Outline

- Griffin
  - Motivation
  - Goals
  - Architecture

- Tapas
  - Motivation
  - Multi-layer protocol trace collection
  - Data preconditioning-based network modeling
  - Network feedback and prediction
Near-Continuous, Highly-Variable Internet Connectivity

- Connectivity everywhere: campus, in-building, satellite…
  - Projects: Sahara (01-), Iceberg (98-01), Rover (95-97)
- Most applications support limited variability (1% to 2x)
  - Design environment for legacy apps is static desktop LAN
  - Strong abstraction boundaries (APIs) hide the # of RPCs
- But, today’s apps see 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→9 orders of magnitude of loss rates from 10⁻³ → 10⁻¹² BER
  - Neither best-effort or unbounded retransmission may be ideal
  - Also, overloaded servers / limited resources on mobile devices
- Result: Poor/variable performance from legacy apps
Griffin Goals

- Users always see excellent (= local, lightly loaded) application behavior and performance
  - Independent of the current infrastructure conditions
  - Move away from “reactive to change” model
  - Agility: key metric is time to react and adapt
- 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
- Leverage Sahara policies and control mechanisms
Griffin: An Adaptive, Predictive Approach

- Continuous, cross-layer, 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: Building accurate models of correlated events

- 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 policies
Some Enabling Infrastructure Components

- **Tapas network characteristics toolkit**
  - Measuring/modeling/emulating/predicting delay, loss, …
  - Provides Sahara with micro-scale network weather information
  - Mechanism for monitoring/predicting available QoS

- **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
Outline

- **Griffin**
  - Motivation
  - Goals
  - Architecture

- **Tapas**
  - Motivation
  - Multi-layer protocol trace collection
  - Data preconditioning-based network modeling
  - Network feedback and prediction
Tapas Motivation

- Accurate modeling and emulation for protocol design
  - Very difficult to gain access to new or experimental networks
  - Delay, error, congestion in IP, GSM, GPRS, 1xRTT, 802.11a/b
  - Study interactions between protocols at different levels
- Creating models/artificial traces that are statistically indistinguishable from traces from real networks
  - Such models have both predictive and descriptive power
  - Better understanding of network characteristics
  - Can be used to optimize new and existing protocols
- Answer several application design questions
  - Q1: Impact of network layering in application design
  - Q2: Effects of network model choices on results (survey?)
  - Q3: Using feedback for adaptive, predictive applications
Tapas

- Novel data preconditioning-based analysis approach
  - More accurately models/emulates long-/short-term dependence effects than classic approaches (Gilbert, Markov, HMM, Bernoulli)

- Analysis, simulation, modeling, prediction tools:
  - MultiTracer: Multi-layer trace collection and analysis (download)
  - Trace analysis and synthetic trace generator tools
    - Markov-based Trace Analysis, Modified hidden Markov Model
  - WSim: Wireless link simulator (currently trace-driven)
  - Simple feedback algorithm and API
  - Domain analysis tool: chooses most accurate model for a metric

- Error-tolerant radio / link layer protocols: RLPLite, PPPLite

- Collected >5,000 minutes of TCP, UDP, RLP traces in good/bad, stationary/mobile environments (download)
Optimizing GSM CS Data Protocol

- Circuit-switched data Radio Link Protocol (RLP)
  - Semi-reliable ARQ Protocol at 9.6 Kbit/s
  - Link resets after $N = 6$ number of retransmissions
- RLP and TCP interaction measurement / analysis
  - Both are reliable protocols (link and transport layers)
  - Researchers claim *competing retransmissions* problem (Q1)
- Fixed logical frame size of 30 bytes
  - 6 bytes header/checksum
  - What are effects of alternative frame sizes on throughput?
  - What effect does the choice of model have on results? (Q2)
- MultiTracer trace collection/analysis tool quickly identifies interaction and performance effects
MultiTracer Measurement Testbed

Application
Packetization
RTP
Socket Interface
TCP/UDP (Lite)
IP
PPP/PPP Lite

Multi-layer trace collection
- RLP, UDP/TCP, App
- Easy trace collection
- Rapid, graphical analysis

Application
Packetization
RTP
Socket Interface
TCP/UDP (Lite)
IP
PPP/PPP Lite

Goals MultiTracer MTA Prediction

MultiTracer

300 B/s

Plotting & Analysis

SocketDUMP
TCPdump
TCPstats
RLPDUMP

Mobile Host
Unix BSDi 3.0

GSM Network

Fixed Host
Unix BSDi 3.0

PSTN

SocketDUMP
TCPdump
TCPstats
Time-Sequence TCP Plot
MultiTracer TCP and RLP Plot
Choosing the Right Network Model

- Collect empirical packet trace: \( T = \{1,0\}^* \)
  - 1: corrupted/delayed packet, 0: correct/non-delayed packet

- Create mathematical models based on \( T \)

- Challenge: domain analysis – which model to use?

- \( T \) may be *non-stationary* (statistics vary over time)
  - Classic models don’t always work well (can’t capture variations)

- MTA, \( M^3 \) – Trace data preconditioning algorithms
  - Decompose \( T \) into stationary sub-traces & model transitions
  - Stationary sub-traces can be modeled with high-order DTMC
  - Markov-based Trace Analysis (MTA) and Modified hidden Markov Model (\( M^3 \)) tools accurately model time varying links
Creating Stationarity in Traces

- Our idea for MTA and M³: decompose $T$ into stationary sub-traces
  - Bad sub-traces $B_{1..n} = 1\{1,0\}^*0^c$, Good sub-traces $G_{1..n} = 0^*$
  - $C$ is a change-of-state constant: mean + std dev of length of $1^*$
- MTA: Model B with a DTMC, model state lengths with exponential distribution, and compute transitions between states
- M³: Similar, but models multiple states using HMM to transition

Model B with DTMC
Role of Accurate Network Modeling

- Choosing optimal RLP frame size for bulk data transfer
  - Use real Block Error Traces from MultiTracer
  - Compare against synthetic traces with same BER generated by different models (MTA, Bernoulli, Gilbert)
  - Measure throughput for different frame sizes (30, 60, ..., 1500 B)
- Optimal frame size is the one with highest throughput
  - Reduced overhead vs. increased retransmission delay
- MultiTracer analysis: errors occur in long bursts
  - Channel is either OK or very bad
  - A few long or many short packets are affected

*Burst effect not captured by classical models*
Right Model $\Rightarrow$ More Accurate Results

- **gilbert_trace** (std err 22)
  - max @ 150 bytes
  - Tput = ~1330 B/s

- **real_trace**
  - max @ 210 bytes
  - Tput = ~1290 B/s

- **bernoulli_trace** (std err 48)
  - max @ 60 bytes
  - Tput = ~1150 B/s

- **actual**:
  - 30 bytes
  - Tput = ~1080 B/s

- **mta_trace** (std err 8)
  - max @ 210 bytes
  - Tput = ~1280 B/s
Predictive, Error-Resilient Video

- Explore benefits of feedback and prediction
  - Dynamic adaptation of data rate to network conditions for adaptive, error resilient video codecs
    - H.263+ error-resilient QCIF codec (< 64 kb/s, 10-15fps)
- Link layer reliability helps with wireless errors
  - But, link layer reliability alone introduces delay ⇒ higher jitter
- Add a simple feedback algorithm
  - High and low resolution versions of video stored on server
  - Channel prediction bad state ⇒ send low resolution frame
  - Goal: switch data rates to minimize jitter effects by keeping inter-frame arrival times relatively flat
Emulation Architecture

Sender Application

Receiver Application

Sender Socket Interface

Feedback

MTA / M3 Model

WEmu / WSim

Non-ARQ / ARQ

MTA / M3 Prediction

Radio Link Emulator (RLE) (in C++)

Feedback

RTP/UDP/IP

IP Packets

Fragmentation

552 bytes

30 bytes

Radio Frames

Statistics

radio.dump

feedback.dump

UDP connection

Socket Interface

Sender Application

Receiver Application

IP Packets

Re-assembly

RTP/UDP/IP

552 bytes

Radio Frames

Goals MultiTracer MTA Prediction
Streaming Video with Predictive Feedback

- Simulation using wireless traces (4480 min):
  - ~4 min per video, bad channel (-105 to -99dB), ~ 1.5 % BLER
- Calculate $S_b$ and $S_g$ (avg bad/good subtrace lengths), and $C$
- Receiver receives radio frame:
  - In “bad subtrace”? $\Rightarrow$ send bad state feedback every $S_b$ frames
  - In “good subtrace”? $\Rightarrow$ send good state feedback every $S_g$ frames
Predictive Feedback Algorithm

**Preliminary results**
- No feedback: jitter std dev 150 ms, many 200+ ms instances
  Feedback: jitter std dev 100 ms, only two 200+ ms instances

**Next steps:** More sophisticated algorithms, full platform, RTCP feedback-based mechanism
Tapas Summary

- A better understanding of effects of multi-layer effects
- Accurate models $\Rightarrow$ Accurate simulation $\Rightarrow$ Better protocol design
- 1st cut simple socket interface model for communicating with lower protocol stack layers
- Preliminary result: Prediction enables better response time to discontinuous changes in error rate
- On-going work:
  - Trace collection: CDMA 1xRTT, GPRS, & IEEE 802.11a
Griffin: Towards an Agile, Predictive Infrastructure

Anthony D. Joseph
UC Berkeley
http://www.cs.berkeley.edu/~adj/
Dagstuhl QoS Seminar Oct 2002