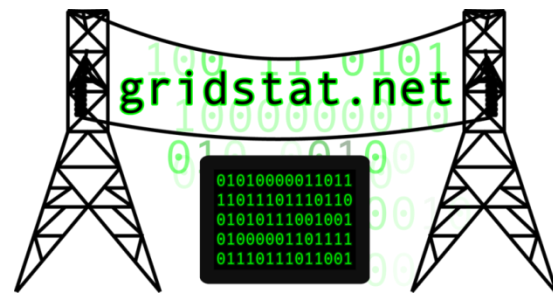# Wide-Area Real-Time Publish-Subscribe for Electric Power Grids

Prof. Dave Bakken

School of Electrical Engineering and Computer Science
Washington State University

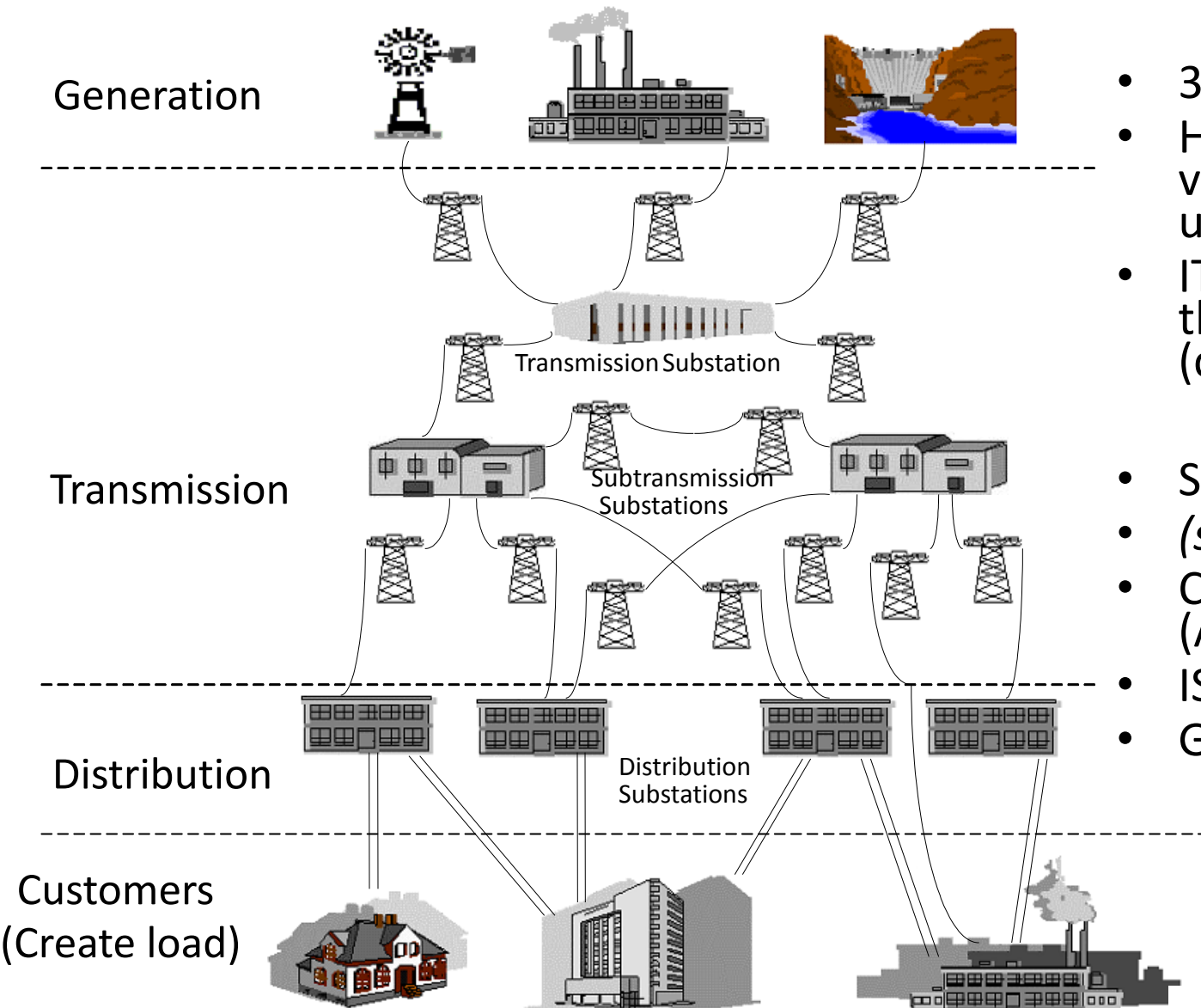BBN Technologies
June 10, 2011

Outline

- **Power Grid Overview**
- North American Synchrophasor Initiative (NASPI)
- Internet vs. Grid Critical Infrastructure
- GridStat Overview
- Advanced GridStat Mechanisms

The Big Picture

- US Power Grid built from bottom up
  - ~3500 entities in US that can affect the grid (1K in CH!)
  - Culturally still seems much like a regulated monopoly (old Ma Bell)
- All power within a single grid is running at the same exact frequency (by definition)
  - Supply (generation) and demand (load) have to be balanced in real-time (frequency drifts…) and sent over long-distance transmission lines
  - Very different from other markets (critical, fast, WAN)
  - **<u>Can greatly benefit from much more sensor data!</u>**
- Grid is amazingly: complex (National Academy century…), under-modeled, under-understood, …

Power Grid Today

Generation

Transmission Substation

Transmission

Subtransmission
Substations

Distribution

Distribution
Substations

Customers
(Create load)

Overview

- 3 fundamental roles
- Historically one vertically integrated utility
- IT/control based on this fixed hierarchy (crude polling)

Hierarchy

- Substation
- *(sometimes sub-area)*
- Control Area/utility (AKA Balancing Area)
- ISO/RTO
- Grid

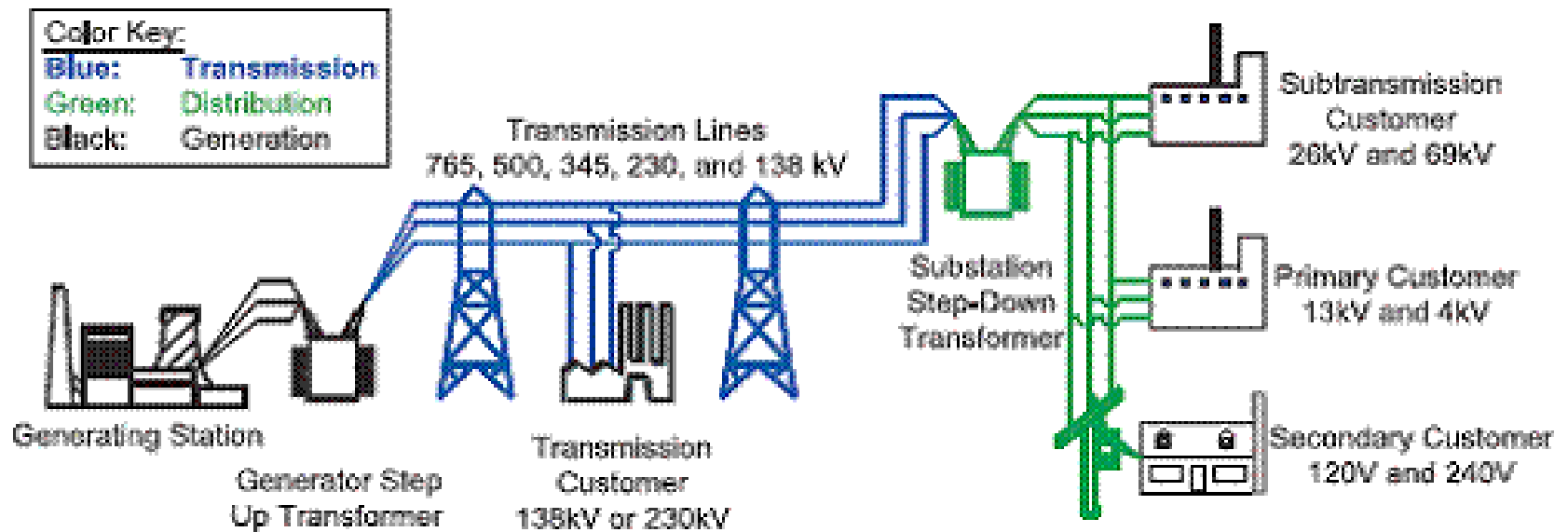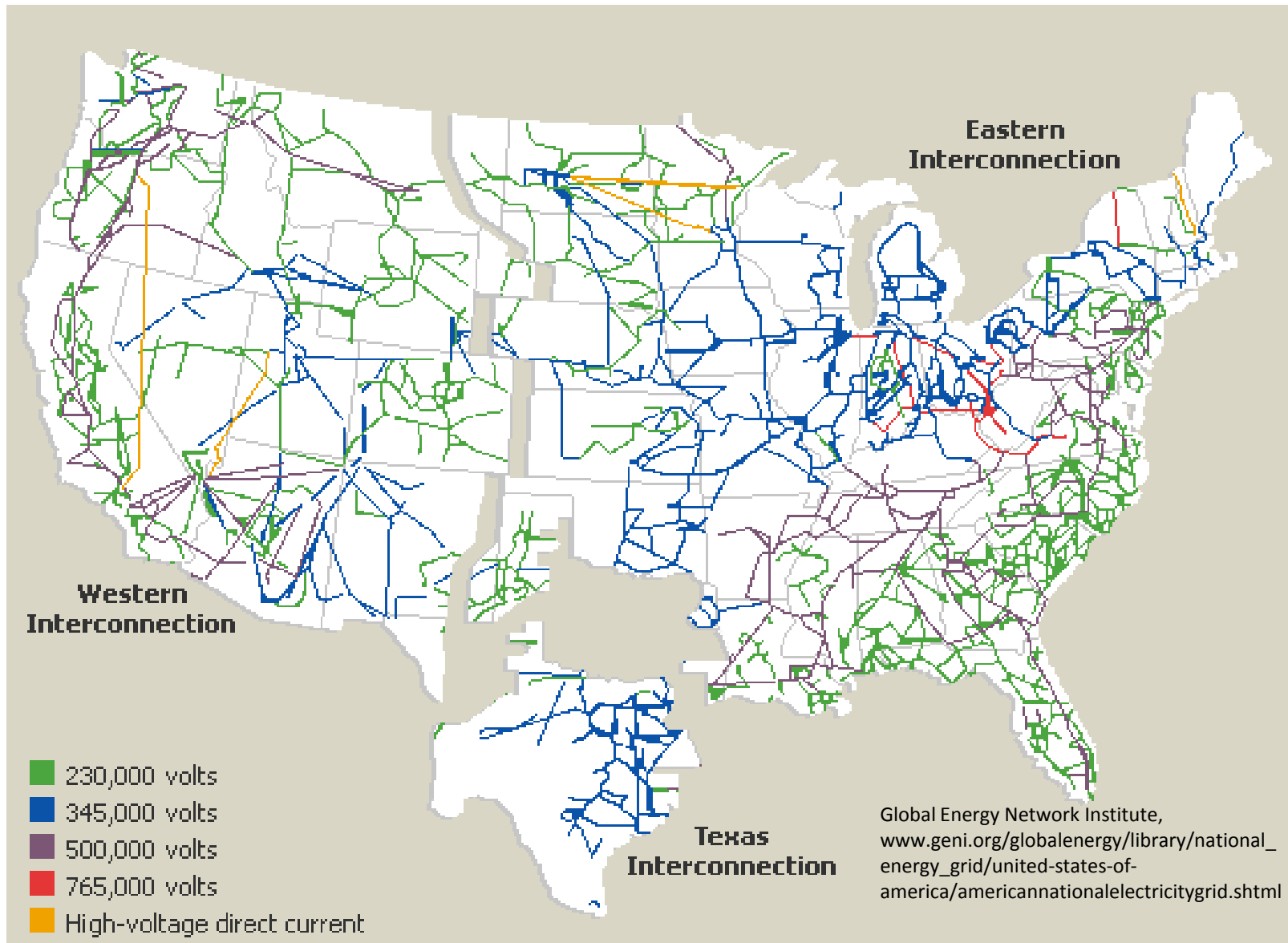Figure credit: NSTAC

Power Grid Today (cont)

Figure courtesy of NERC

US Electric Power Grids

Eastern Interconnection

Western Interconnection

Texas Interconnection

Legend:
- 230,000 volts
- 345,000 volts
- 500,000 volts
- 765,000 volts
- High-voltage direct current

Global Energy Network Institute, www.geni.org/globalenergy/library/national_energy_grid/united-states-of-america/americannationalelectricitygrid.shtml

Extremely complex machines ... National Academies last cent.

The Big Picture (cont.)

- Demand & Generation Outstrips Transmission
- Renewables
- Retiring Operators
- Cyber-attacks

Problem's In Today's Grids

- Reliability
  - Grid is getting more stressed each year
  - WAMS-DD can help (not quite deployed for real yet)
    - Load close to limits➡monitoring tools alert operators to limit violations and the system from instability or collapse: operators can't react fast enough, too vulnerable to contingencies
    - Prevent most (?virtually all) cascading events (e.g., 2003 blackout in NA)

Problems in Today's Grids (cont.)

- Efficiency
  - Day-ahead predictions can be too conservative➔WAMS-DD can potentially help operate grids closer to thermal limits with
    - More efficiency (huge $$$ for even small gains)
    - More inherent safety (if done right)
- Renewable Integration
  - Renewables have different power characteristics than more traditional sources; affects largely unknown
- Retiring Operators
  - Their "seat of the pants" operating knowledge has compensated for very crude WAMS-DD to date
- Cyber-security
- HUGE problem: very little deep (even non-trivial) knowledge across the Power-IT Chasm in utilities, regulatory and government agencies, research communities, ….

Outline

- Power Grid Overview
- **North American Synchrophasor Initiative (NASPI)**
- Internet vs. Grid Critical Infrastructure
- GridStat Overview
- Advanced GridStat Mechanisms

Synchrophasors

- **SCADA** (1960s technology) provides
  - Slow updates (2-4 seconds)
  - Crude: polling architecture, no QoS, little cyber-security
  - Clocks at sensors can be highly unsynchronized
- **Phasor Measurement Units (PMUs)**
  - Idea around for a few decades (Phadke)
  - Practical recently with cheap GPS chips, etc
  - Measure voltage, current, frequency, …. with microsecond accuracy
  - Lets industry move from (nonlinear) "state estimation" towards   direct "state measurement"

Actual Example – Aug. 14th, 2003 Blackout
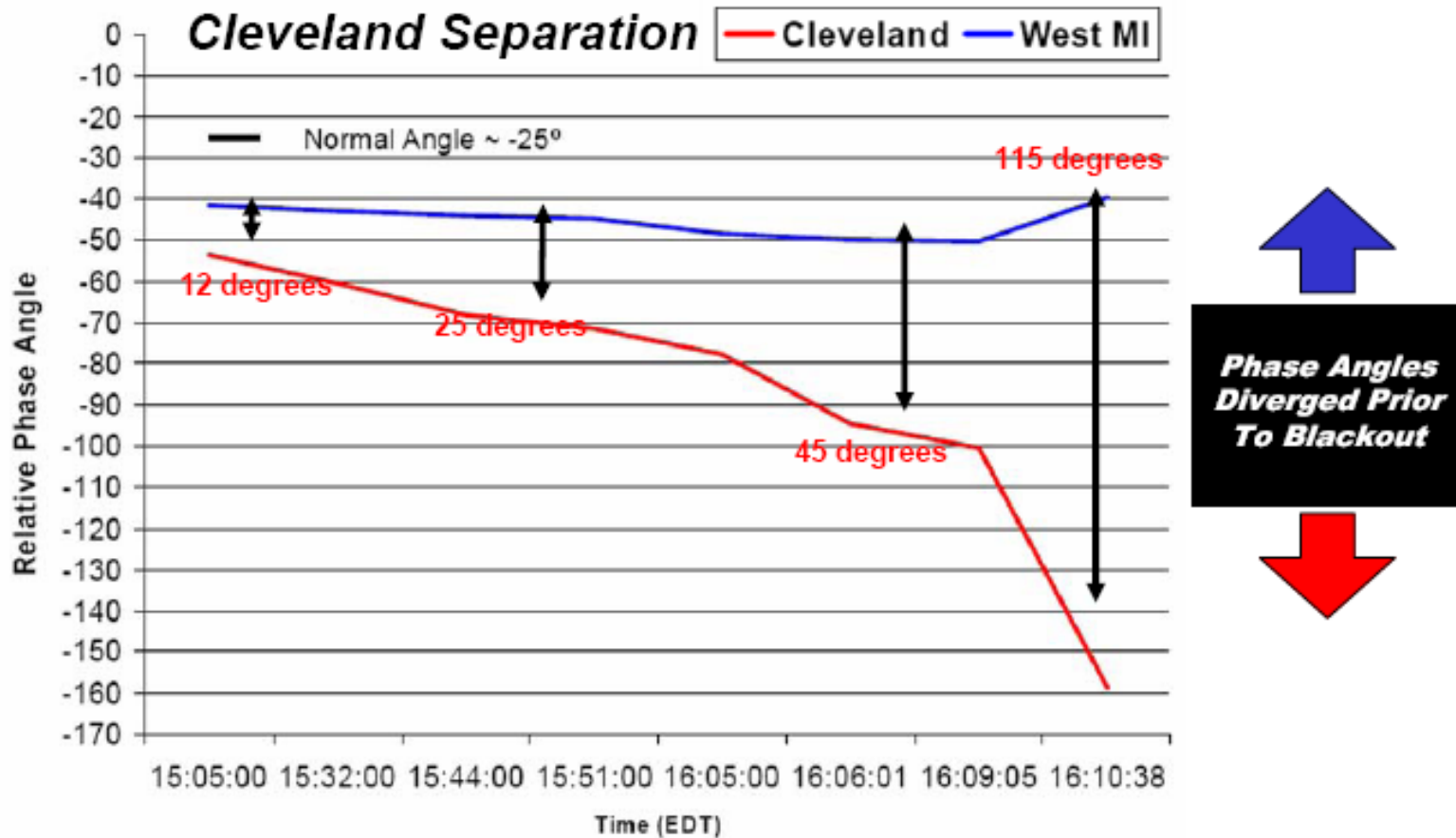
Note: Angles are based on data from blackout investigation.   Angle reference is Browns Ferry.

According to NERC, "A valuable lesson from the August 14 blackout is the importance of having time-synchronized system data recorders. NERC investigators labored over thousands of data items to synchronize the sequence of events. … That process would have been significantly improved … if there had been a sufficient number of synchronized data recording devices."

NASPI

- Vision: "The vision of the North American SynchroPhasor Initiative (NASPI) is to improve power system reliability through wide-area measurement, monitoring and control."
  - Synchrophasor: a sensor with a very accurate GPS clock…
  - Becoming much more deployed in US, Europe, …
- Great need for much better data delivery services
  - Can no longer send "all data to control center at the highest rate anyone might want to"
- Very involved with spec of "NASPInet" services
  - Many requirements come from GridStat research (cited)

NASPInet Conceptual Architecture
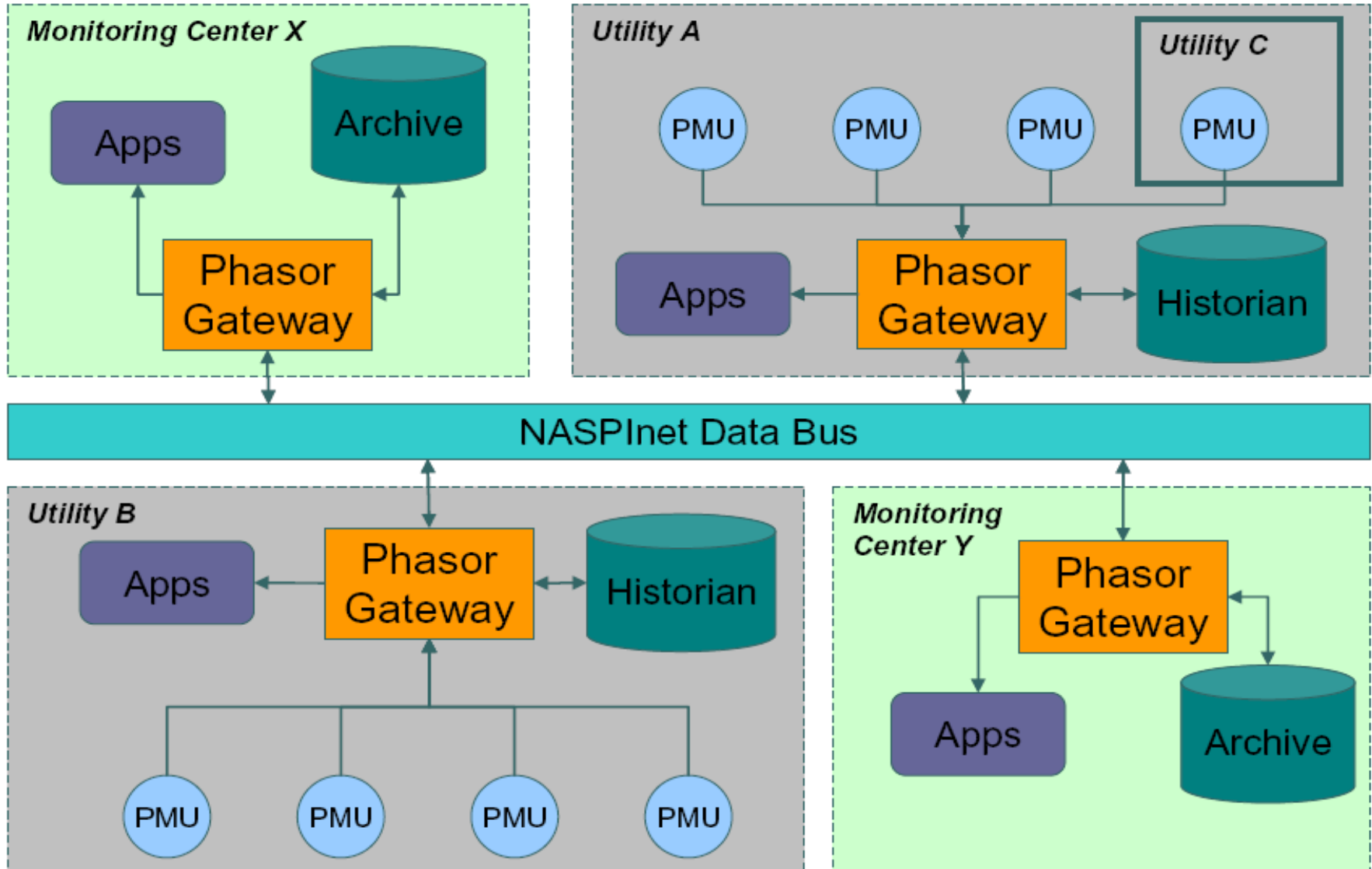
Outline

- Power Grid Overview
- North American Synchrophasor Initiative (NASPI)
- **Internet vs. Grid Critical Infrastructure**
- GridStat Overview
- Advanced GridStat Mechanisms

Grid/CIP Comm. Service Requirements

Kinds of Requirements:

1. Quality of Service (QoS) …

2. Flexibility …

3. Other … (cyber-security, trust management) not discussed here…

- Given now in detail, see Bakken et a. Proceedings of the IEEE, June 2011.

- We believe power grid has most severe data delivery service requirements of all critical infrastructures….

QoS Requirements

- **Latency**
  - 4 ms within substation, 8-12+
- **Rate** (1/minute to 250/second)
- **Availability of Data** (EPRI IntelliGrid 2004)

Level	Availibility (%)	Downtime/Year
Ultra	99.9999	~ ½ second
Extremely	99.999	~5 minutes
Very	99.99	~1 hour
High	99.9	~9 hours
Medium	99.0	~3.5 days

- Delivered QoS **must** be tailorable per data item & changeable (in SW)

Flexibility Requirements

- Multicast (1→many, efficiently)
- Heterogeneity of communication topologies
- Heterogeneity of delivery latency and delivery rate
- Temporal synchronism of rate filtering
- Heterogeneity of computing resources
- Extensibility to new kinds of computing resources
- Open architecture: easy interoperability across multiple vendors

Tomorrow's applications need this flexibility, too: smart grids, advanced metering infrastructure (enabling demand response), distributed generation (microgrids, renewables), …

Internet vs. NASPInet environment

Characteristic	Internet	EPInet
Network size	10^9 interconnected hosts worldwide	10^5 hosts in a power grid 10^{3-4} "routers"
Per-Flow state?	Death (RSVP)	Very feasible
Network design goal	Provide best-effort delivery for any user and purpose	Provide guaranteed QoS in several dimensions for specific users and purposes
Admission Cntl Perimeter	None	Complete
Fraction of Managed Traffic	None/Very Little	Almost all. All traffic subject to policing. >>90% periodic.
Central topology knowledge	Not attempted, because of large scale and dynamicity	Feasible, because of small scale and slow changes
Topology changes (!failure)	Often & without warning	Not often & virtually always with warning (except failure)
Frequency of route changes	Frequent; route changes computed using distributed algorithms that may converge slowly in the face of changing topology	Infrequent; route changes computed centrally assuming stable topology

Internet vs. NASPInet environment

Characteristic	Internet	EPInet
Latency Level Achievable	Slow to Medium	Very Fast
Latency Predictability	Poor	Very Good to Excellent
Recovery delay after dropped packet (with "reliable" delivery)	High (timeout waiting for data or acknowledgement)	Zero (redundant copy sent over disjoint path arrives virtually at the same time) **DO NOT USE post-error recovery, be proactive!**
Forwarding Unit	Uninterpreted packet	Update of a variable
Traffic Predictability	Low	Very High
Elasticity of QoS requirements	None/Low	Medium-High
Multicast: multiple subscribers to a single update flow	A small fraction of the overall traffic; does not justify significant optimization	The common case. Multiple subscribers to a single update flow may have different latency and reliability requirements. Significant opportunity for optimization.

Periodically Updated Variables (PUVs)

- Generic pub-sub system: can NOT drop an arbitrary message when being forwarded

- Rate-based variable update: CAN drop an update if not needed downstream at a given rate
  - AKA rate filtering
  - PASS influence

- Need synchronized filtering w/synchrophasors; E.g.
  - PMU #1: deliver {#1, #11, #21, …}
  - PMU #2: deliver {#2, #12, #22, …}
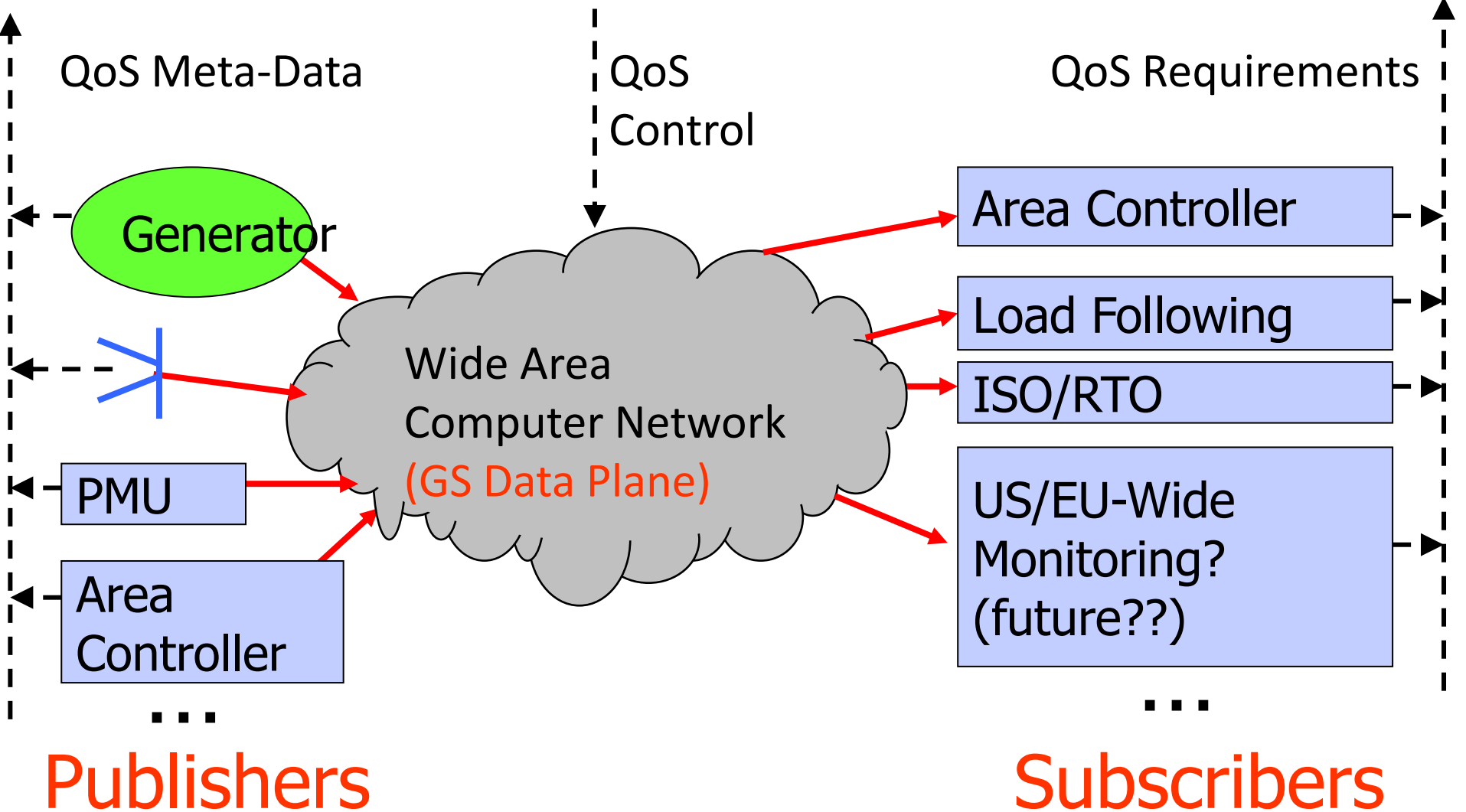 - We call this *temporal synchronization*, AKA *rate decimation*

Outline

- Power Grid Overview
- North American Synchrophasor Initiative (NASPI)
- Internet vs. Grid Critical Infrastructure
- **GridStat Overview**
- Advanced GridStat Mechanisms
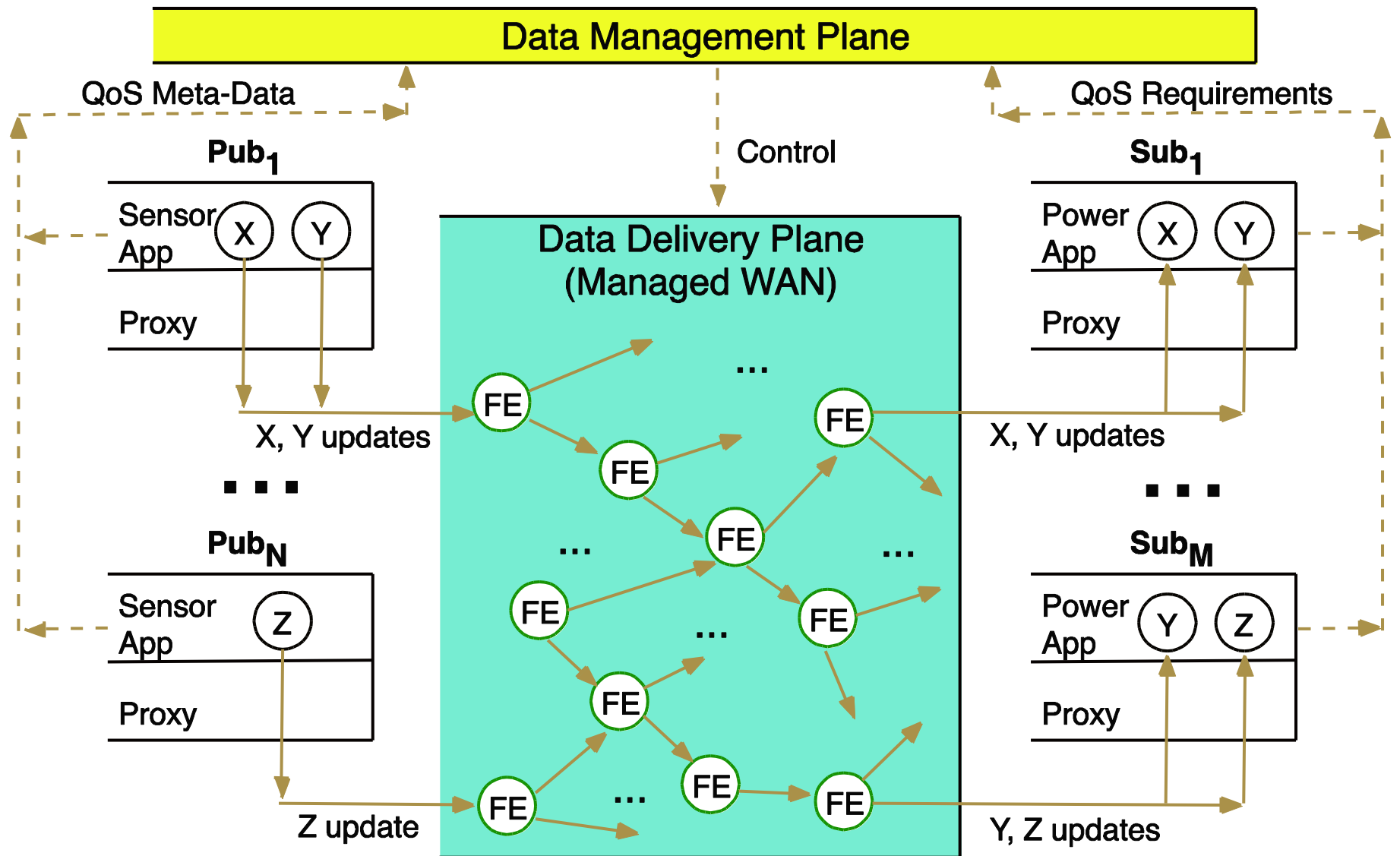
What is GridStat?

- Bottom-up re-thinking of how and why the power grid's real-time data delivery monitoring services need to be

- Comprehensive, ambitious data delivery software suite in coding since 2001
  - Rate-based pub-sub, different subsribers to same variable can get different QoS+ {rate, latency, #paths}

- Rare collaboration of EE (power) and CompSci (distributed computing, networking, …) researchers

- Influencing NASPI's emerging data delivery requirements and architecture
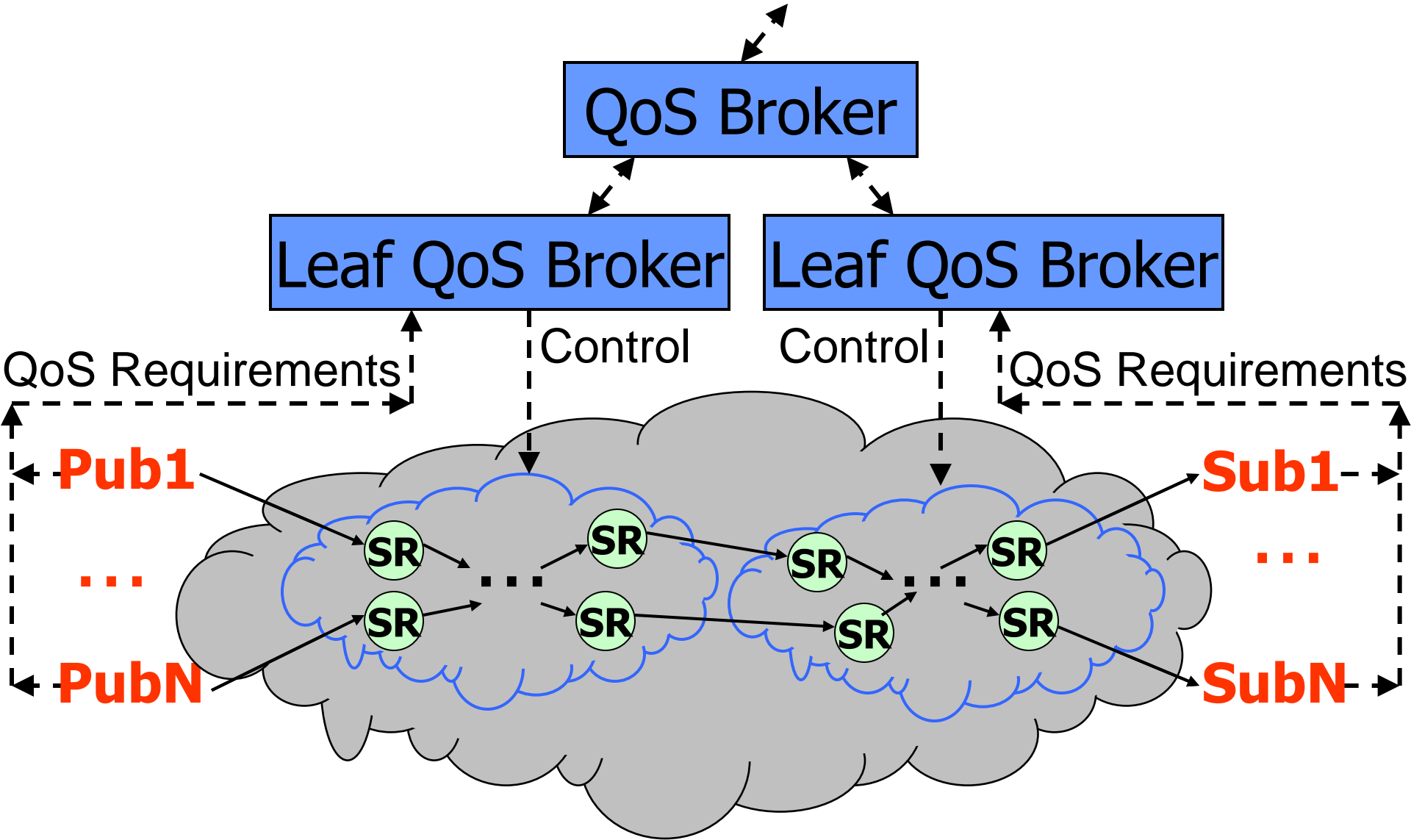
GridStat (GS) Functionality

GS Management Plane

QoS Meta-Data

QoS Control

QoS Requirements

Generator

PMU

Area Controller

...

Wide Area Computer Network (GS Data Plane)

Area Controller

Load Following

ISO/RTO

US/EU-Wide Monitoring? (future??)

...

Publishers

Subscribers

GridStat Architecture

GridStat Architecture

GridStat APIs

- **Pull**
  - A cache instance of the variable kept at each subscribe
  - Subscriber can use just like a local object, when needed
  - Distribution transparency
- **Push**
  - Subscriber can register to get each update
  - Good for database integration (yuk!)
- **QoS Push**
  - Subscriber can register callback to get notified if QoS violated
  - Most apps won't use, but great for aggregation: end-to-end QoS violation

Overview of GridStat Implementation & Perf.

- Coding started 2001, demo 2002, real data 2003, inter-lab demo 2007-8
  - But power industry moves very, very slowly……
    - "Utilities are trying hard to be first to be second" D. Chassin
    - "Utilities are quite willing to use the latest technology, so long as every other utility has used it for 30 years" unknown
  - And NASPI is pretty dysfunctional in a number of dimensions
- Implementations
  - C++, Java: ~2007 0.1 msec/forward, 30k+ forwards/sec
  - Network processor: 2003 HW ~.01 msec/forward, >1M fwds/sec
    - Current network processors are ~10x better, and you can use >1 …
  - Hardware-enhanced security: thesis Dec10 by Thanigai
  - Near future: FPGA (?with current NetProc?), ASIC
    - No need to use IP for core …… (ssshhhh!): less jitter and likely more bullet-proof (no IP vulnerabilities)

Outline

- Power Grid Overview

- North American Synchrophasor Initiative (NASPI)

- Internet vs. Grid Critical Infrastructure

- GridStat Overview

- **Advanced GridStat Mechanisms**

Remote Procedure Call

- Builds a two-way request-reply from a one-way delivery system with QoS+

- Obvious stuff: can set #paths, temporal redundancy, etc for both request and reply messages

- Using GridStat's data reflectively
  - Request: when arrives at subscriber, can abort call if predicate over live GridStat variables returns false
  - Reply: can set timeout and predicate to use physical feedback loop to confirm that RPC request was completed by server

GridStat Modes

- Observation
  - Path allocation algorithms complex, not for a crisis 10^3+
  - But power grid plans way ahead of time
- GridStat supports **<u>operational modes</u>**
  - Can switch forwarding tables very fast
  - Avoids overloading subscription service in a crisis
- Hierarchical
  - can define at Level j, in force at levels ≥ j
  - Implies multiple modes in effect at once in a given FE
  - Coarse way to provision resources
- Two change algorithms: flooding & multi-level commit

Multi-Level Contingency Planning & Adapting

- Electricity example: Applied R&D on coordinated

    1. Power dynamics contingency planning
    2. Switching modes to get new data for contingency
    3. New PowerWorld visualization specific for the contingency

involving contingencies with

    A. Power anomalies
    B. IT failures
    C. Cyber-attacks

- State of art and practice today: 1 & A only, offline
- Very possible: {1,2,3} X {A,B,C} and online

Data Load Shedding

- Electric Utilities can do **load shedding** (I call **power load shedding**) in a crisis (but can really hurt/annoy customers)

- GridStat enables **Data Load Shedding**
  - Subscriber's desired & worst-acceptable QoS (rate, latency, redundancy) are already captured; can easily extend to add priorities
  - In a crisis, can shed data load: move most subscribers from their desired QoS to worst case they can tolerate (based on priority, and eventually maybe also the kind of disturbance)
  - Works very well using GridStat's operational modes
  - Note: this can prevent **data blackouts**, and also does not irritate subscribers

- Example research needed: systematic study of *data load shedding* possibilities in order to prevent *data blackouts* in contingencies and disturbances, including what priorities different power apps can/should have…

- Lets critical infrastructures adapt the data communications infrastructure to benign IT failures, cyber-attacks, power anomalies, …

Conclusions

- Electric power grid has to balance supply and demand in real-time, but is getting more unstable

- Better instrumentation can help the grid's reliability and efficiency

- GridStat is a rate-based pub-sub system optimized for the power grid

Outline

- Power Grid Overview

- North American Synchrophasor Initiative (NASPI)

- Internet vs. Grid Critical Infrastructure

- GridStat Overview

- Advanced GridStat Mechanisms

- **Backup Slides**

  - **QoS Multicast Issues**

  - Scheduling/Routing Issues

Motivation - IP Multicast?

➢ Advantages of IP Multicast

  ➢ Prevents multiple copies of the same bits traversing the same link

  ➢ More efficient than End System or Application Multicast as routing decisions are made by the protocol stack

➢ Disadvantages of IP Multicast

  ➢ Routing is more rigid. Pattern based routing?

  ➢ Lacks support for higher layer functionality

Motivation - End-System Multicast

➢ Advantages

   ➢ Provides Flexible routing as it is managed by end-system applications

   ➢ Facilitates creation of overlay networks and peer-to-peer groups

➢ Disadvantages

   ➢ Slower because decision making happens at the application layer

➢ Comparison and Performance Evaluation - A Case for End System Multicast Yanghua Chu, Sanjay G. Rao, Srinivasan Seshan and Hui Zhang
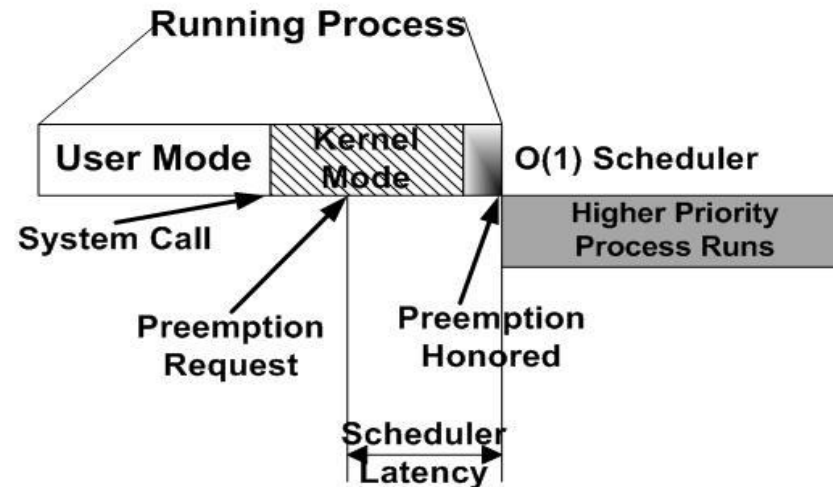
Motivation - QoS Requirements

- ➢ Majority of peer-peer system route Multimedia traffic audio/video streaming, Voice over IP (VoIP)
- ➢ Pub/Sub systems have potential to be used in critical systems (stringent QoS requirements)
- ➢ General Purpose OS limitations and extensions
  - ➢ Widely used
  - ➢ Lack hard real-time capabilities without extensions
  - ➢ Do not exploit more controllable+controlled nature of EPInet
- ➢ Scheduling Algorithm Needed?
  - ➢ Best effort Internet services – congestion, packet loss
  - ➢ UDP commonly used in multicast based applications

Outline

- Power Grid Overview

- North American Synchrophasor Initiative (NASPI)

- Internet vs. Grid Critical Infrastructure

- GridStat Overview

- Advanced GridStat Mechanisms

- **Backup Slides**

  – QoS Multicast Issues

  – **Scheduling/Routing Issues**

Solution – OS Real-Time Support

➢ Factors affecting the real-time behavior of an OS
- ➢ Kernel Preemption
- ➢ Scheduler Latency
- ➢ Scheduler's Run-Time Complexity
- ➢ Priority Scheduling

Ref : Guide To Realtime Programming, DIGITAL UNIX Version 4.0 or higher, March 1996.
http://mia.ece.uic.edu/˜papers/WWW/books/posix4/TOC.HTM

Scheduling Algorithm – Which?

➢ Delay-EDD scheduling algorithm can meet the end to end delay bounds of EPG - Joel Helkey's MS Thesis

➢ In Delay-EDD, each destination 'i' declares its performance requirements in terms of end to end delays

➢ The end to end delay is broken down into local delays at each router

➢ Scheduling is done based on deadline

➢ Packet Deadline is set to the time the packet should have been sent had it been received according to the traffic contract

Scheduling Algorithm – Where?

➢ Under heavy traffic load, incoming bandwidth exceeds outgoing bandwidth

➢ Packets queue at the Queuing Discipline (qdisc)

➢ For UDP, if no qdisc implemented, packets are just dropped. No feedback to the application

➢ Buffers within the application are of no use

➢ Delay-EDD should be implemented as qdisc

• Ref : M. Rio et al. A Map of the Networking Code in Linux Kernel 2.4.20. Research & Technological Development for a TransAtlantic Grid, March 2004.
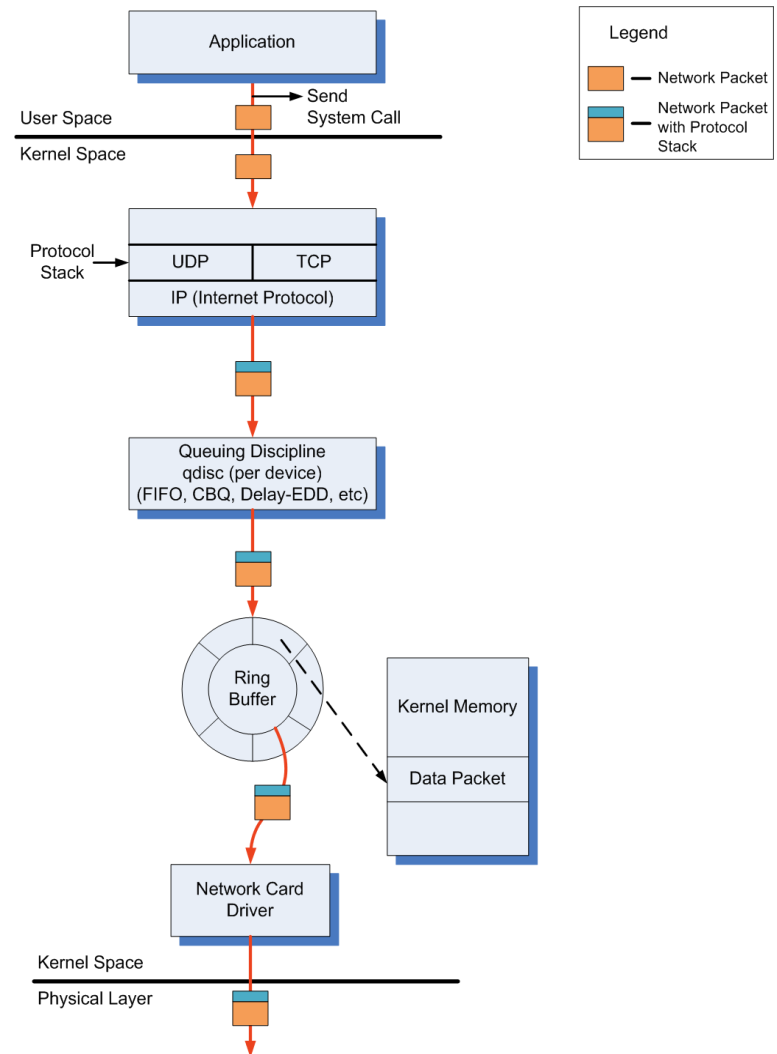
Scheduling Algorithm – Challenges?

➢ Delay-EDD requires minimum inter arrival time to calculate deadline

➢ This information is provided to the Status Router by the publishers-subscribers during connection initiation phase

➢ Implementing Delay-EDD as qdisc (Linux Kernel Module) restricts its ability to communicate with user applications

➢ System Calls bridge the user-kernel space gap

➢ Additional System calls added to the Linux kernel -registerVariable, unRegisterVariable, registerFlow and unRegisterFlow
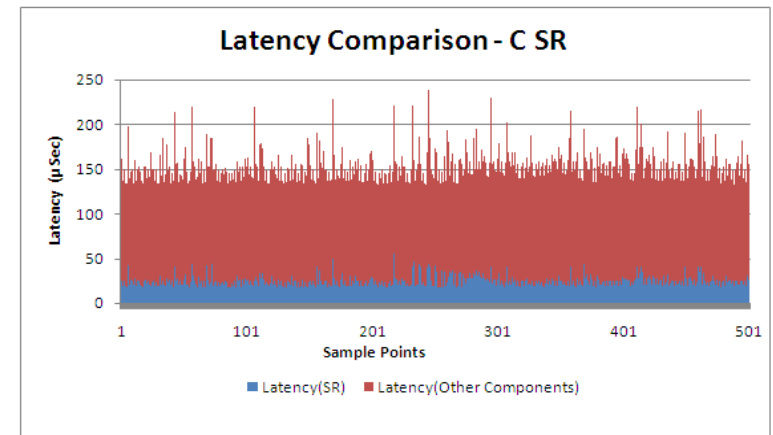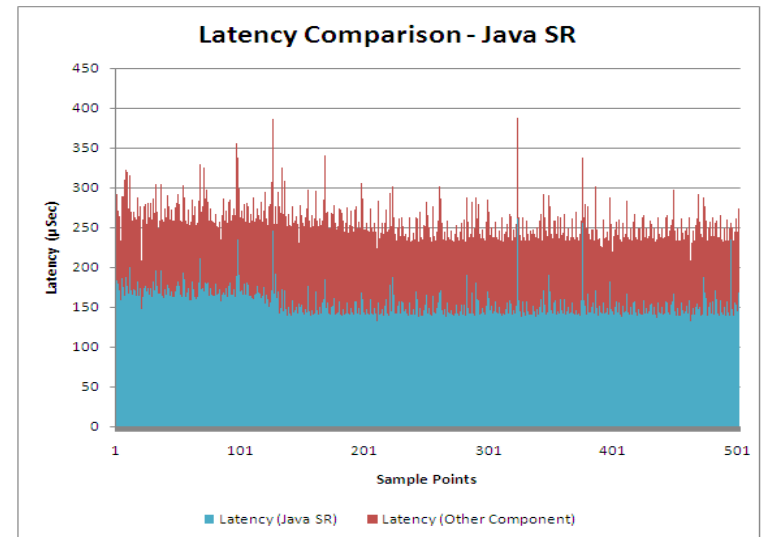
Experiments – Base Experiment

Publishing-Subscribing interval –
  1 ms

Topology

Results

	Latency (μSec)			
SR ▼	Average ▼	Min ▼	Max ▼	Std. Dev. (σ) ▼
C SR	152	128	1042	25.640
Java SR	260	202	11742	94.076
	End-To-End Latency			

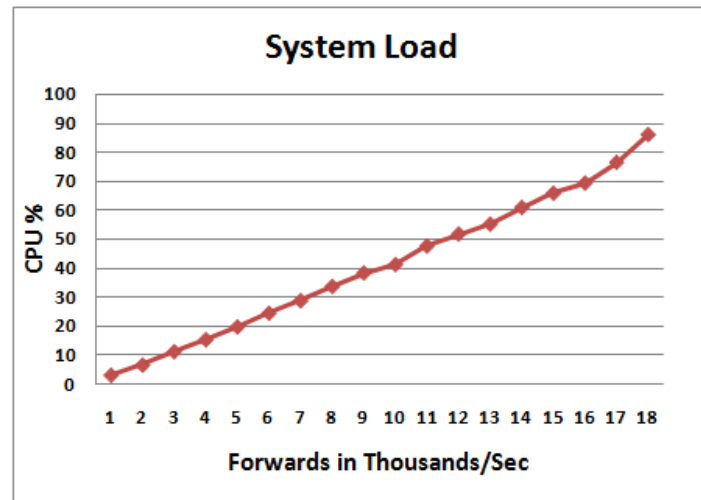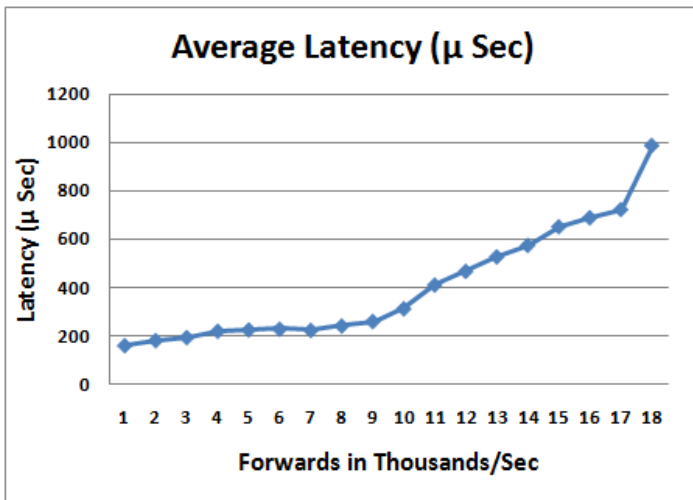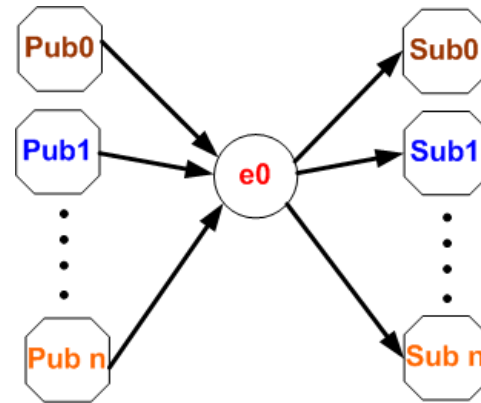|	Latency (μSec)			
SR ▼	Average ▼	Min ▼	Max ▼	Std. Dev. (σ) ▼
C SR	24	16	71	6.046
Java SR	158	107	11617	83.472
	Local Delay at SR			

Experiments – Maximum Throughput/Load

Publishing-Subscribing interval –
   1ms

Topology –

 Results

Experiments – Scheduling Algorithm

- ➢ Delay-EDD Vs FIFO
- ➢ Topology
  - ➢ UDP Client/Server Programs
  - ➢ 12 Load Flows – Packet Send every 10ms, with end-to-end delay bound 20ms
  - ➢ 1 Reference Flow – Packet Send every 1ms with end-to-end delay bound 5ms
- ➢ Results

	Latency (µS)			
Scheduling Discipline	Min	Max	Average	Std. Dev.
Delay EDD	191	2449	741	249
FIFO	4767	39056	5263	266

Status

- SR implemented in C/C++ with delay-EDD packet scheduling in linux

- Additional routing issues - multiple disjoint paths with bounded latency

  - What *is* a good solution to this problem?

    - Find feasible solutions if they exist

    - Minimize use of network resources

    - Problem is NP-hard in several ways: what is a good measure for the quality of a heuristic