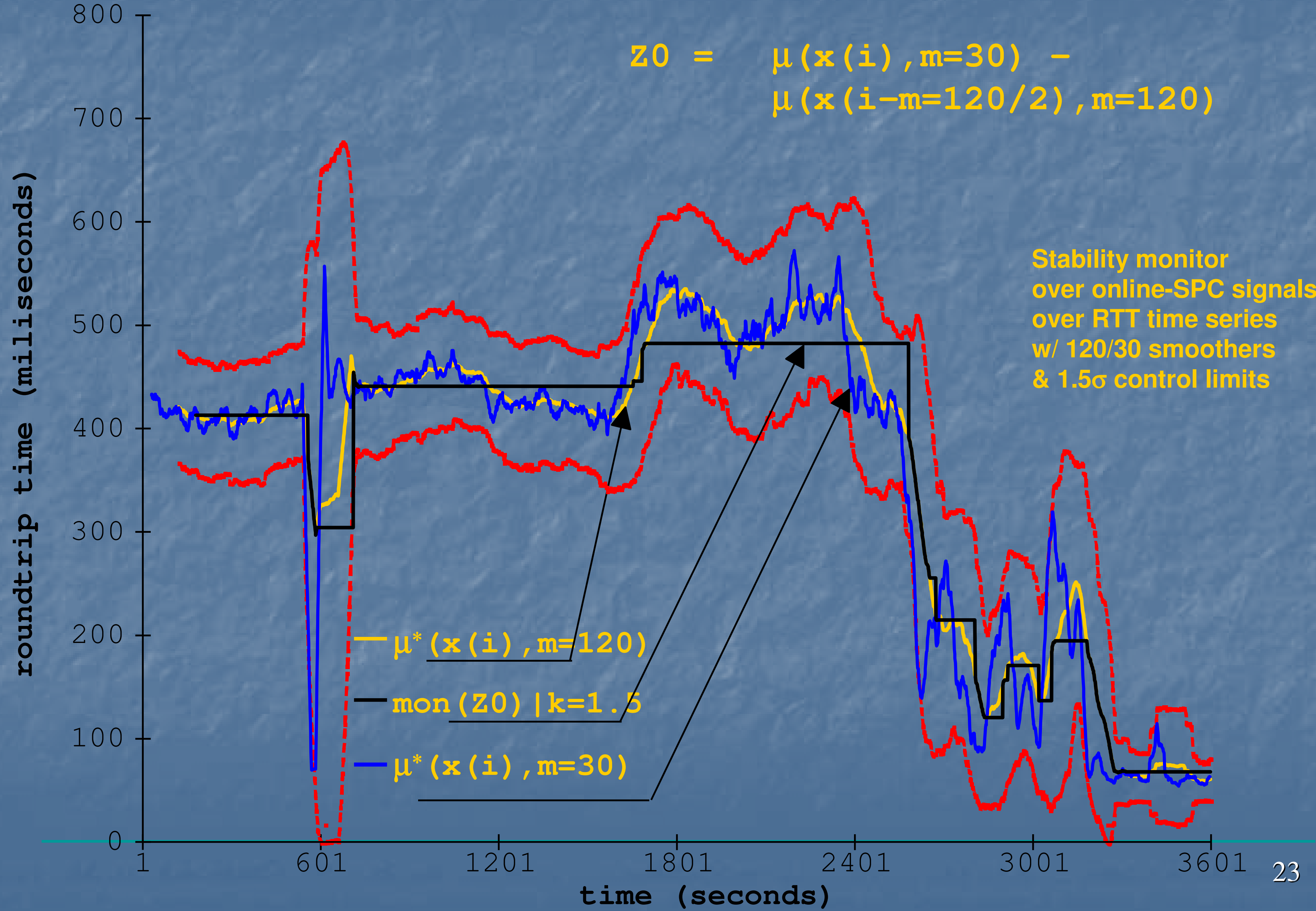
Online-SPC Process State Tracking Kernel

- **Running Window Indicators**
  - UWMA(x_i,m) smoothers
  - $\mu(i,m') = \mu(x_i \,..\, x_{m'-i})$
  - $\mu(i,m) = \mu(x_i \,..\, x_{m-i})$
  - $\sigma(i,m) = \sigma(x_i \,..\, x_{m-i})$
- **Hypothesis Testing**
  - $H0 : \mu(i,m') = \mu(i,m)$
  - $Z0 = \mu(i,m') - \mu(i - {}^{m}/_{2}, m)$
- **Process State Generation**
  - if $|Z0| < k * \sigma(i,m)$
  - then $mon_i = mon_{i-1}$
  - else $mon_i = \mu(i,m')$
  - $mon^*_{i+1} = mon_i$
  - $error = mon^*_i - mon_i$



Network Performance Envelope:

Network Stability Monitor



Network Performance Envelope:

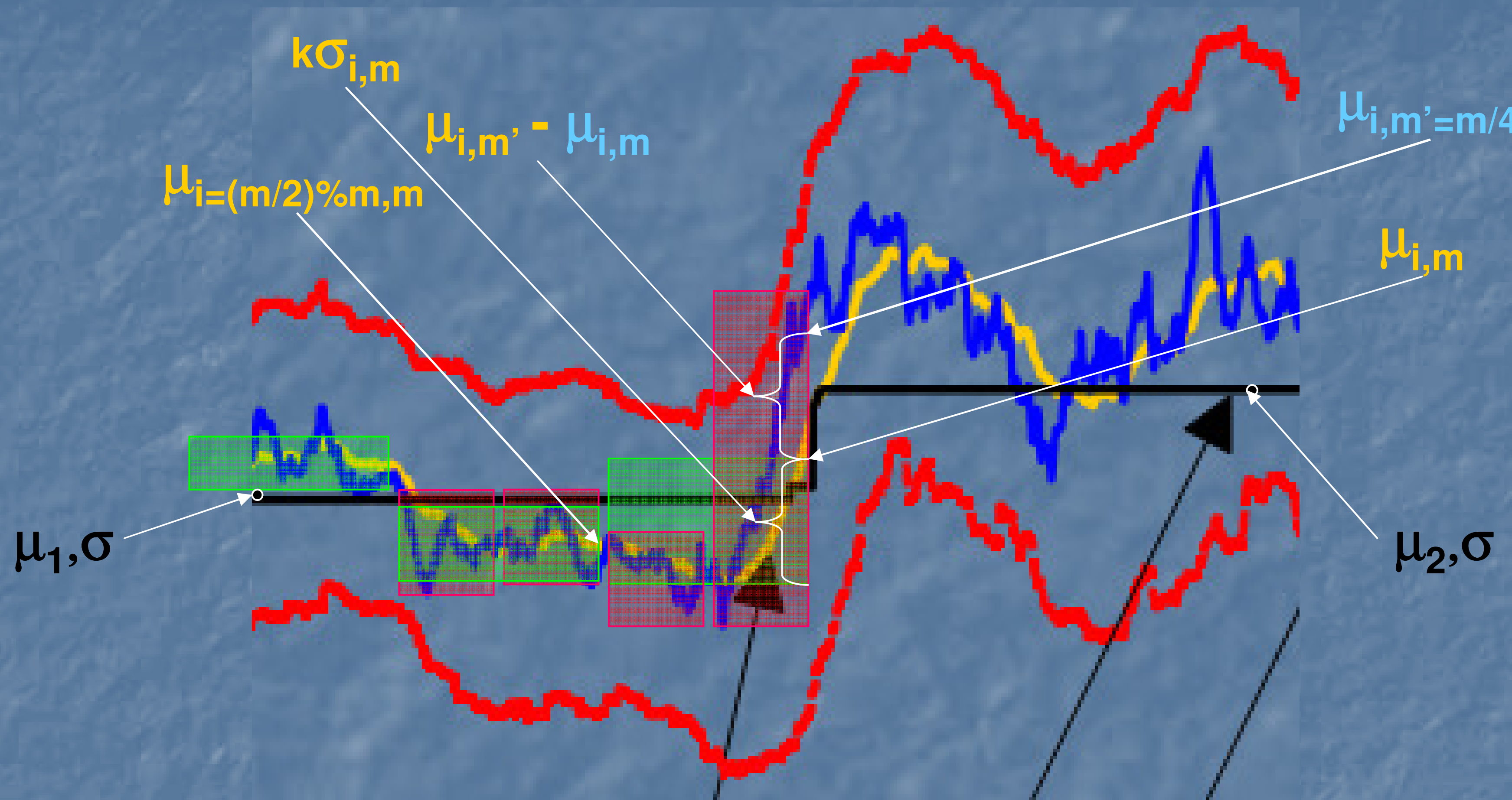
Stationarity Test: Comparison of Sampled Means

$$\underbrace{Z0 = \left\{ \begin{array}{c} \mu(i, m' = 30) \\ \text{fast signal (front - end)} \end{array} \right\} - \left\{ \begin{array}{c} \mu(i - \frac{m}{2}, m = 120) \\ \text{slow signal (\sim uncorrelated past)} \end{array} \right\}}_{\text{if } (|Z0| < k\,\sigma(i, m=120))}$$

- **Hypothesis tested by comparing two indicators**
  - (a) slow signal and
  - (b) fast signal
- **This resulting in two process states**
  - (a) stationary state or
  - (b) process change
- **Sensitization parameters**
  - m’ (fast signal), m (slow signal)
  - k (confidence), m/c (decorrelator)

Network Performance Envelope:

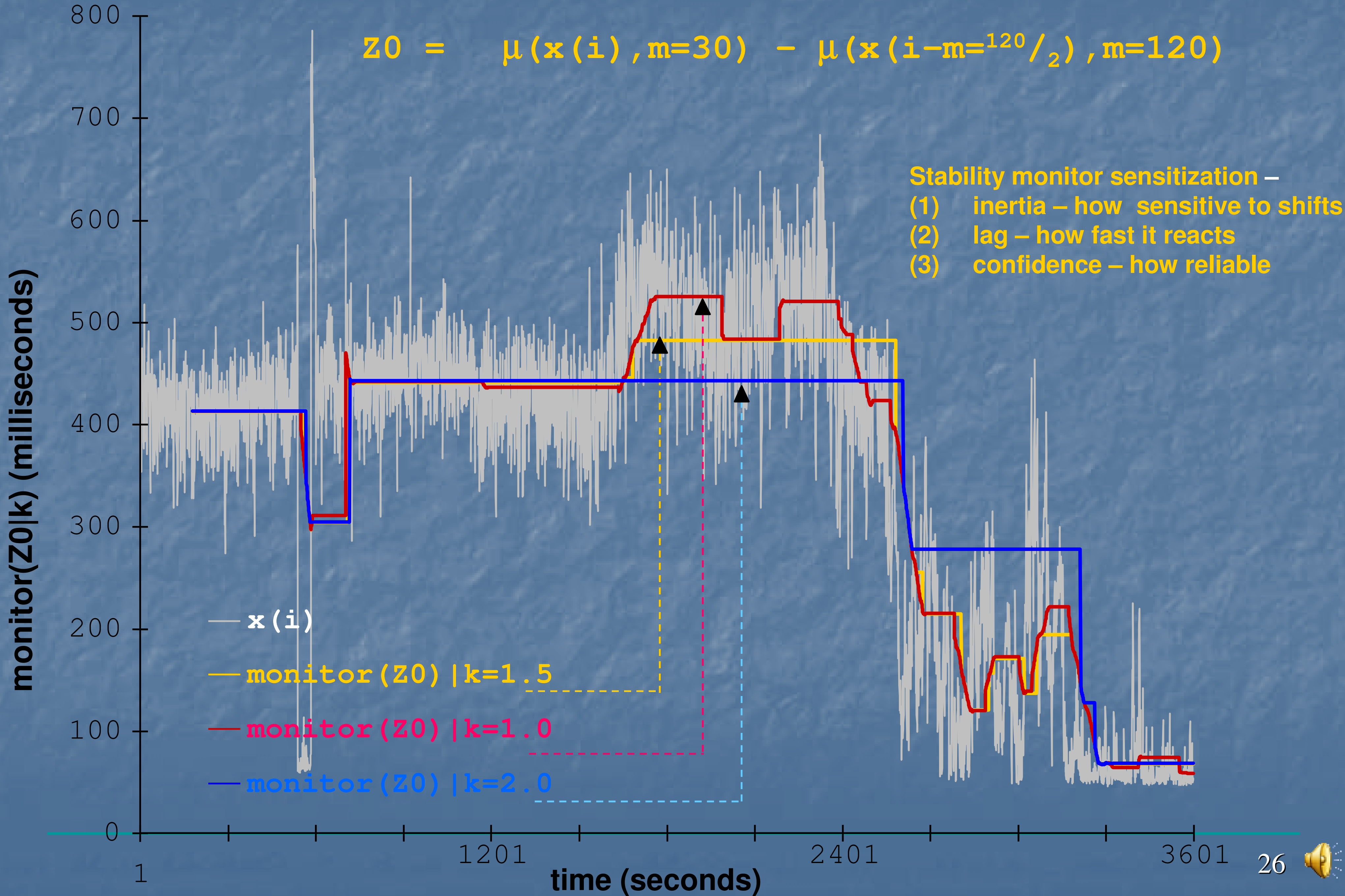
Equality Between Two Population Sampled Means



- ***t*-test on unknown mean and unknown variance for a ~normal population with test statistic**
$$t_0 = \frac{\mu(i,m) - \mu(i-m/2,m)}{\sqrt{\frac{\sigma(i,m)^2 + \sigma(i-m/2,m)^2}{m}}}$$
- **at (α=0.5, m=120) → t ≈ 3.3**

Network Performance Envelope:

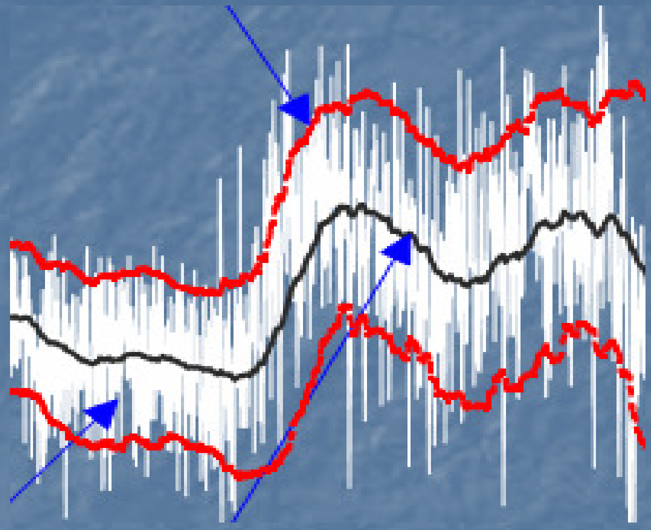
Sensitivity Parametrization

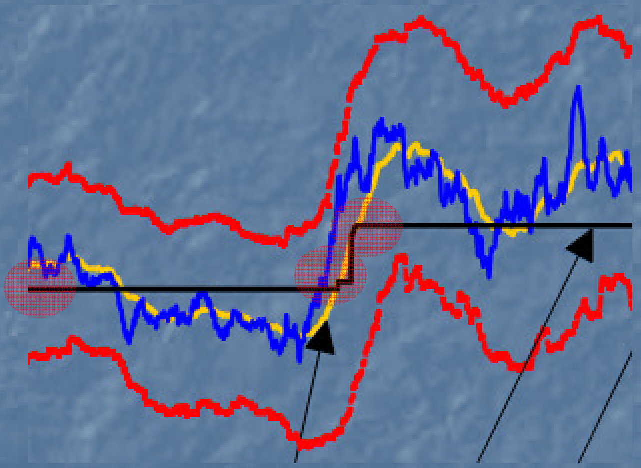


Network Performance Envelope:

Network Stability Monitor Requirements

- **robust process state tracking**
  - quantifiable confidence interval
  - forecast confidence
- **low tracking overhead**
  - moving window kernels have straightforward O(1) complexity
- **easy to implement**
  - moving window process state and control rules are simple (see flowchart)
- **low signaling overhead**
  - process state communicated only when necessary, that is, when it changes (i.e., only when statistically significant)





Research Contributions

- **online generation of a robust floating envelope**
  - showed the particular relevance of online-SPC - despite its simplicity
  - used to track the process performance associated with an indicator
  - shown for RTT – indicator for network congestion and flow management
- **formulated online SPC-based monitor of network state**
  - detection of stationary conditions (temporal stability, **process state**)
  - detection of piece-wise linear (**process changes**)
  - fast, robust, reliable , and low overhead
  - some setup vs. parametrizable statistical performance
- **robust state tracking building block**
  - near-stationarity conditions used to distinguish between process state and process changes
  - timescale and robustness targeted for process-performance

Future Work

- **simulations of SPC+ARC()**
  - further sensitivity analysis of SPC+ARC kernels
    - lag (m, m'), confidence (kσ), smoothing (m), smoother/sampling (UWMA, EWMA, etc.), adaptation (ω) and adaptation strategy (linear, multiplicative, constant), quantization process (mon), outlier recognition (kσ), fitness/residual analysis (err*), etc.
  - further performance g and optimality
    - comparative performance of SPC+ARC()
    - statistical performance (alpha errors, beta errors, etc.)
- **implementation of applications of SPC+ARC()**
  - multimedia networking performance
  - other distributed online process control (see next segment)

Outline of the Talk – Update

- **Background [10%]**
- **Computer-Supported Collaboration [25%]**
- **Dynamically Customized Web Touring [25%]**
- **Multimedia Computing Networking [35%]**
  - Building Robust Network Performance Indicator
  - **Distributed Resource Management for Multimedia**
- **Wrap-Up [5%]**

Distributed Resource Management

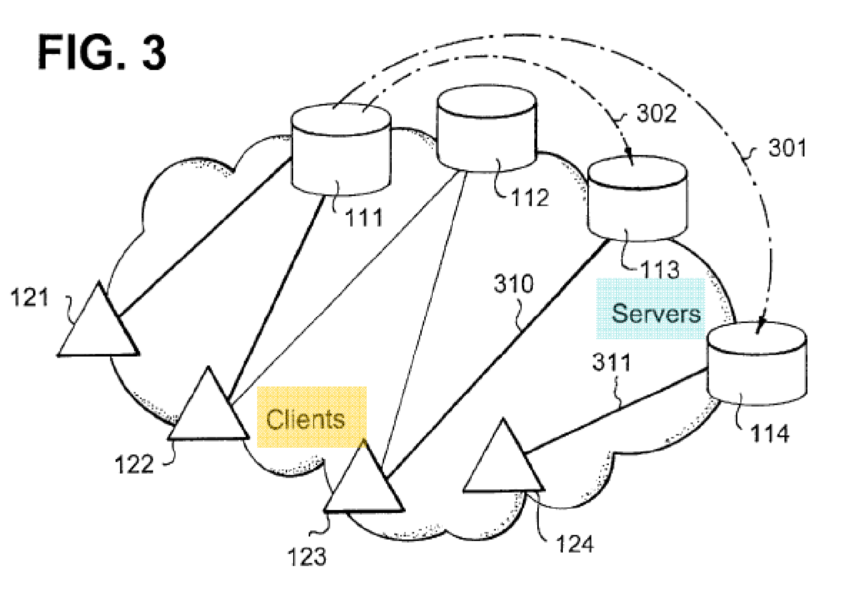
Work at IBM T. J. Watson Research Center

By N. R. Manohar, L. Lumelsky, and S. Wood

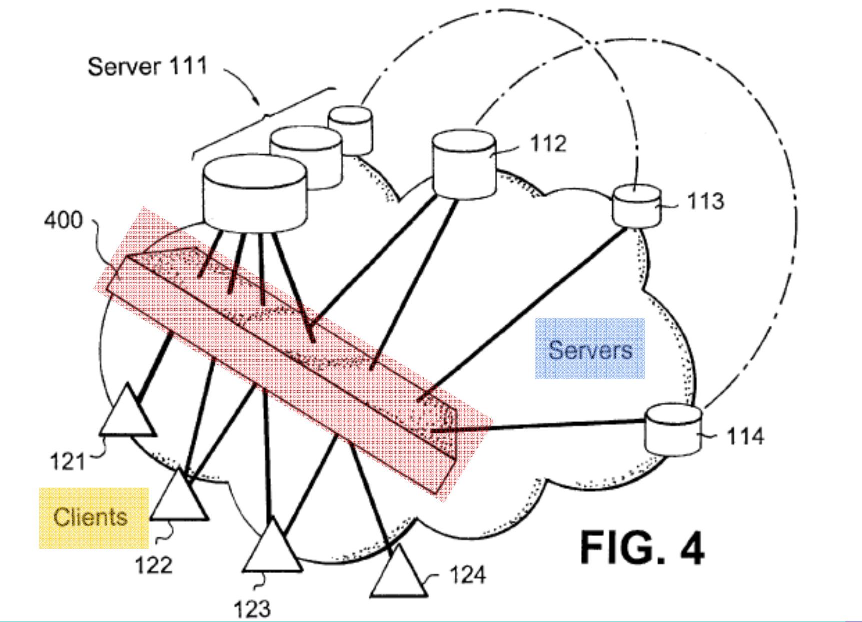
U.S. Patents: *6,516,350; 6,463,454; 6,529,950; 6,377,996; 6,460,082; 6,466,980;*

Distributed Resource Management Plane

- **<u>From</u>: Ad-Hoc Resource Management**
  - ad-hoc client-services
    - streaming, metering, etc.
  - ad-hoc server-services
    - caching, load-balancing, etc.



- **<u>To</u>: Resource Management Plane**
  - distributed management plane
  - brokering of clients to servers
  - standard plane server-services
    - caching, replication, load-balancing, dynamic hosting, etc.
  - compliant plug-in servers



Motivation and Goals

- **Evaluation**
  - Multimedia Computing Networking at Internet2-like networking level
    - very high performance Backbone (vBNS 2.4Gbps OC-12)
    - differentiated class services, RSVP, RTCP, RTP, etc.
- **What could benefit from this?**
  - <u>large</u> multimedia objects under a value-asset model
  - e.g., movies
- **What could we do now?**
  - distributed resource management

Very Large Multimedia Objects?

- **internet-perspective**
  - flow-oriented, end-to-end, distributed QoS/reservation problem
  - non-negligible, considerable, replication cost
- **web-perspective**
  - extremely large in comparison to dominant web objects
  - request characterization of web-servers [MSNBC00]
    - some objects significantly more requested than others
    - small set of objects accounts for majority of requests (Zipf relative frequencies)
  - spatial and temporal locality on requests [IBM97]
    - movie is blockbuster, demographic and temporal consequences
  - and then, some are exceptions, i.e., statistical outliers
    - for example, fads (30s movies, some-actor movies, etc.)
- **provisioning-perspective**
  - value-asset model, digital rights model
  - non-cacheable

Distributed Resource Management?

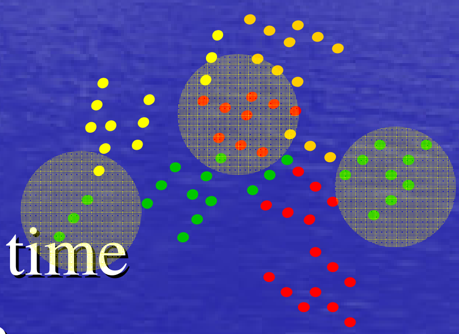
- **What else can we do there? – again, viewpoint matters**

- **adapting demand to the network (variable client demand, fixed server capacities)**
  - balance client demand presented to the network *wrt* fixed server capacities found across the network
  - traditional load balancing, i.e., demand shaping
  - that is, “demand-follows-capacity” policy

- **adapting the network to demand (variable client demand, variable server capacity)**
  - regulate the allocation of server capacities across the network *wrt* client demand presented to the network
  - the other side of the coin, i.e., distributed capacity shaping
  - that is, “capacity-follows-demand” policy

Capacity Shaping – Basic Idea

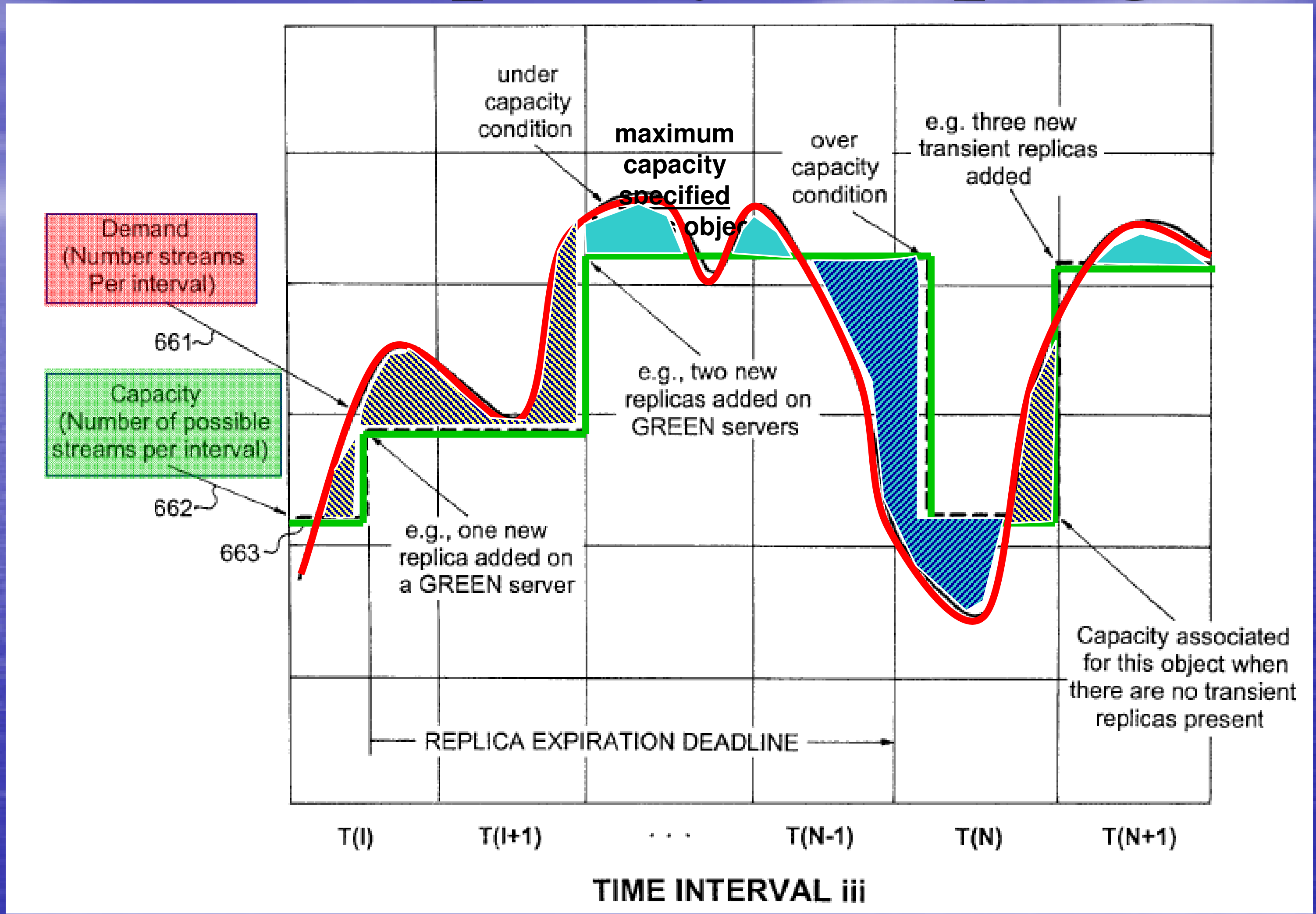
- **capacity shaping mechanism**
  - object is associated with object replicas
  - object replica specifies (serving) resource requirements (i.e., normalized capacity)
  - allocation of an object replica commits the replica’s resource allocation (i.e., normalized capacity allocation)
- **distributed dynamic capacity shaping**
  - expiration time associated with object replica
  - server placement of object replicas varies over time
  - total number of object replicas varies over time



- **thus, allocated object-serving capacity placed over the network shaped over time**
  - done for selected objects – referred to as hot-objects

Self-Regulated Capacity Shaping

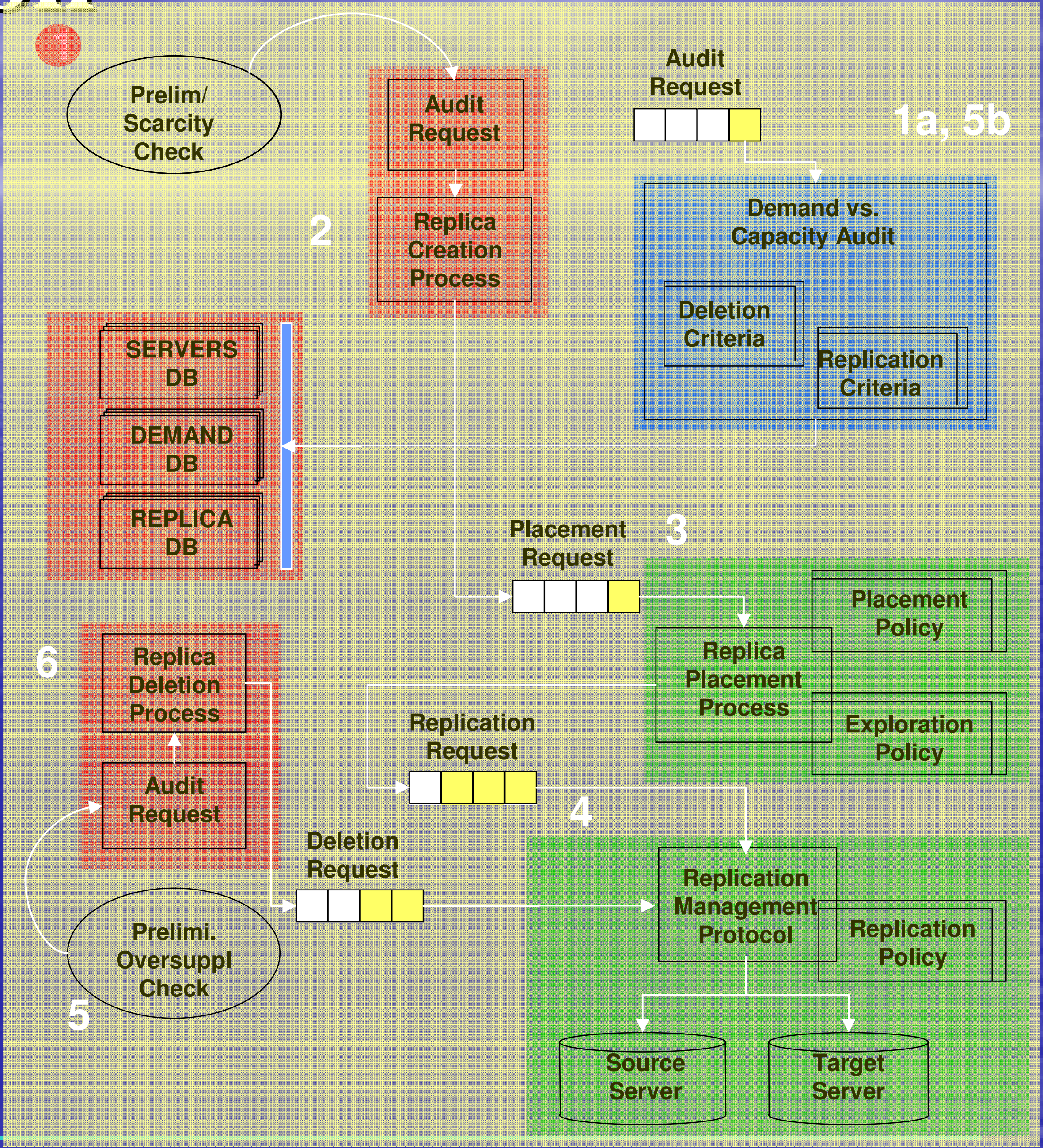
- load balancing and capacity shaping are complementary
- traditionally, throttle control (against fixed capacity) for demand-shaping
- now, self-regulation (against variable capacity) for capacity shaping
- building block: robust state signaling



- dynamic capacity shaping for which objects?
  - Zipf relative frequency distribution – hot objects
- then, how is self-regulated capacity shaping implemented?

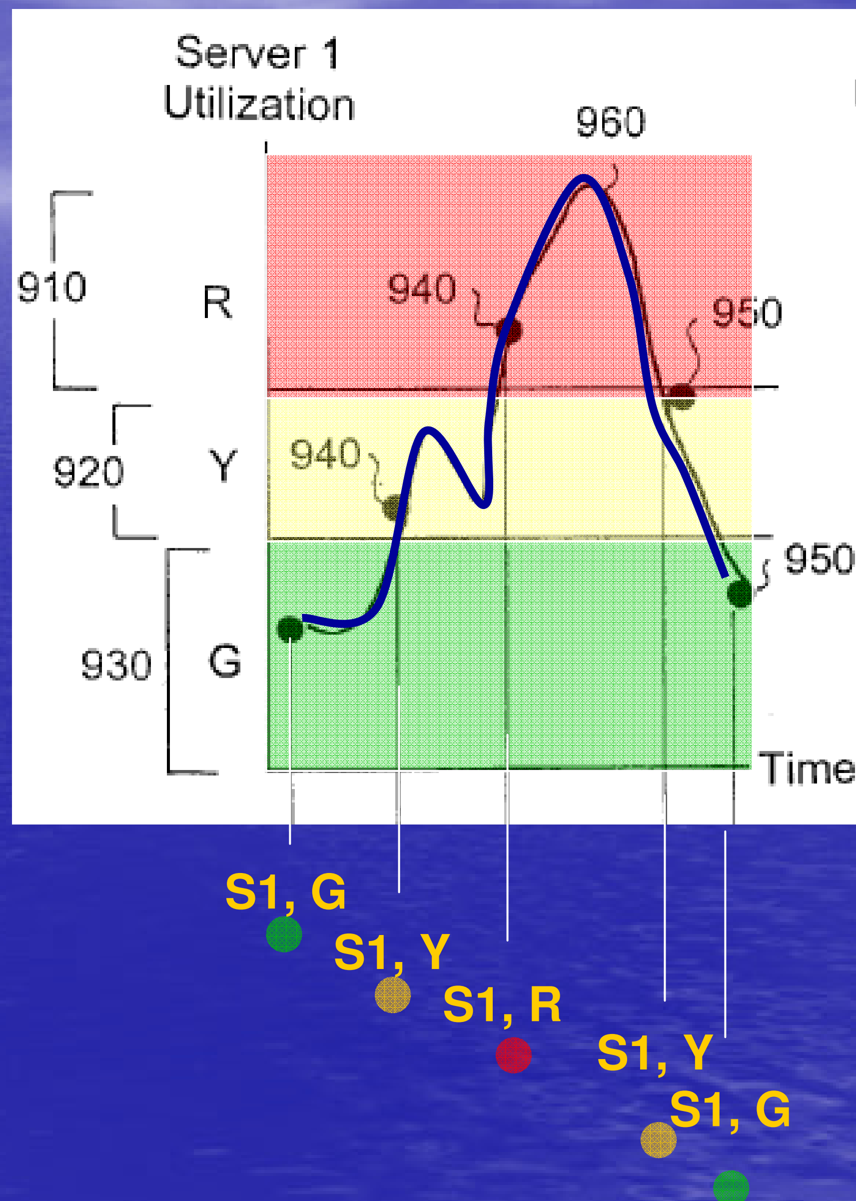
Capacity Shaping Implementation

1. scarcity check
   - demand/capacity analysis (a)
2. replica creation
3. replica placement
4. replication
5. oversupply check
   - demand/capacity analysis (b)
6. replica deletion



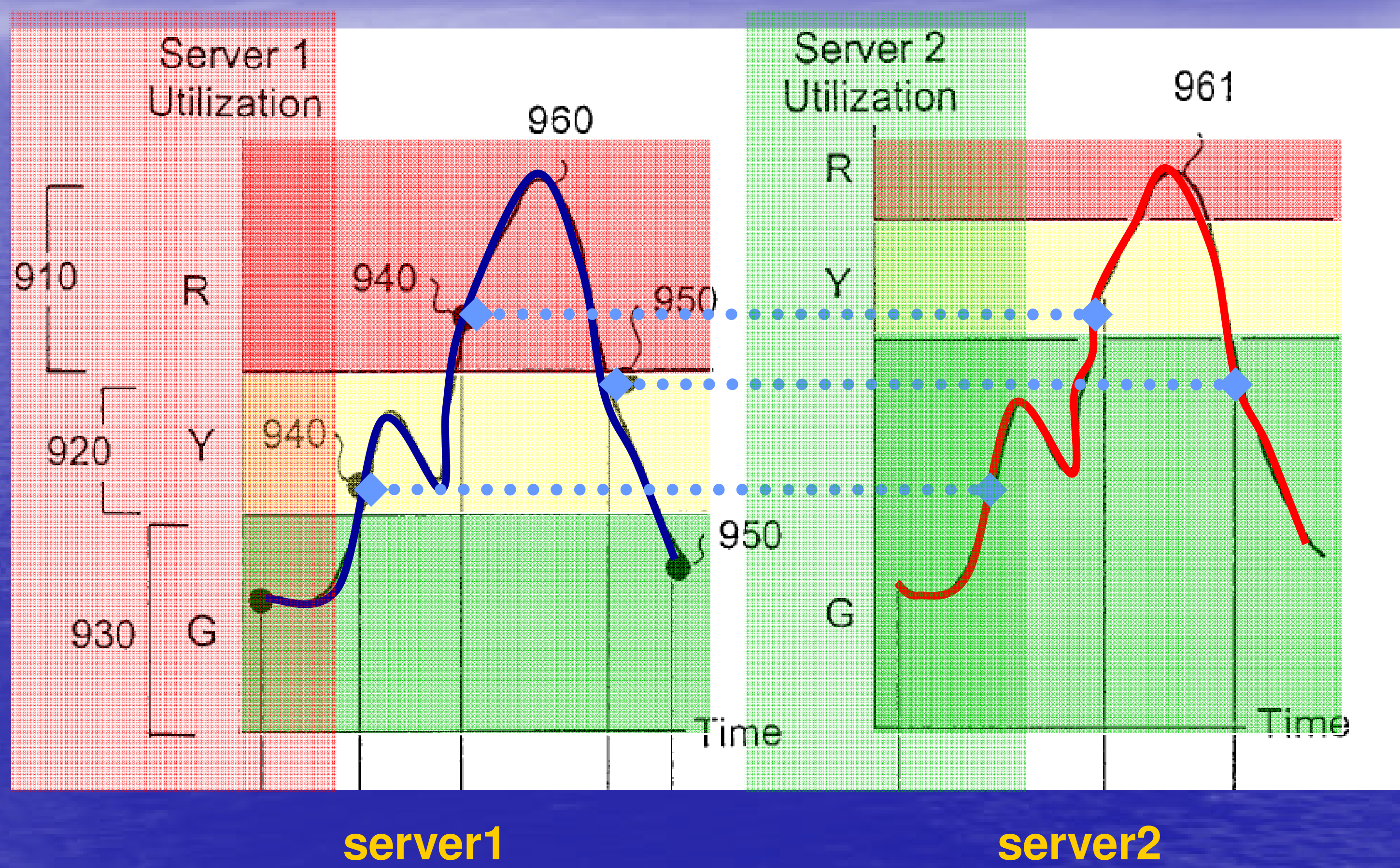
Management of Remote Capacities

- **low overhead on tracking of distributed state**
  - server capacity is normalized
  - semaphore-like
  - green, red, yellow regions
  - trigger-management problem
- **then, to handle transient changes**
  - regions act as buffering regions
  - green
    - safe operating region
    - low management overhead region
  - red
    - critical spare (X%) region
    - non-allocable (safe operating margin)
  - yellow
    - trigger region <u>and</u> buffering region
    - buffers transient state changes



Management of Heterogeneous Servers

- to handle server heterogeneity
  - regions are server specified
- management goal becomes about
  - handling of red servers
- where
  - forewarning through state signaling
  - capacity spare used to manage signaling



Resource Management at Plane

- **Low-overhead State Tracking**
  - quantized demand
  - normalized state & capacity
- **Object to Demand**
  - demand volume
  - object-demand rating
  - interval-tracking
- **Object to Replica**
  - server placement
  - server capacity
  - expiration time
- **Server to Capacity**
  - capacity rating
  - utilization state
  - globality

ObjectID	Demand rate req/s	Volume $t_{(i-1)}$ req	Volume $t_{(i-2)}$ req	Hot Object	Time Stamp
420	10	120	60	yes	t_1
425	5	60	55	no	t_1
428	5	30	62	no	t_0

object 420: hot, large demand

Object_ID	Replica	Server	Transient Replica	Time-to-Live
420	421	1211	NO	
	422	1221	YES	060599-133000
440	441	1211	NO	

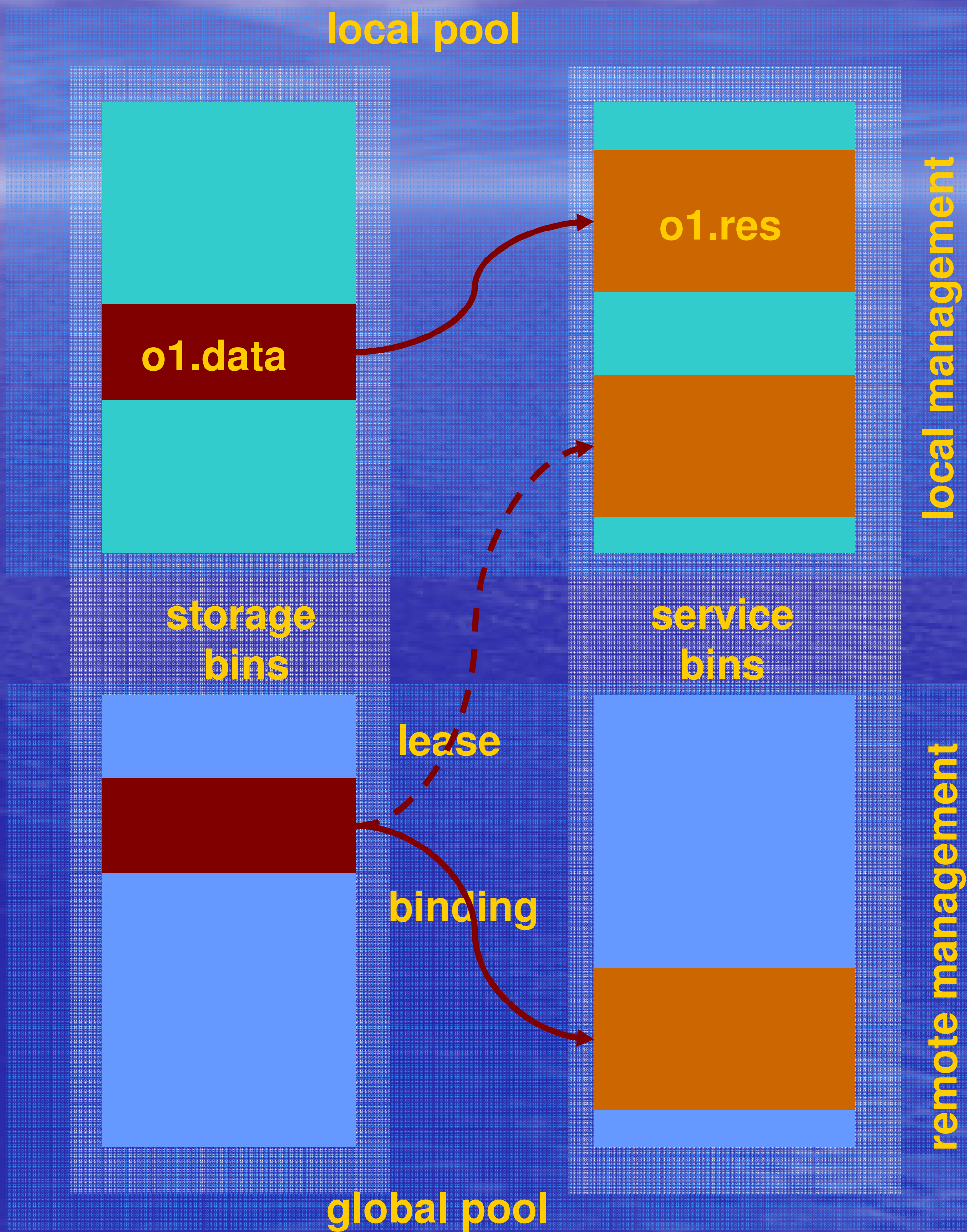
transient replica for 420 at server 1221

Server	IP Address	Capacity Rating	Utilization State	Timestamp	Globality
1211	209.09.9.127	Low	Red	t1	local
1221	128.0.0.1	High	Green	t2	global

global server 1221: high capacity, low use

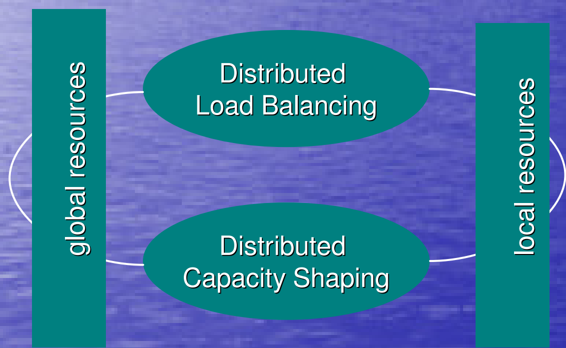
Resource Management at Server

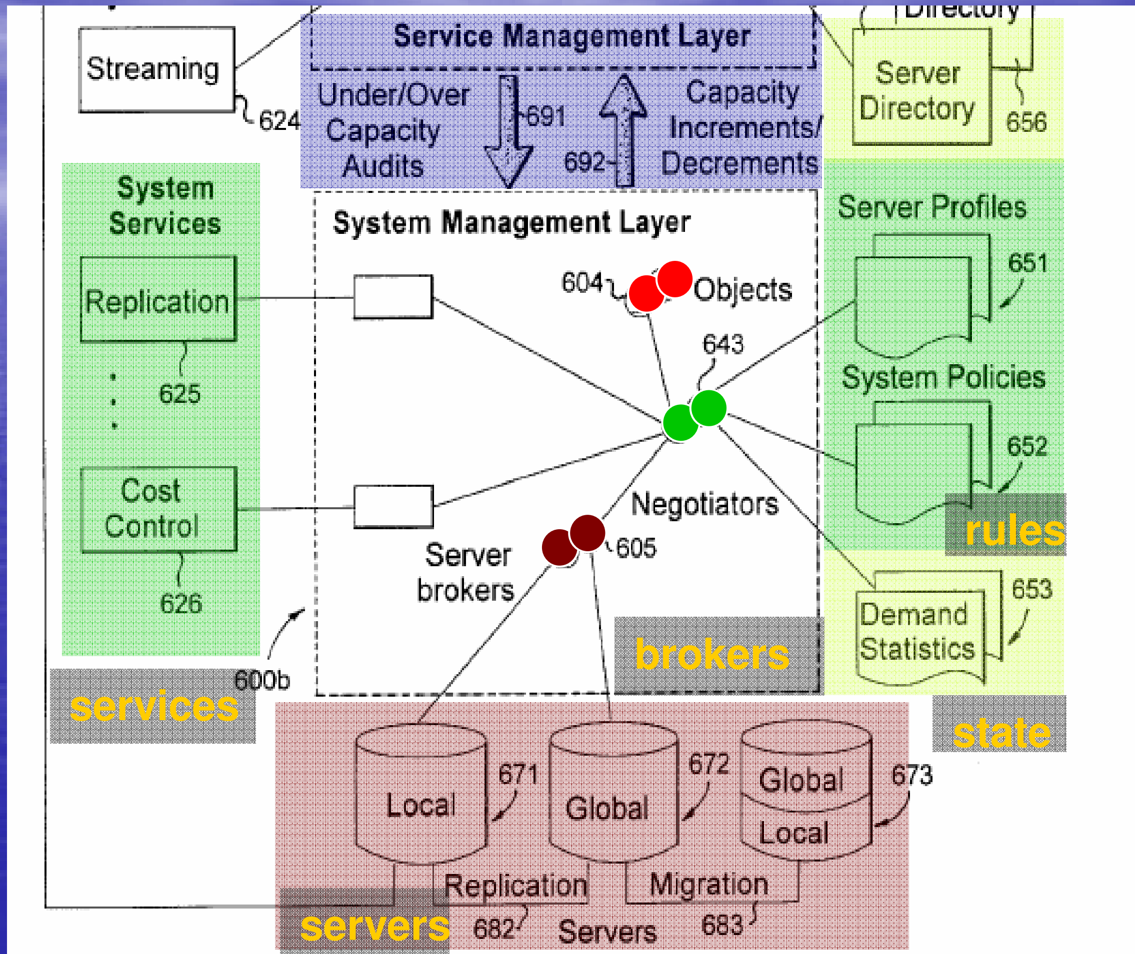
- **reservation bins abstraction**
  - normalized management of replication and provisioning problems
  - storage-bins – for replica placement problem
  - service-bin – for resource reservation problem
- **reservation pools abstraction**
  - for <u>local resource management</u>
    - local serving resources
  - for <u>global resource management</u>
    - *degree-of-freedom* resource for the distributed capacity-shaping problem



Capacity Shaping Management Layer

- capacity shaping plane brokers object replication between servers
  - dynamically creating capacity for load balancing

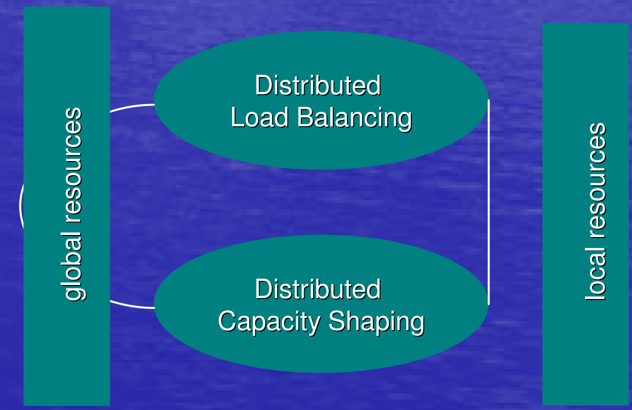




- driven by service management layer
  - by demand-capacity analysis resulting in dynamic replica hosting at brokered servers

Contributions

- **Distributed Dynamic Capacity Shaping**
  - for distributed resource management of large (multimedia) objects over high-bandwidth internet
  - self-regulated distributed capacity shaping
    - integrated load-balancing and dynamic capacity shaping
    - management of global and local replicas *wrt* server capacities
  - distributed state management
    - red-server avoidance management
    - low-overhead distributed state tracking
    - stability-oriented trigger signaling
    - tradeoff of local capacity to remote state tracking
- **complementary to existing technologies**
  - multicast, rights, RSVP, grids, etc.
  - but levering high bandwidth internet (e.g., internet2)



Related Work

- **Dynamic Capacity Virtual Hosting**
  - server farms (e.g., IBM Global Services) – demand-shaping of capacity at centralized center(s) vs. capacity-shaping of demand through placement of distributed capacities
- **Distributed Programmable Planes**
  - grid computing (e.g., Globus) – resource management for entirely different problem, small number of large (computations and datasets) objects with low viewership vs. large number of large objects with high and wide viewership
- **Distributed Replica Management**
  - edge-caching network (e.g., Akamai) – on-demand caching of relatively small objects vs. valuable asset very large object
- **Multicast Streaming**
  - multicast is complementary technology – plane represents the self-regulated placement of the sources of the multicast trees
- **Brokered Distributed QoS Architectures**
  - distributed QoS brokering is complementary technology – delivering end-to-end QoS feasibility used for brokering of clients to servers

Outline of the Talk

- **Background [10%]**
- **Computer-Supported Collaboration (Groupware) [25%]**
  - Record and Replay Paradigm
  - Multimedia Synchronization
- **Dynamically Customized Web Touring [25%]**
- **Multimedia Computing Networking [35%]**
  - Building Robust Network Performance Indicators
  - Distributed Resource Management for Multimedia
- **Wrap-Up [5%]**

Selected Research Contributions

- **Record And Replay (By Re-execution) Paradigm**
  - for asynchronous collaboration, capture of intra-task content
  - manipulation of computer sessions as first class objects
- **Multimedia Scheduling/Synchronization Protocols**
  - integrating fine-grained re-executable events and continuous media
- **Dynamically Customizable Web-touring**
  - touring content control through token-based projections
  - tokens visible and controllable also by user
- **Robust Online-SPC based Process State Indicator**
  - **robust statistical process state indicator (network probing)**
  - **process-performance guiding of adaptive rate control problems**
- **Distributed Resource Management**
  - **self-regulated capacity-shaping (time-variant number and placement of capacities)**

Proposed Focus Areas

- **CSCW/HCI/Groupware Applications**
  - groupware, collaborative intelligence, information management, distance learning, user interfaces, etc.
- **Intelligent Infrastructure**
  - multimedia computing networking, distributed resource management, etc.
  - utility computing, pervasive computing, sensor-based computing,
- **Software Systems Research**
  - applications and middleware systems for the above
  - formalization, theory, experimentation, simulation, etc.

Proposed Courses

- **Classes**
  - Introduction to Databases
  - Introduction to Software Engineering
- **Special Topics**
  - Special Topics: Collaborative Systems
  - Special Topics: Multimedia Computing Networking
- **Advanced Classes**
  - Special Topics: Software Systems Principles
  - Software Systems Research Seminar