A Framework for Self-Healing Home Networks

Amy Csizmar Dalal
Department of Computer Science, Carleton College

BACKGROUND

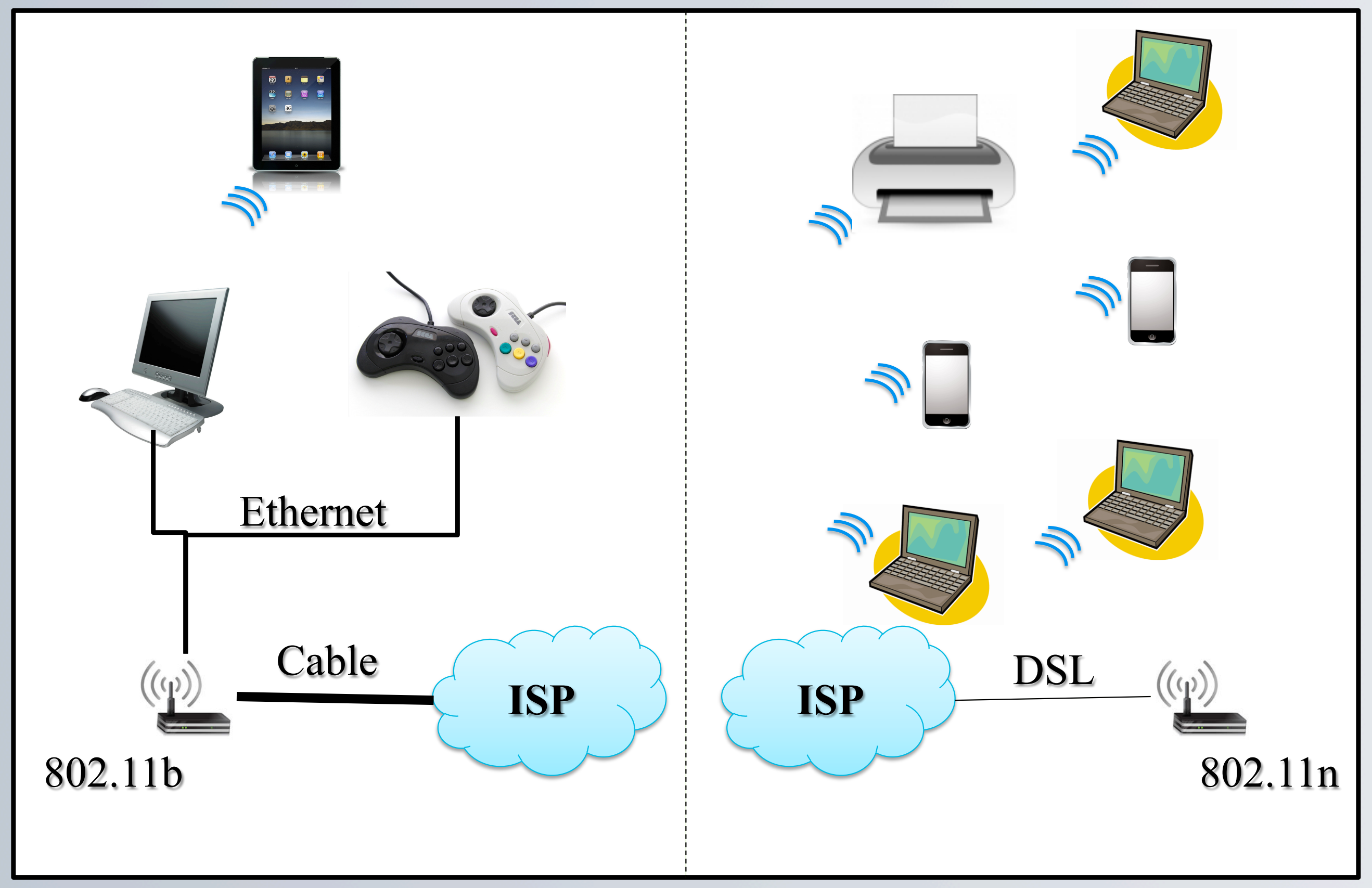
GOAL: Develop a self-healing network for the home network space that can troubleshoot, predict, and mitigate problems before they occur, with minimal user intervention or awareness.

Self-healing network

- A computer network that can **detect** existing and/or potential pathologies and **mitigate** them with minimal human intervention.
- Useful in a variety of scenarios (e.g. large and/or complex networks).

Why the home network space?

- **Heterogeneity**: in devices, topologies, connections to Internet.
- Set up and maintaned by **non-experts**.

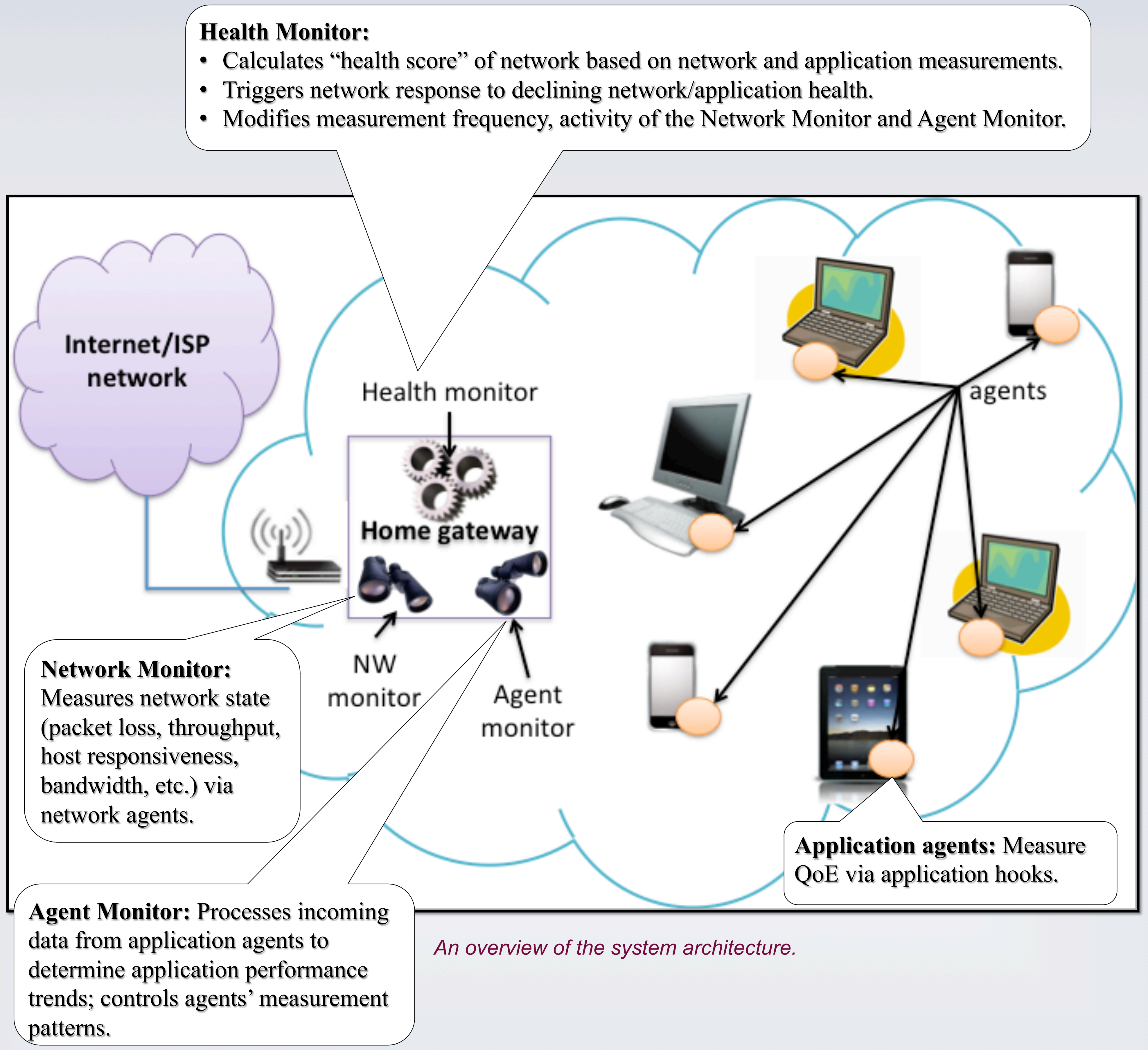


Home networks may have very different topologies, connection speeds, devices, ISPs, and application mixes.

SYSTEM REQUIREMENTS

- **Proactive, not reactive.** To maximize the user experience, the network should detect and mitigate problems before they manifest in the applications.
- **Partially decentralized.** Functionality should be spread throughout the system to avoid bottlenecks.
- **Focused on application QoE.** Applications are the closest to the end user and thus are a good proxy for the user's quality of experience (QoE).
- **Combines both network and application measurements.** Application measurements can predict end user QoE, while network measurements are early indicators of future application issues.
- **Minimal to no user intervention required.** Expert knowledge is not a prerequisite for a highly-functioning network.

SYSTEM ARCHITECTURE



Health Monitor:

- Calculates "health score" of network based on network and application measurements.
- Triggers network response to declining network/application health.
- Modifies measurement frequency, activity of the Network Monitor and Agent Monitor.

Network Monitor: Measures network state (packet loss, throughput, host responsiveness, bandwidth, etc.) via network agents.

Agent Monitor: Processes incoming data from application agents to determine application performance trends; controls agents' measurement patterns.

Application agents: Measure QoE via application hooks.

An overview of the system architecture.

Currently deployed agents

<u>Application/host-level</u>

- Devices on/active
- Active applications
- Download throughput
- Video QoE (from previous work)

<u>Network-level</u>

- Available bandwidth (using Iperf)
- Packet loss
- Delay
- Jitter

CHALLENGES

Timing	Frequency of measurements shouldn't overwhelm the system, yet still detect pathologies. Determine experimentally and modify as conditions degrade or improve.
Data freshness	System utilizes historical data about network conditions, but should favor more recent measurements. Ensure that training set is updated regularly to include recent data.
Privacy	Sharing sensitive data outside the network, e.g. with the home's ISP, can improve performance by demonstrating commonalities among different topologies. Anonymize data before sharing.
Third-party cooperation	Sharing data outside the network may improve performance but may also expose breach of service contracts. Care must be taken to protect the interests of all parties.

EXAMPLE SCENARIOS

Status quo (normal operation)

- Network Monitor commissions periodic measurements from network agents, analyzes data from agents.
- Agent Monitor determines which hosts are up and which applications are running.
- Agent Monitor commissions periodic measurements from appropriate application agents.
- Agents collect application measurements, send data to Agent Monitor.
- Agent Monitor analyzes data from agents.
- Health Monitor analyzes network and application measurement results.
- Health Monitor learns "normal state" of network.
- Health Monitor learns and modifies appropriate measurement frequency.

Bandwidth hog?

- Agents measure bandwidth usage per client/per application.
- Health Monitor applies heuristics to reallocate bandwidth.
- Example: prioritize Skype session in office over gaming session in den.

Sudden outage

- Network Monitor detects increase in delays and packet losses.
- Network Monitor takes targeted measurements to help determine cause.
- Health Monitor applies heuristic (e.g. locate alternate gaming servers).

ONGOING WORK

- Collecting performance data from our currently-deployed network and application agents on a home network testbed.
- Modifying our data mining algorithms (k-nearest neighbors with dynamic time warping as distance metric) to determine connections between application performance and network performance.
- Development, testing, and deployment of the Health Monitor, Network Monitor, and Agent Monitor.
- Development of use cases to determine appropriate heuristics and actions that the Health Monitor can implement (see Example Scenarios, above).

CONTACT INFO

Amy Csizmar Dalal
Associate Professor of Computer Science, Carleton College
Northfield MN, USA

Email: adalal@carleton.edu
URL: http://www.cs.carleton.edu/faculty/adalal
Twitter: @drcsiz

***Acknowledgements**: Thanks to Yanshan Guo ('16) and Emily Shack ('15) for developing and deploying the network agents and some of the application agents, and for collecting and analyzing preliminary data.*