Griffin: Towards an Agile, Predictive Infrastructure

Anthony D. Joseph

Sahara Retreat

June 10, 2002

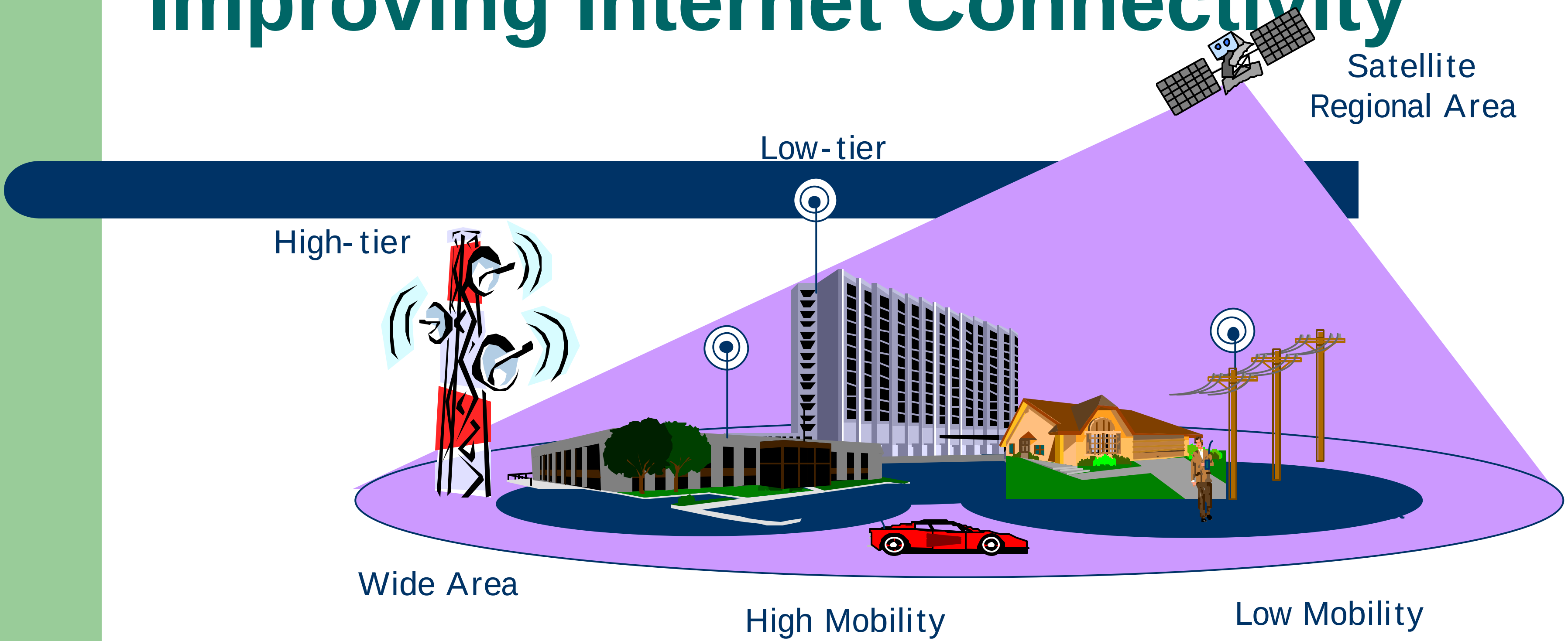
Motivation

- Existing Sahara components provide:
  - Service composition, topology-awareness, brokering / confederation, market-based resource allocation
- Other necessary components
  - Application mechanisms for conveying information to/from the infrastructure
  - Multi-layer network weather modeling and prediction
  - Overlay network for better than IP functionality

⇒ Griffin project

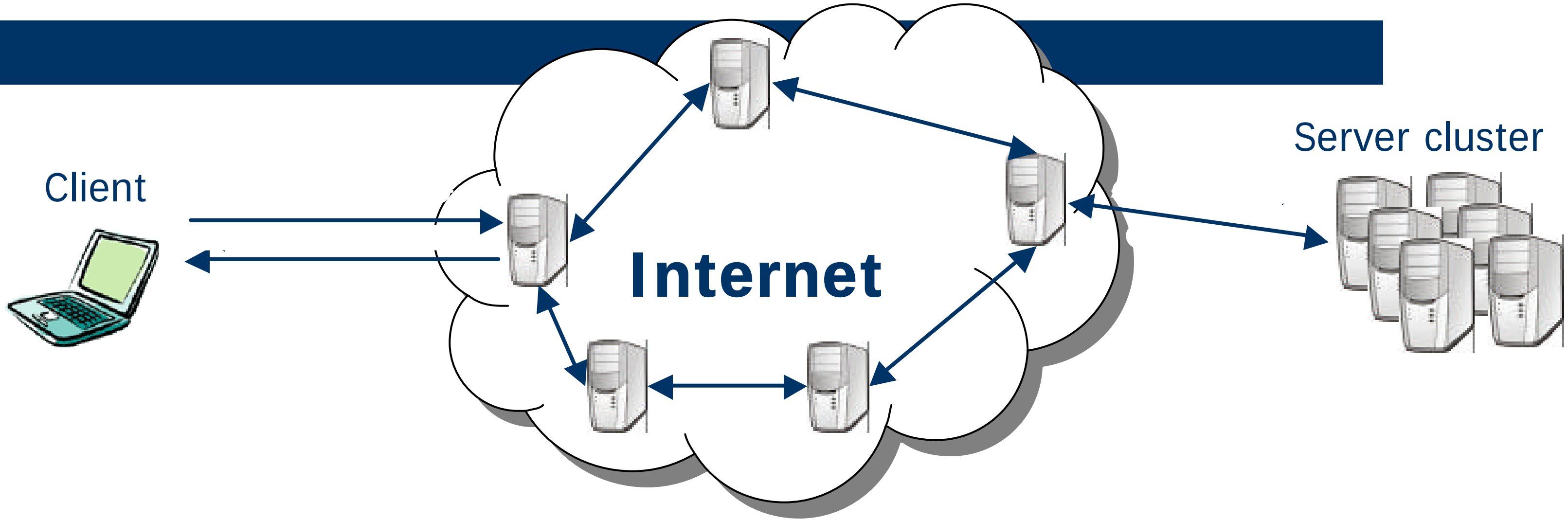
- Focus on agility and behavior prediction for legacy and new applications

Improving Internet Connectivity



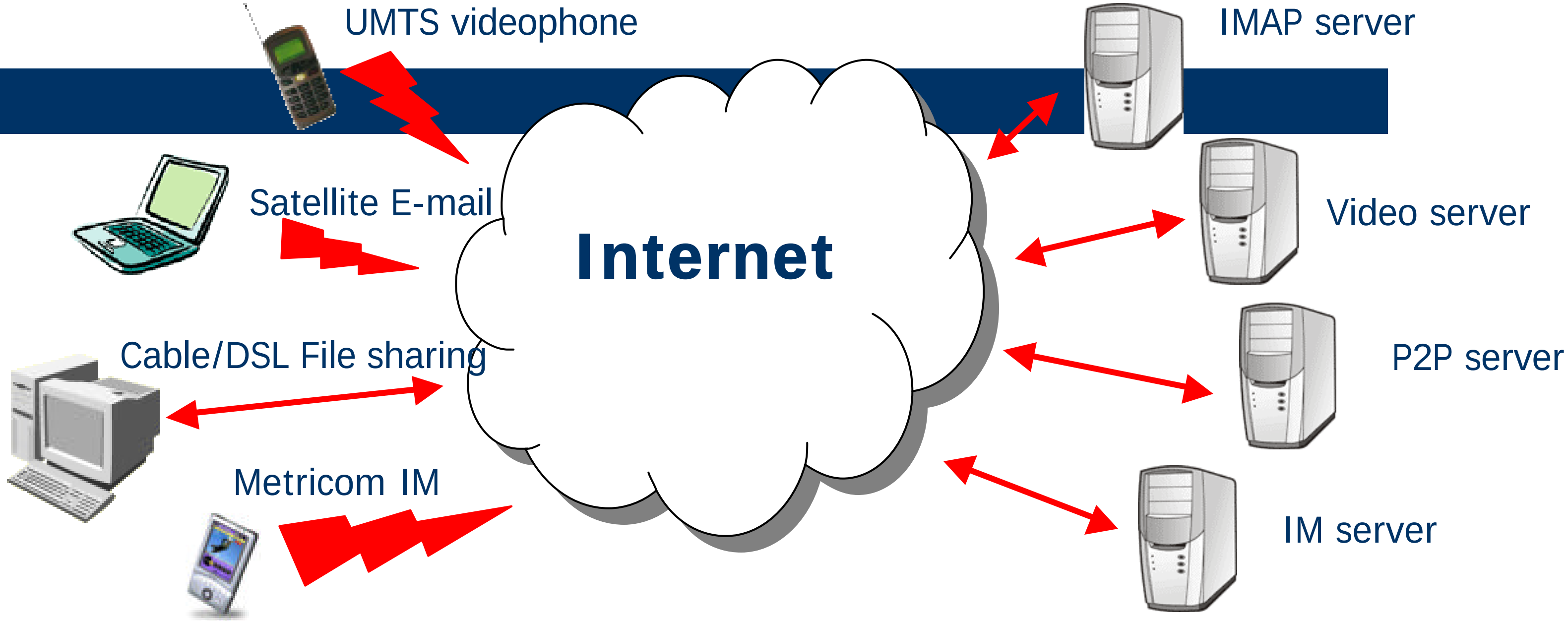
- Near-continuous connectivity: plane, satellite, in-building, ... (BARWAN, Daedalus, Rover)
- But, a laptop app sees a wider range of *variability*
  - 3→5 orders of magnitude of bandwidth from 10's Kb/s→1 Gb/s
  - 4→6 orders of magnitude of latency from 1 μsec →1,000's ms
  - 5→8 orders of magnitude of loss rates from 10^{-4} to 10^{-12} BER

Complex Internet Infrastructure



- Today: 3-tier hierarchy
  - Server, Client, Proxy
  - Static partitioning (edge-side includes, server → client applets)
- Sahara: Dynamic application partitioning
  - Push functionality and data everywhere
  - Broker-based service composition/confederation

New Applications / Mobile Devices



- New real-time apps w/ diff. constraints: latency, BW, …
  - Most apps designed for desktop environment
  - Neither best-effort or unbounded retransmission may be ideal
- Mobile devices mean varying loss/latency/BW conditions
- Result: Poor/variable performance from traditional apps

Problems with Traditional Distributed Applications

- Current approaches are insufficient
  - Static client/server/(proxy) partitioning
  - Strong abstraction boundaries hide differences and variability
- LAN assumption (low latency, low loss, high BW)
  - IP: abstracts different link technologies (wired / wireless)
  - Abstraction boundaries (APIs) hide the number of RPCs
- Added assumption: stability of environment
  - Unlimited power, stable network connection, …
  - Static applications break or perform poorly
  - Dynamic applications built ad hoc w/ “reactive to change” model
- Agility: key metric is time to react and adapt
  - Latency and RTT limit agility

Griffin

- A creature with the head, beak, wings, torso, and claws of an Eagle, and the hind legs and ears of a Lion
  - Signifies the union of strength, agility, and intelligence
  - Sacred to the Greek god Apollo
- Protectors of kings in many Greek stories, guarding their treasures and mines

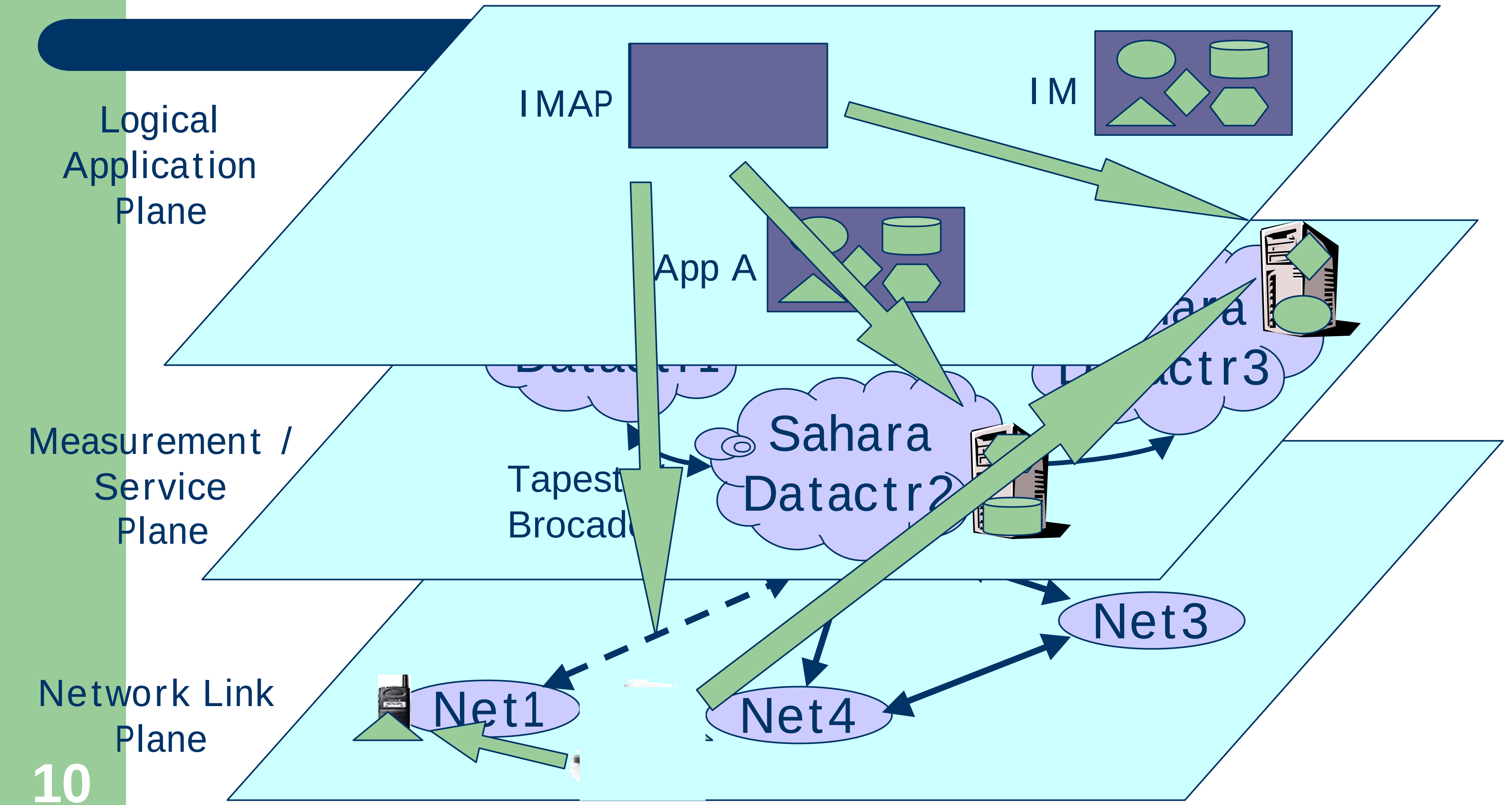
Griffin Goals

- Leverage Sahara policies and control mechanisms
- Users always see excellent (≡ local, lightly loaded) application behavior and performance
  - Independent of the current infrastructure conditions
- Help legacy applications handle changing conditions
  - Analyze, classify, and predict behavior
  - Pre-stage dynamic/static code/data (activate on demand)
- Architecture for developing new applications
  - Input/control mechanisms for new applications
  - Application developer tools

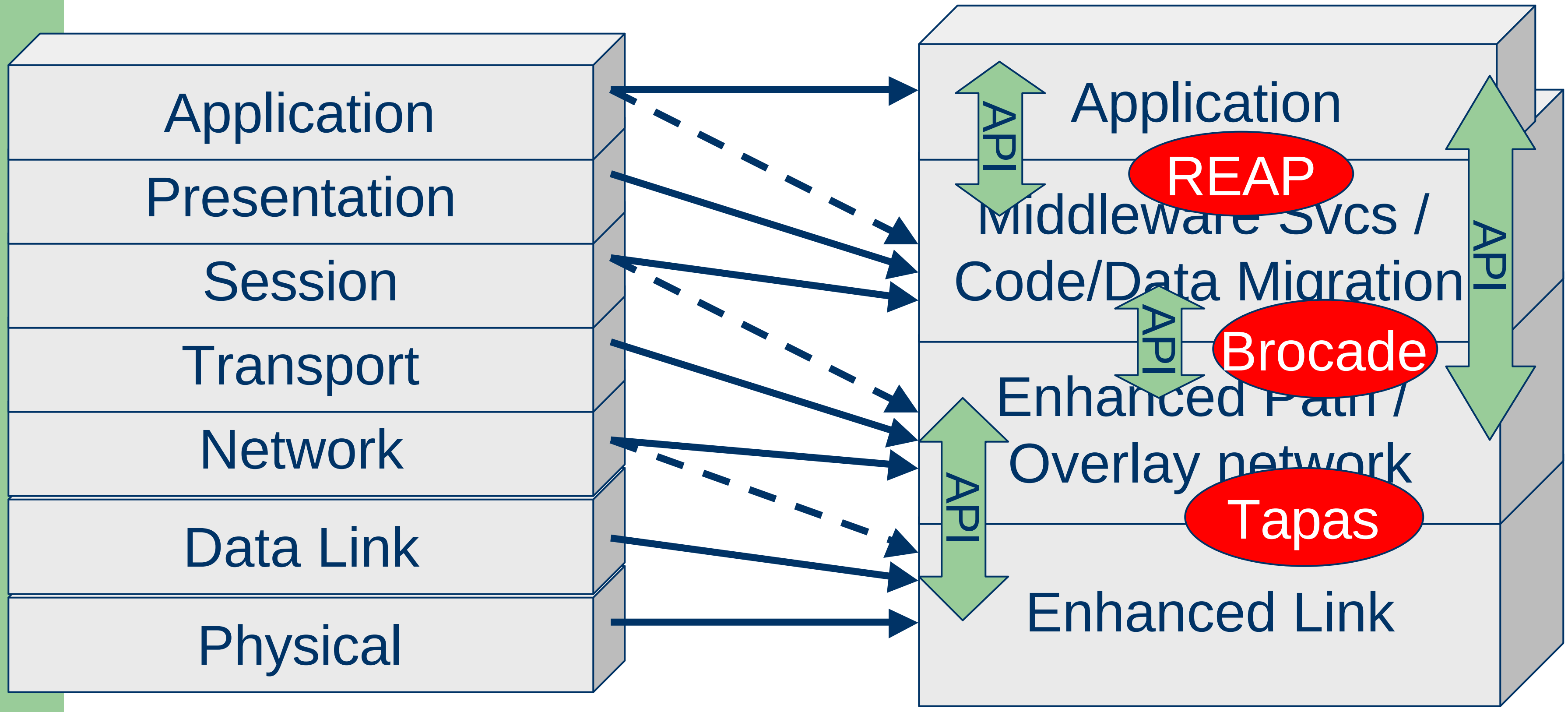
Griffin: An Adaptive, Predictive Approach

- Continuous, multi-level, multi-timescale introspection
  - Collect & cluster link, network, and application protocol events
  - Broader-scale: Correlate AND communicate short-/long-term events and effects at multiple levels (breaks abstractions)
  - Challenge: Data mgmt (RT analysis, storage, propagation)
- Convey app reqs/network info to/from lower-levels
  - Break abstraction boundaries in a controlled way
  - Challenge: Extensible interfaces to avoid existing least common denominator problems
- Overlay more powerful network model on top of IP
  - Avoid standardization delays/inertia
  - Enables dynamic service placement
  - Challenge: Efficient interoperation with IP routing

Towards an Agile, Predictive Infrastructure



Griffin / Sahara Network Layers



Some Enabling Infrastructure Components

- Tapas network characteristics toolkit
  - Measuring/modeling/emulating/predicting delay, loss, …
  - Provides Sahara with micro-scale network weather information
- REAP protocol modifying / application building toolkit
  - Introspective mobile code/data support for legacy / new apps
  - Provides Sahara with dynamic placement of data and service sub-components
- Brocade, Mobile Tapestry, and Fault-Tolerant Tapestry
  - Overlay routing layer providing Sahara with efficient application-level object location and routing
  - Mobility support, fault-tolerance, varying delivery semantics

Tapas

- Novel data preconditioning-based analysis approach
  - More accurately models/emulates long-term and short-term dependence effects than previous approaches
    - Versus Gilbert, n-order Markov
- Tools:
  - Multitracer: Multi-layer trace collection and analysis
  - MTA: A Markov-based Trace Analysis Algorithm
  - M^3: Multi-layer Markov Model Algorithm
  - Synthetic trace generators using both algorithms
  - WSim: Wireless link simulator (currently trace-driven)
- Developing prediction-based feedback algorithms

Some Tapas Results

- A better understanding of effects of link-level effects on network transport
- 1st cut simple socket interface model for communicating with lower protocol stack layers
- Accurate models ⇒ Accurate simulation ⇒ Better application-level protocol design
- Preliminary result: Prediction enables better response time to discontinuous changes in error rate
- See Tuesday morning Sahara talk for more details
- http://www.cs.berkeley.edu/~almudena/tapas/

REAP

- Introspective code / data migration in 3-tier hierarchies
  - Distributes server load, empowers limited devices
  - Provides illusion of high connectivity
- Combines static trace analysis w/ dynamic monitoring of clients to predict appl'n / communication behavior
  - Identify and optimize code/data placement
  - Pre-stage statically/dynamically generated components
  - Explore various granularities of code & data migration
  - Predict costs using multiple criteria
- Building E-mail OceanStore application this summer
  - Exploring conflict resolution strategies

REAP Toolkit

- 1st cut toolkit to explore and test ideas
  - Embeds remote evaluation in servers
  - Clients execute mobile procs (batched protocol cmds) on server
  - Reduces RTT and bandwidth sensitivity
- Showed significant IMAP/SMTP improvements
  - 46 to 89% performance improvement for slow networks
- Protocol event clustering tool
  - 100x reduction in number of states to analyze (really 1,000x)
- See Tuesday morning OceanStore talk about E-mail app
- http://www.cs.berkeley.edu/~czerwin/research.html

Brocade, Mobile Tapestry, and Fault Tolerant Tapestry

- Starting point is Tapestry
  - Distributed Object Location and Routing (DOLR) overlay network built as a part of OceanStore
- Extend Tapestry with unique, powerful routing functions
  - SLA-compatible efficient wide-area routing
  - Rapid, scalable mobility support
  - Rapid fault route-around using pre-computed backup routes
  - Monitoring, measurement, and analysis entry point
- See Monday joint and Tuesday OceanStore talks for more details
- http://www.cs.berkeley.edu/~ravenben/tapestry/

Brocade

- Overlay networks can suffer from inefficient cross-domain routing
  - Higher per message latency and wasted wide-area bandwidth
- Brocade
  - Eliminates unnecessary wide-area hops for inter-domain msgs
  - Reduces wide-area bandwidth utilization
  - Intuition: route directly to destination domain / AS
- Results:
  - 60 to 70% reduction in Relative Delay Penalty (latency)
  - Up to 75% reduction in average message bandwidth in wide area cases

Mobile Tapestry

- Handle large-scale, rapid, simultaneous node mobility
- Handling basic mobility
  - Add a layer of indirection to Tapestry by treating nodes as objects and using Tapestry object location
  - Uses Tapestry locality-based routing to reduce overhead
- Handling rapid, simultaneous mobility
  - Add a layer of hierarchy
- Results:
  - 87% reduction in RDP (latency)
  - 75% reduction in msgs for 1K simultaneous roaming nodes
  - 1000x reduction in msgs for 1K nodes roaming together

Some Open Issues

- Challenges of prediction
  - Time to disseminate information and predictions
  - Accuracy of predictions
- Complexity of building agile applications
  - Domains of applicability: latency, bandwidth, congestion, ???
  - Extensible interfaces: How general? Approaches?
- SLA compatibility
  - Extra cost? Loss of fault tolerance?
  - Extracting policies from BGP

Next Steps

- Introspection
  - Exploring new modeling and clustering algorithms
  - Exploring prediction algorithms
- Explore data migration in OceanStore
  - IMAP/SMTP E-mail application
- Standardize DOLR interface
  - Deploy testbed on Intel’s PlanetLab
  - Develop guarantee-based analysis for Fault Tolerant Tapestry

Griffin: Towards an Agile, Predictive Infrastructure

Anthony D. Joseph

Sahara Retreat

June 10, 2002