TECHNICAL ARTICLE

SIMULATION 77:3-4,

ISSN 0037-5497/01
Printed in the United States of America

Visualization of Mobile Network Simulations

T.A. Dahlberg and K.R. Subramanian
Computer Science Department
University of North Carolina at Charlotte
9201 University City Boulevard
Charlotte, NC 28223, USA
{tdahlber, krs}@uncc.edu

The use of adaptive techniques in mobile networks permits scalable resource allocation policies to meet varying demand as well as Quality of Service (QoS) performance objectives. As these algorithms operate at multiple layers of the communications architecture, evaluation of such techniques must take into account a variety of scenarios, which are in turn parameterized by a large number of variables. The need to monitor algorithm behavior in real time results in a data explosion. In this work, we propose new real-time metrics to characterize and identify the critical states of a mobile network in the wake of channel failures, congestion, signal degradation, etc. We use these metrics to define a survivability index, a measure of mobile network performance in the wake of failures. We demonstrate the effectiveness of information visualization techniques in understanding the complex spatial and temporal relationships between performance and cost metrics that influence adaptive algorithms. Our visualization system is highly scalable and interactive, permitting multiple algorithms to be simultaneously evaluated. We demonstrate applications to network monitoring and to the design and evaluation of adaptive admission control algorithms.

Keywords: Mobile networks, visualization

1. Introduction

A wireless access network is characterized by bursts of demand from mobile users with diverse service requirements, attempting to gain access to wireless links with varying signal quality. Allocation of resources to meet varying demand is constrained by the limited frequency spectrum. Hence, current research on mobile networks is focusing on the development of adaptive techniques to scale resource allocation policies to meet Quality of Service (QoS) performance objectives [1, 2, 3, 4, 5, 6]. Furthermore, protocols operating at higher layers (in the communications architecture) cannot be completely shielded from the uncertainties of the unreliable, mobile physical environment. Therefore, multi-layer adaptive techniques are being pursued to coordinate the resource management task among protocols at multiple layers within the communications architecture [7, 8]. Finally, due to the complexities inherent in modeling a heterogeneous mobile network, simulation has become a primary method for performance analysis of mobile network protocols [4, 5, 8, 9, 10, 11, 12].

The basic operation of an adaptive algorithm is twofold: (1) to monitor the system to determine the system state (e.g., failure, congestion, burst activity), and (2) to invoke a particular adaptation policy that is

best suited for that state. In general, greater emphasis has been placed on adaptation policies. However network monitoring for system state identification is crucial for distinguishing "real problems which must be addressed" and "burst activity that should be ignored." Current approaches to evaluating adaptive algorithms have given insufficient attention to understanding the sensitivity of algorithms to make this distinction over a wide variety of simulation scenarios. As will be detailed in Section 6, a considerable number of variables and parameters influence the adaptive resource management policies/algorithms (e.g., offered load, threshold values). The need to simultaneously comprehend this large problem space of variables both spatially and temporally for designing adaptive algorithms results in a data explosion, and we turn to modern information visualization techniques to abstract the essence of their complex relationships. We describe such a system and its application to both network monitoring as well as adaptive admission control. Finally, our own earlier work on survivability analysis has shown the need to consider performance during the transient period immediately following a failure, during the steady state failure period, and following recovery [13]. This further justifies the need for data visualization to be an integral part of the analysis.

An overall goal of our work is development of adaptive resource management policies for radio-level survivability of cellular mobile networks. Towards this goal, the primary objective of this study is to use visualization techniques to characterize system states of mobile network simulations in terms of real-time metrics. We introduce new real-time metrics that have been found to be the most useful for state characterization and identification. We demonstrate the essential insight provided by the visual animations in evaluating these metrics. We define a survivability index (SI) for comparative survivability analysis of resource management policies. We describe how state characterization is used to enable these policies to adapt to current network conditions. Our examples include description of our adaptive admission control algorithms and channel access algorithms. Our results demonstrate that dynamic visualizations of system variables and interactions among them are fundamental to leveraging the user's intuition and domain expertise to help explore (and reduce) a large search space of these variables.

2. Background

An ideal model of the radio-level layer of a wireless access network is shown in Figure 1. A cell refers to a hexagonal area surrounding a basestation tower. Each basestation has a group of wireless channels for use in communicating with a mobile user in its cell. A mobile user receives a wireless signal from the closest basestation. The user makes a new-connection request

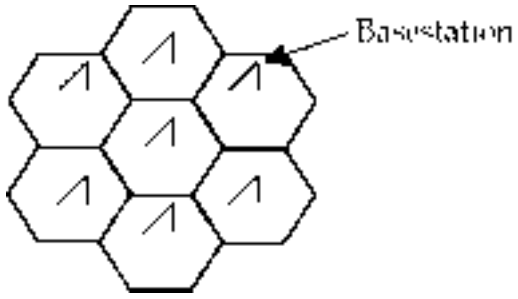

Figure 1. Ideal cellsite architecture

to the closest basestation upon placing (or receiving) a new call. As the user moves to a new cell, the user must relinquish its old channel and obtain a new one from the target basestation via a handover request.

Radio-level resource management policies include admission control to limit the number of new calls/connections allowed into the system, and channel access control to schedule use of wireless channels by ongoing calls/connections [14]. Resource management policies are often implemented as distributed protocols executed at each basestation to manage local resources (channels). However, due to user mobility, spatial relationships among neighboring cells significantly impact protocol performance.

Survivable Resource Management refers to the degree to which such protocols can maintain performance standards in the wake of failure and overload conditions. Due to the novelty of wireless devices, very little work has yet been done on mobile network survivability. However, as mass use of mobile networks increases, users will demand the same service guarantees delivered by current wired networks. Comparative analysis of competing survivability strategies requires a definition of a *Survivability Index* (SI) as a measurable metric. The review of survivability metrics, as described in [29], reveals the need for additional cost and performance metrics for survivability analysis. Additional challenges are the development of techniques to measure these metrics and the definition of how metrics will be used to adapt network policies in real time.

For instance, the simulation analysis in [8], which studies admission control and channel access control for mobile networks, uses a quality metric that is a function of 14 variables. Results are presented as 2D graphs, each graph point representing the time-wise (over 1 simulated hour) and system-wide (over all cells) average of the quality metric for one value of offered load. The spatial (between cells) and temporal (critical times) relationships of the proposed policies/algorithms are not considered at all. Also not considered is an understanding of how the 14 variables contribute to the quality index.

In our own preliminary studies [15, 16], we experimented with two Adaptive Admission Control

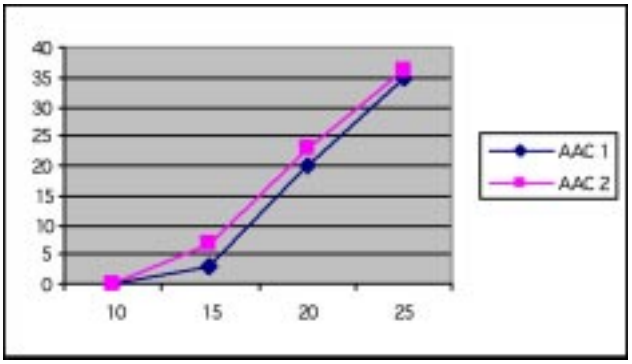

Figure 2. Averaged Blocking Rates versus Offered Load

(AAC) protocols, called AAC 1 and AAC 2. We defined the "percentage of handover requests that are denied," called handover blocking rate (*hbr*), as a performance metric. 2D plots of simulation results, as shown in Figure 2, do not clearly explain the performance differences in the two approaches. To do this, it was necessary to look at these metrics in real-time for each cell of the network. As this immediately results in the generation of a large amount of data (determined by the number of cells, metrics, the simulation period and the sampling rate), we turned to visualization techniques, and in particular, multi-variate visualization schemes. Multi-variate visualization techniques are relevant in this domain, since we have a large number of variables and the challenge is to understand their relationships and interactions spatially and temporally.

The earliest visualization schemes for multivariate data were projection-based, such as scatterplot matrices [17], dimension stacking [18], Worlds within Worlds [19] and HyperSlice [20]. These techniques look at a subset of the space (usually 2 or 3 dimensions) at a time, some support linking variables across these projected views [21] and some require an ordering of the dimensions. The major disadvantage of projection schemes are their inability to scale with the number of dimensions and the needed ordering of variables in some schemes, which prevents all dimensions from being treated uniformly. A second class of techniques based on displaying multi-variate point data through simultaneous multiple views, and linked together by line segments connecting the views, include the window wiring diagrams of rooms [22] and *parallel coordinates* [23, 24]. Parallel coordinates, in particular have received considerable attention. It maps multidimensional data into 2D plots, treating all dimensions uniformly, by plotting *n*-dimensional points as polyline segments through the *N* axes, all of which are parallel to each other. Finally, [25] describes XmdvTool, a system that integrates some of the most important multivariate visualization techniques.

Fundamental to any of these techniques is *brushing*, an interactive capability to look at subsets of the data. This allows users to highlight, analyze or focus on small or manageable portions of their data. Brushing may be screen based, using input devices to select data elements, or *indirect* (data driven), such as manipulation of multiple sliders to specify containment criteria. A third form of brushing is *structure-based* [26]; this could involve specification of level of detail, data cluster size, etc., and is appropriate for very large datasets. Brushing is a key capability that scales with data size and makes *drill down/roll up* operations very efficient.

3. Modeling and Simulation Framework

Our system model is illustrated in Figure 3 (details in [14]). Rectangles represent static models that are fixed throughout a simulation run, while ovals represent dynamic models that define time-varying properties of a simulation.

The *geographical data* set is a two-dimensional array of grid points, as shown in Figure 4, representing the mobile user's physical environment (e.g., terrain properties, transportation system, radio signal path loss model). The data set used in this study represents a region of North Carolina.

The *cellsite architecture* describes the placement of basestation towers, basestation coverage area, and the wireless capacity of a basestation, in the context of the physical environment. As shown in Figure 4, our cellsite architecture represents the use of two reuse partitions to provide two groups of channels, with longer and shorter coverage ranges at each basestation. It is assumed that a user at a grid point experiences the physical environment defined by the geographical data set at that point and receives a usable wireless signal from the basestation tower in the center of any cell that covers this grid point.

We have developed an object-oriented discrete-event simulator in C++ to implement the system model. During simulation, calls arrive into the system according to a *teletraffic model*, and are initially located at a grid point according to location distributions within the geographical data set. Calls move among grid points according to a *mobility model* demanding resources (channels) from overlapping cells in the cellsite architecture. During simulation, *fault models* inject failures into the cellsite or degraded conditions into the geographical data.

We have implemented various *teletraffic models*, Figure 3, to define calling patterns with respect to call arrival and holding times. One can choose among various loads (in the same or different simulation runs). A nominal load represents users' demand for channels equivalent to that for which the system was optimally designed.

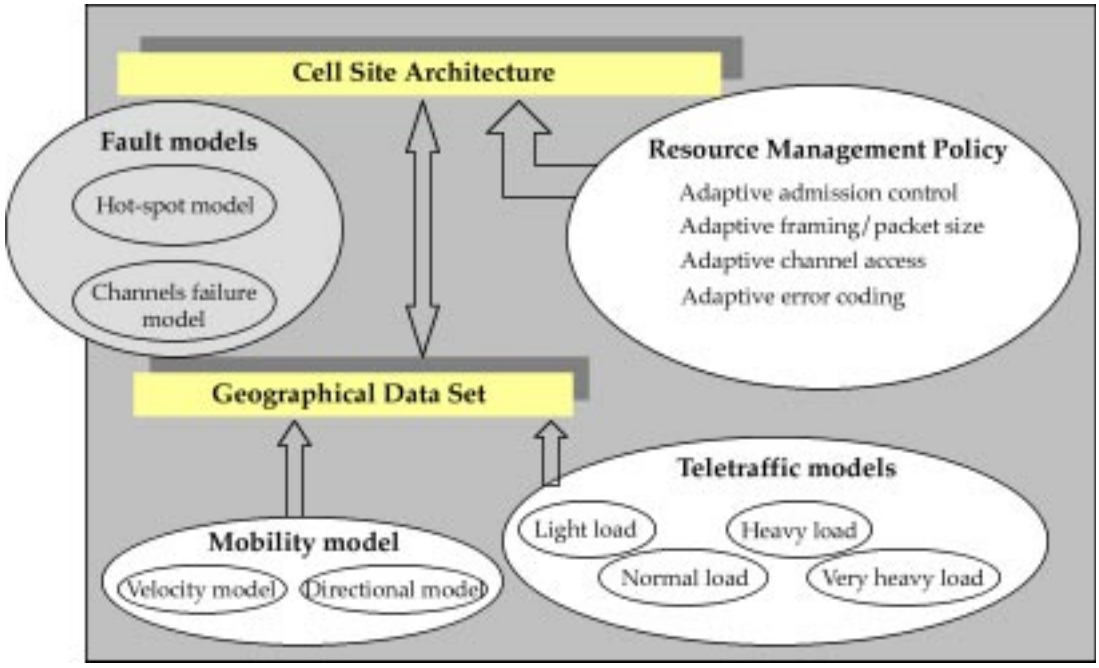

Figure 3. Modeling framework

Our *mobility model* defines characteristics of how a user moves among the geographical grid points, Figure 3.

Our results include use of two *fault models,* Figure 3. The channel failure model fails 75% of the channels in specified cells for any time-period (models partial hardware/software failure). The *hot-spot model* increases offered load significantly beyond the nominal value in specified cells for a specified time period (models user congestion in the physical environment, such as that caused by traffic surrounding a highway accident, or people leaving a stadium event).

Example multilayer resource management policies are also shown in Figure 3. Thus far, we have implemented *Adaptive Admission Control* and *adaptive Channel Access* (CA) control algorithms [14, 15].

Ideally, a simulation analysis of these algorithms would explore all simulation scenarios in terms of the various models described. However, each model contains multiple factors (variables that affect performance results) that can take on multiple values. A full factorial analysis [27] is not possible, and an approach to design a fractional factorial set of experiments is not clear. Thus, our approach in this work is to use application domain expertise to design real-time metrics for better understanding of adaptive resource allocation policies. We turn to information visualization techniques to assist us in viewing multiple real-time metrics simultaneously within a highly interactive environment, in order to facilitate the analysis of such policies, as well as the design of new algorithms.

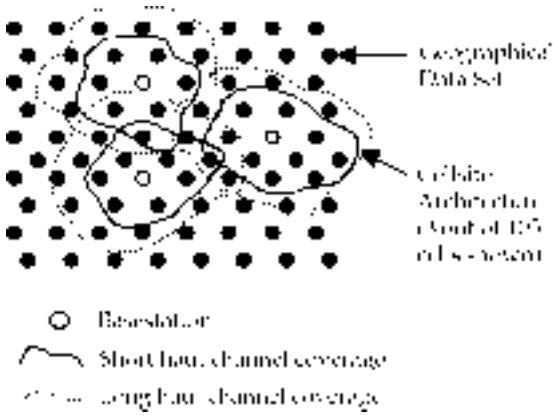

Figure 4. Cellsite Architecture

4. Visualization Framework

All visualizations have been constructed using the Visualization Toolkit (VTK) [28]. We are currently using three types of visualizations,

1. *color mapped planes,* where the metric values are mapped into a set of colors (we use a rainbow color map, from blue to red),
2. *height fields*, where the height corresponds to a metric value at a specific cell, and
3. *parallel coordinates*, where each metric is represented as a separate vertical axis.

The cell site network used in our experiments consists of 105 cells arranged in a square grid. A *structured grid* (a grid that is topologically a multidimensional array, but its lattice points are not uniformly spaced) is used to represent each metric. The geometry of the grid can be displayed either as solid colors

Table 1. Characterization of system states

System states	Characterization
Normal	Little activity, arbitrary, sporadic increases in all metrics.
Degraded	Persistent spatial & temporal increase in *oc*, *cu* and in *ct*, and Recurring spatial & temporal spikes in *sum*, *hbr*, and *nbr*
Transient failure	Spatial & temporal increase in *dcr*.
Steady-state failure	Persistent spatial & temporal decrease in *cu* and in *ct*, and Recurring spatial & temporal spikes in *sum*, *hbr*, and *nbr*
Recovery	Recurring spatial & temporal dips in *oc*, *cu*, and *ct*.

or *wireframe* (which is useful in focusing on specific cells) format; also, the hexagonal lattice structure can be overlaid for better perspective of the cell site architecture.

The visualization system can look at all 16 metrics within a single run (Figure 5-7), or be configured to look at multiple runs with specified metrics (Figure 9 shows 4 runs with 4 metrics, with each run in a separate row). The ability to look at multiple runs is especially useful in designing adaptation policies/algorithms in terms of the metrics/parameters. Once the simulation data is input to the visualization system, the entire run can be animated and controlled using VCR style controls (play, stop, forward step, backward step, etc.). In the multi-run mode, the VCR controls will affect all simulation runs, again permitting direct comparison between simulation runs, parameterized by adaptation policies, algorithms or simulation parameters.

A number of analysis tools are in use and in development. For instance, it is possible to compute statistics of a particular run (peak and average values of the metrics over the entire run, variances, etc.). Any metric value for a particular cell can be queried when it is necessary to focus on a particular cell. The visualization also allows tracking a small group of cells, for instance, cells in the vicinity of a failure to focus on spatial relationships.

5. Real-time Metrics

The objective of network monitoring is to identify the current state of the system so that a resource adaptation policy can be invoked that best maintains survivability for that state. For this study, five system states, also called system operating modes, are characterized—*normal*, *degraded*, *transient failure*, *steady-state failure*, and *recovery*. Table 1 summarizes how metrics were used to characterize and identify the systems states as the offered load to the system varied among *light*, *nominal*, *heavy*, and *very heavy* offered load values.

A *normal mode* implies that no explicit failure has occurred. A degraded mode refers to a time during which an explicit system failure may not be present, but system performance is diminished due to environmental reasons (e.g., increased demand in one cell due to highway congestion resulting from a traffic accident or due to an event, such as people leaving a football stadium). We generate a degraded mode using our hot-spot fault model, described in Section 3. A *transient failure mode* refers to the time-period immediately following a failure when a system may be somewhat unstable while trying to maintain ongoing connections affected by the failure. *Steady-state failure mode* refers to the time period in which a fault still persists, but the system has reached a somewhat stable, though degraded mode of operation. A *recovery mode* is the time period following failure, during which a system attempts to reinstate normal usage patterns. These latter three modes are generated using our channel failure fault model. In addition, we will define our survivability index in terms of the survivability objective in Table 2, in a later section.

The real-time metrics found to be most useful for state characterization are defined next and explained more fully in Section 6.

Metric Definitions

Each metric is measured as a sliding window variable. The metric is evaluated at a rate determined by a *sampling_rate* parameter, by calculating raw data over a time period specified by a *time_window_size* parameter. The summation symbols in the equation that fol-

Table 2. Survivability objectives

Survivability Objectives
○ Handover blocking rate ≤ 1%
○ New connection blocking rate ≤ 2%
○ Minimize overall blocking rate
○ Handover blocking rate equal to half of new connection blocking rate
○ Maximize carried traffic

lows imply that data is summed over a period equal to the *time_window_size*.

Handover blocking rate, *hbr*, is the percentage of forced handover requests that are denied. A handover request is considered to be forced, if denial of a channel will result in termination of the call.

$$hbr = \frac{\sum forced\_handovers\_denied}{\sum forced\_handover\_requests}, 0 \le hbr \le 1$$

The derivative of *hbr* represents the rate of change in *hbr* between successive samples of the metric.

$$hbr' = \frac{hbr_{time\_t} \pm hbr_{time\_t\pm1}}{window\_size},\ hbr' \in \Re$$

Similarly, new connection blocking rate, *nbr*, and its derivative represent the percentage of new connection requests that are denied, and the rate of change of this value.

$$nbr = \frac{\sum new\_connections\_denied}{\sum new\_connection\_requests}, 0 \le nbr \le 1$$

$$nbr' = \frac{nbr_{time\_t} \pm nbr_{time\_t\pm1}}{window\_size},\ nbr' \in \Re$$

Because of the nature of voice traffic, which currently dominates our teletraffic model, the *hbr* and *nbr* values are bursty in nature. Therefore, we are also interested in looking at the real-time sum of these values, as measured with the sum metric. The assumption is that a significant increase in both metrics at the same temporal and spatial location more likely conveys a problem rather than a random burst.
The blocking ratio metric, *br*, is an indicator of how well the system is managing to meet the survivability objective of keeping a 0.5 ratio of handover blocking rate to new connection blocking rate. A $br = 0$ is most desirable as it indicates that either the *hbr*/*nbr* ratio is 0.5, or that *hbr* and *nbr* are within the desired range, less than 1% and less than 2%, respectively.

$$br = \begin{cases} 0, & if\ hbr \le 0.01\ AND\ nbr \le 0.02 \\ 0.5, & if\ \frac{hbr}{nbr} \ge 1.0 \\ \frac{hbr}{nbr} \pm 0.5, & otherwise \end{cases},$$

$$-0.5 \le br \le +0.5$$

The dropped connections rate metric, *dcr*, measures the number of connections that are prematurely terminated due to something other than a failed handover attempt.

$$dcr = \frac{\sum dropped\_connections}{output\_size * (1 + \sum channels\_used)},\ dcr \ge 0$$

For the failure models that we have used, a connection is terminated if the channel being used fails. We experimented with various strategies to determine how to view the number of dropped connections, e.g., as an absolute value, or with respect to time or offered load. We determined that a ratio of dropped_connections to offered load provided the most meaningful values. As such, the denominator of *dcr* represents a scaled estimate of offered load. That is, the denominator is an estimate of the offered load in Erlangs divided by 3600. (1 Erlang is equal to usage of one channel for 1 hour, which is 3600 seconds.) The output_size parameter is the rate at which raw data is sampled. We estimate channel usage by assuming that if a channel is in use at the time of sampling, then the channel has been in use since the prior sample.

The offered connections metric, *oc*, measures the rate of new connection arrivals with respect to time. Later we will demonstrate how *oc* is very useful in illustrating the behavior of our distributed adaptive CA algorithm. We experimented with measuring offered load in units of Erlangs (as we do for carried traffic). Offered connections measures arrival of new connections without regard to connection bandwidth or length, while *offered load* considers bandwidth and length. However, only *oc* proved useful for this current study.

$$oc = \frac{\sum new\_connection\_requests}{window\_size},\ oc \ge 0$$

The carried traffic metric, *ct*, is an estimate of the carried load in units of Erlangs. Note that offered load is capacity demanded by users, while *carried load* or traffic is capacity actually delivered to users by a system.

$$ct = \frac{\left(\sum channels\_used\right) * output\_size}{3600\ seconds/hour},\ ct \ge 0$$

The channel utilization metric, *cu*, is an estimate of the percentage of channels used with respect to time.

$$cu = \frac{\left(\sum channels\_used\right) * output\_size}{window\_size},$$

$$0 \le ct \le maximum\ \text{channels}$$

The handover activity rate metric, *har*, represents the number of successful forced and voluntary (described later) handovers with respect to time.

$$har = \frac{\sum number\_of\_handovers}{window\_size},\ har \ge 0$$

We define a survivability index, *SI*, as a measure of how well the system meets the survivability objectives specified in Table 1. Most current wireless network providers aim to maintain handover and blocking rates below 1 and 2 percent. From a user's perspective, terminating an ongoing connection is perceived to be a worse problem than blocking of a new connection attempt. However, keeping an extremely low *hbr* is only achievable by allowing an extremely high *nbr*. As such, since *hbr* and *nbr* blocking rates will exceed desired values during failure or con-

gestion conditions, the objective is to maintain a 0.5 *hbr*/*nbr* ratio. Lastly, since carried traffic represents network provider revenue, an objective is to maximize this value.

Based on these objectives, we define the following survivability index, where optimum survivability is achieved when $SI = 0$.

$$SI = h_1\left[\frac{(k_1A + k_2B + k_3C)}{\sum_{i=1}^{i=3} k_i}\right],\ 0 \le SI \le 1,$$

$$A = 2 * |br|,\ 0 \le A \le 1,$$

$$B = \frac{hbr + nbr}{2},\ 0 \le B \le 1,$$

$$C = \frac{|cu \pm maximum\_channels|}{maximum\_channels},\ 0 \le C \le 1.$$

6. Experiments

In Section 6.1, we demonstrate how the visualizations proved essential for state characterization of mobile network simulations that incorporate all of the models described in Section 3. (Initially, we use a resource management model that includes our adaptive channel access scheme and no admission control.) The visual appearance of each metric during normal mode is described. An explanation of the metrics that characterize abnormal modes is then given. The effects of teletraffic loading conditions are discussed throughout, and summarized in a separate section, followed by a section discussing the effects of varying sampling rate and time window size. In Section 6.2, we discuss potential survivability strategies that could be invoked during each state and give specific examples for adaptive admission control and channel access control.

6.1 State Characterization

6.1.1 Normal Mode

Snapshots of normal mode are shown in Figure 5. (Refer to Figure 10 for the color map.) The optimal value for blocking rate metrics *hbr*, *nbr*, and *sum* is zero—visualized by a flat dark blue color. During simulation of a lightly loaded system, *hbr*, *nbr*, and *sum* will primarily remain flat and dark blue, with occasional lighter blue bumps indicating normal bursty traffic. This agrees with our intuition since even in a lightly loaded system some, blocking will occur as random call arrivals may occasionally cluster in the same cell, causing resources to be depleted. The *sum* metric helps to filter out some randomness, since in the absence of faults, it is less likely that *hbr* and *nbr* will burst at the same spatial and temporal instant. During simulation of a heavily loaded system, all three metrics become increasingly bursty. The animation displays lighter blue bumps moving among all cells, with an occasional larger, more orange peak in a particular cell.

Under light loading, the blocking rate derivative metrics, *hbr'* and *nbr'*, are mostly flat green with an occasional color spot. These metrics can take on positive or negative values. A zero value indicates no change (which corresponds to green on a blue to red rainbow color scale). As loading increases and *hbr* and *nbr* become more variable, *hbr'* and *nbr'* become slightly more colorful, indicating sharp rises and dips in *hbr* and *nbr* values.

The aim of *br* is to illustrate the degree to which the network meets the survivability objective of maintaining a 0.5 ratio for *hbr*/*nbr*. This ratio is managed by admission control. *br* varies between [-0.5,+0.5]. Therefore, a *br* that is flat green indicates that the blocking ratio objective is being met in real-time. A spike up towards red indicates that *hbr* is too high relative to *nbr* (i.e., our performance has degraded). Conversely, a downward spike towards blue indicates that *nbr* is relatively too high (i.e., our cost has degraded). Later, we describe how our adaptive admission control algorithms managed this cost/performance tradeoff.

The *dcr* metric measures dropped calls and varies from zero only during transient failure mode. The *oc* metric displays new connections arriving to cells and the *har* metric displays handovers occurring within each cell. The *cu* metric displays channel utilization at each cell. *oc*, *har*, *cu*, and *ct* are useful for illustrating the behavior of our load-based adaptive channel access (CA) protocol, as described in Section 6.2. *ct* is similar to *oc*, which indicates that for normal operation, carried traffic does not vary greatly from offered load. The values of *oc*, *har*, *cu*, and *ct* increase with offered load. The visual appearance of the metrics maintains a spatial uniform look as the color of each metric moves towards red.

6.1.2 Degraded Mode

Snapshots of degraded mode caused by a 20 minute hot-spot in a cell in the very center of the cellsite grid are given in Figure 6. The presence of the hot-spot is most clearly characterized by spatial and temporal changes in metrics *oc*, *cu*, *ct*, *sum*, *hbr*, and *nbr* (Table 1). A spatial change implies that the value of the metric at one cell varies significantly from the value of the same metric at neighboring cells. A temporal change implies that the value of the metric at one cell varies significantly as compared to a long-term average of the value of the metric at the same cell over a recent-past time interval. The visualization animations between the times of each snapshot in Figure 6 show that the spatial and temporal increase in *oc*, *cu*, and *ct* are persistent, meaning their values remain at an increased level for a relatively long time period. The animations between the two snapshots also show that increases in *sum*, *hbr*, and *nbr*, are recurring, meaning

that their values oscillate from a noticeable high to a lower value within a relatively small timeframe. (Note that this is not the same as an arbitrary random spike.)

6.1.3 Transient Failure Mode

Snapshots of transient failure mode caused by a 20-minute failure of channels in a cell in the very center of the cellsite grid are given in Figure 7. The onset of the failure is identified by a sharp temporal increase in the *dcr* metric. We could also consider the spatial value of *dcr* to help identify the location of the failure within the mobile network, e.g., an increase in *dcr* in a single cell could indicate a problem at the cell basestation, while an increase within multiple adjacent cells could indicate a problem at the basestation controller or mobile switching center (which connect multiple basestation to the fixed network).

6.1.4 Steady-state Failure Mode

As noted in Table 1, this mode is characterized by persistent spatial and temporal decreases in *cu* and *ct*, and recurring spatial and temporal spikes in sum, *hbr*, and *nbr*. (Snapshot not shown.) The "look" of sum, *hbr*, and *nbr* for steady-state failure is similar to that for degraded mode. However, these two modes are distinguished by whether *ct* and *cu* increase or decrease. The former implies that congestion results from an unexpected increase in user demand, while the latter implies that congestion results from an unexpected reduction in cell resources. It is important to make this distinction since one might invoke different survivability strategies for each mode.

6.1.5 Recovery Mode

As listed in Table 1, this mode is characterized by recurring spatial and temporal rises and dips in *oc*, *cu*, and *ct* metrics. The *sum*, *hbr*, and *nbr* metrics gradually return to their normal look. Actually, we expected that all of the unusual looking metrics would gradually return to their normal look during recovery. However, *cu* and *ct* exhibited a series of high to low oscillations that seemed to be exacerbated under heavier loading conditions. This very surprising discovery required further investigation, which we elaborate upon in Section 6.2.

6.1.6 Effects of Offered Load

The visual appearance and the measured values of some of the real-time metrics vary significantly with offered load. This affects state characterization across varying loads. For example, bursty metrics such as *hbr* and *nbr* become more bursty under increased loads. During very heavy loading, peak values of these metrics increase in normal cells, and decrease in failed cells as compared to peak values during light loading. The reason for the decrease is that with greater offered load, the impact of each blocked call is diminished (i.e., the denominator of hbr and nbr equations becomes larger, but the numerator remains the same). For our experiments, *oc*, *cu*, and *ct* remain basically unchanged over variations in offered load. As such, we conclude that the characterization of each of the five modes in Table 1 holds across all four offered loads considered. However, our continuing efforts are to measure each metric as a function of offered load to provide more robust metrics.

6.1.7 Effects of Metric Measurement

We experimented with various values of sampling rate and time window size to determine the best frequency and time period over which to measure real-time metrics. A larger time window size causes the visual appearance of bursty metrics to become smoother and diminishes the impact of peaks, especially those that indicate abnormal conditions (e.g., *dcr*). A smaller time window size has the opposite effect. If the time window size is too small, it is impossible to distinguish between bursty activity and failure. If it is too large, failure is masked. For a given time window size, variations in sampling rate only affect how quickly a failure is detected (i.e., resolution of failure time is proportional to sample rate). In general, a large range of values is acceptable and does not dramatically alter the look and usefulness of each metric.

6.2 Survivability Strategies

We next consider the use of state characterization for survivable adaptive resource management in two ways. One, our adaptive algorithms use state characterization for network monitoring. Two, the visualizations proved to be an essential aid for design/debug of our adaptive algorithms. The visual animations enabled us to quickly understand algorithm behavior and dramatically reduce the time taken to evaluate various algorithm design and parameter choices. We discuss examples of our adaptive channel access and adaptive admission control protocols. As previously explained, our survivability objective is given in Table 2.

6.2.1 Adaptive Channel Access Example

Our system model (Section 3) is such that a mobile user moves among grid points in the geographical data set. A mobile user in need of a channel will consider the channel utilization, *cu*, at the basestation of each cell that overlaps the user's grid point. The mobile will request a channel from the basestation with the lowest *cu* (assumed to be most lightly loaded). In addition, mobile users periodically compare *cu* values at all accessible cells, and may request a voluntary handover if one of these cells has significantly lower *cu*. (Details are in [14, 15].) Therefore, the normal

mode appearance of *oc*, *har*, *cu* and *ct*, as smallish bumps, with a uniform spatial distribution, confirms that our load-based CA algorithm is indeed evenly distributing load throughout the network.

In degraded mode, we detect an expected increase in *oc*, *cu*, and *ct* in the faulty cell, and an increase in these metrics in cells surrounding the faulty cell. This further confirms operation of our CA algorithm. As a cell becomes overloaded, it will attempt to offload calls in overlap regions to its neighboring cells. During failure mode, *cu* and *ct* decrease, since less traffic is being carried by the cell that has lost most of its channels (a failed channel cannot be considered to be "in use").

The most surprising discovery as to the behavior of our CA algorithm came from further investigation of recovery mode behavior of *oc*, *cu*, and *ct*. The oscillation of these metrics during recovery did not match our intuition. As soon as all failed channels are restored in a cell, the cell immediately has lower *cu* values as compared to its neighbors. Therefore, we expected *oc* to continue increasing in the restored cell until *cu* and *ct* in the restored cell matched those values in its neighboring cells. Then we expected *oc*, *cu*, and *ct* to return to their normal appearances. Instead, we found that each of these metrics would oscillate (to unexpectedly low values) for up to 30 minutes beyond the point of restoration.

A detailed analysis uncovered that a sudden fluctuation in *cu* causes our CA algorithm to generate unnecessary voluntary handover activity. Immediately following recovery (Figure 8, far left), the failed cell's *cu* returns to green (lower loading) as it picks up load with newly restored channels. However, soon after (Figure 8, center left), the cell drops to deep blue, while its neighbors turn red. This indicates that the restored cell became overloaded, as compared to its neighbors, and off-loaded calls using voluntary handover. However, too many calls were off-loaded, leaving the cell under-utilized. Thus, the cell returns to green again (Figure 8c, center right) and repeats the cycle (Figure 8, far right). We attempted to fix this oscillation through our restoration model. Rather than restore all failed cells simultaneously, we experimented with restoring channels over a period of time (up to 30 minutes). A longer restoration period slowed the oscillation cycle, but did not completely solve the problem. As such, we are looking at ways to further modify our CA algorithm to better adapt to recovery mode.

6.2.2 Adaptive Admission Control Example

Our basic approach to adaptive admission control (AAC) is to vary the size of a guard band maintained at each cell. The guard band specifies a percentage of channels, called δ, that should be set-aside for handover requests and not allowed for use by new-connection requests. This enables one to manage a tradeoff between *hbr* and *nbr* (details are in [15, 16]). There are a great many ways to implement AAC. Each approach consists of monitoring a real-time metric, comparing its value to upper/lower threshold values, and then increasing or decreasing δ based on this comparison. We experimented with 4 AAC algorithms, called AAC1, AAC2, AAC4, and AAC5. These monitor *hbr*, *hbr'*, *br*, and *br'*, respectively. Figure 9 shows metrics *hbr*, *hbr'*, and *br*, along with the real-time value of δ. (AAC 1, 2, 4, 5 occupy rows 1, 2, 3, 4, respectively of the snapshot.) The performance of each algorithm varies significantly depending on the choice of upper and lower threshold values. The responsiveness of each algorithm to abnormal conditions also varies greatly. Looking at averaged simulation results revealed differences in performance, but gave little insight into the cause of such differences.

The visual animations provided quick, powerful insight. For example, the AAC2 and AAC5 that use derivative values are extremely sensitive to minor changes in threshold values. In many instances the value of d would increase in response to burst activity in normal mode. The value of δ would then remain "stuck" at an unnecessarily high value for a long time. Also, these algorithms are less sensitive to degraded mode, which is characterized by a more gradual increase in metric values (rather than quick rate of change). Thus, in general, using derivatives caused too many false reactions in normal mode, and non-responsiveness in fault modes.

The algorithms that used actual values, AAC1 and AAC4, were more responsive, meaning that the value of δ was increased when needed and decreased when not needed. However, the optimum setting of threshold values is different for different values of offered load. Also, at higher loads, metric spatial relationships are more important to help distinguish faulty states. With the help of the visualization, our study on AAC algorithms continues.

6.2.3 Survivability Index

We are also studying the development of a survivability index (SI). The SI function used is described in Section 5. As shown in Figure 10, the visualization enables us to evaluate the real-time contribution of individual components (e.g., A, B and C) on the SI function.

7. Conclusions

In this work, we have taken the first step in performing a systematic evaluation of adaptive techniques for comprehending the complex behavior of mobile networks under a variety of network conditions, and in particular, during failures. In contrast to most current work, we look at real-time metrics with the aid of multi-variate visualization schemes to not only design and evaluate adaptive algorithms, but also to under-

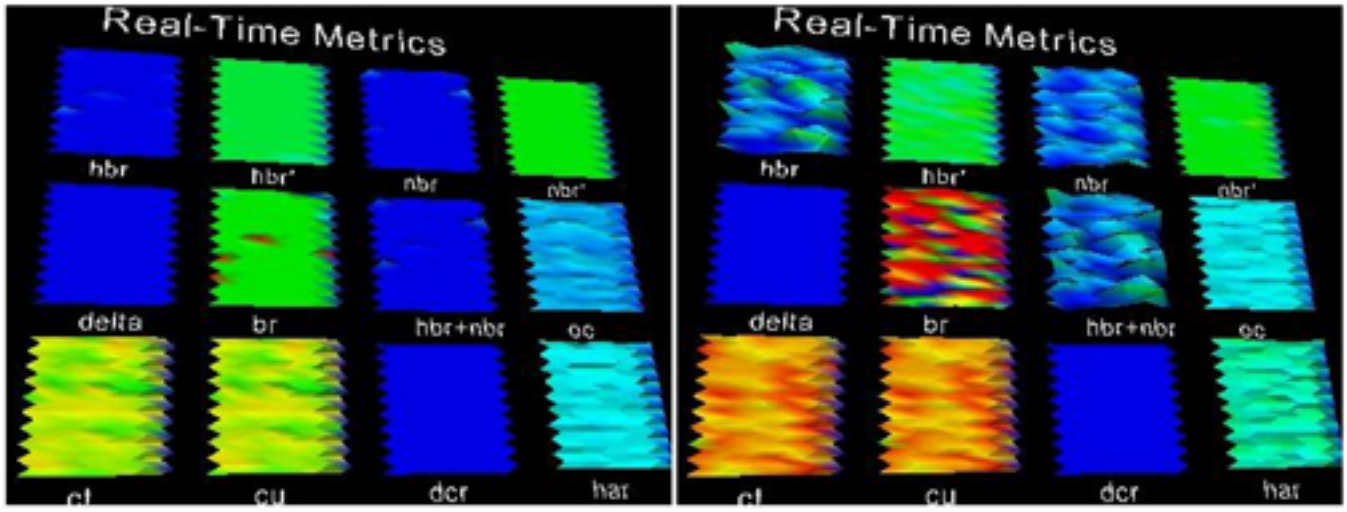

Figure 5. Normal Mode – Left: Lightly Loaded; Right: Heavily Loaded

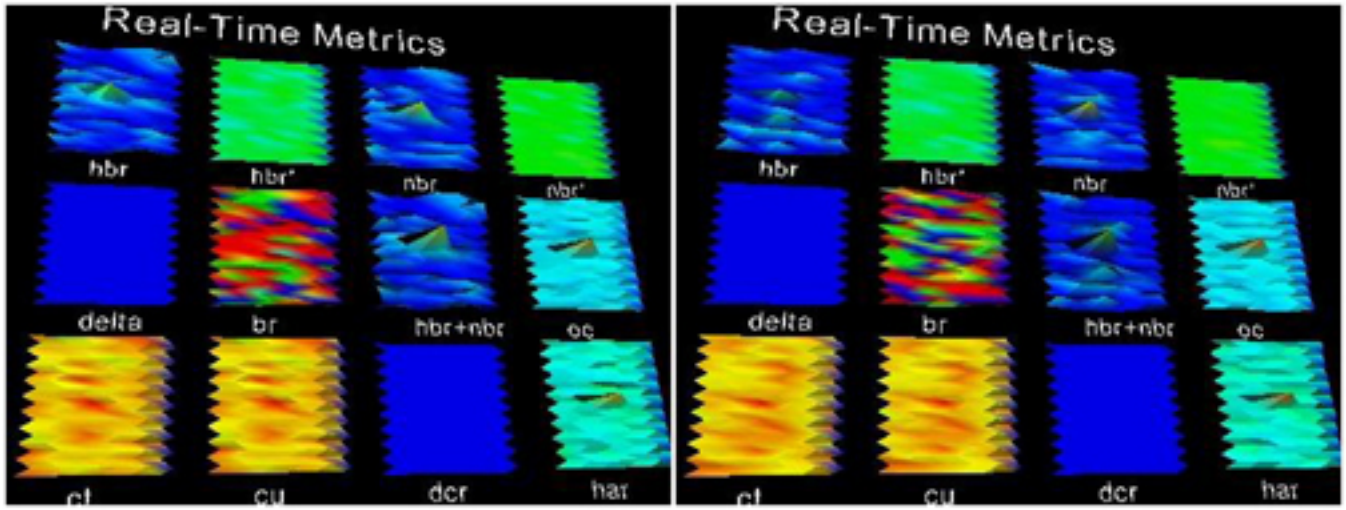

Figure 6. Degraded Mode Snapshots – Left: Onset of Hotspot; Right: 600 Seconds Later

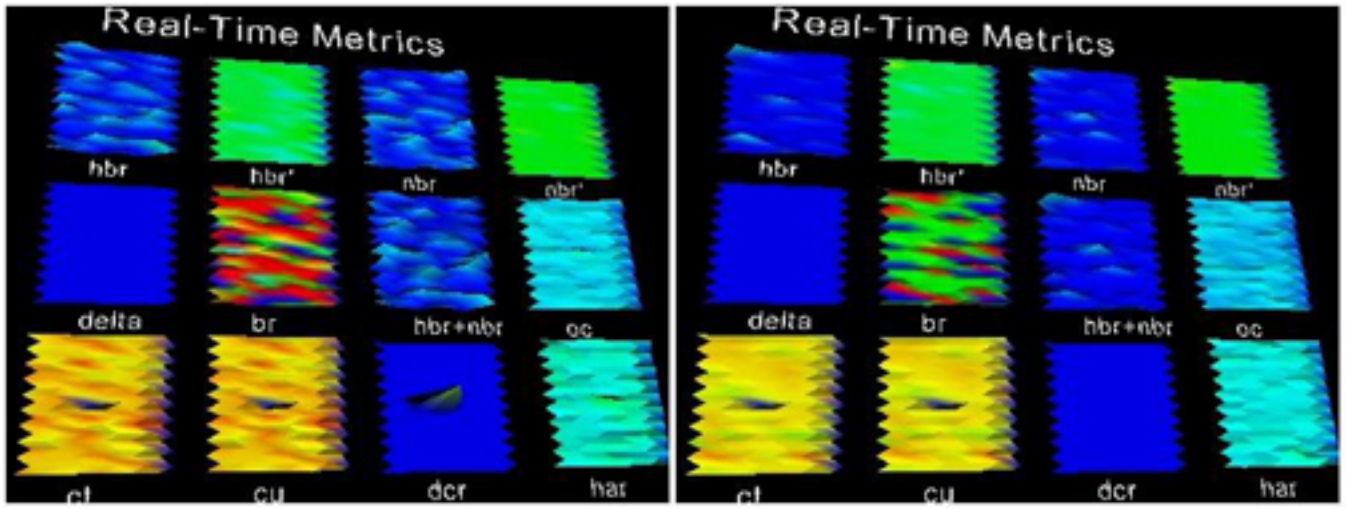

Figure 7. Transient Failure Mode Snapshots – Left: Onset of Failure; Right: 300 Seconds Later

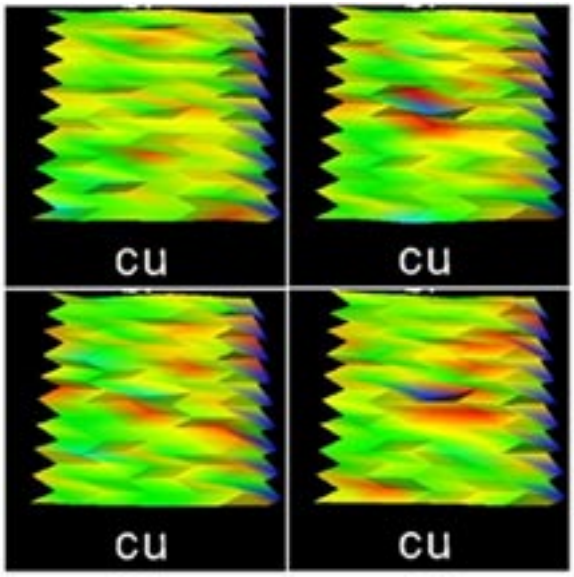

Figure 8. CU Oscillations During Recovery Mode

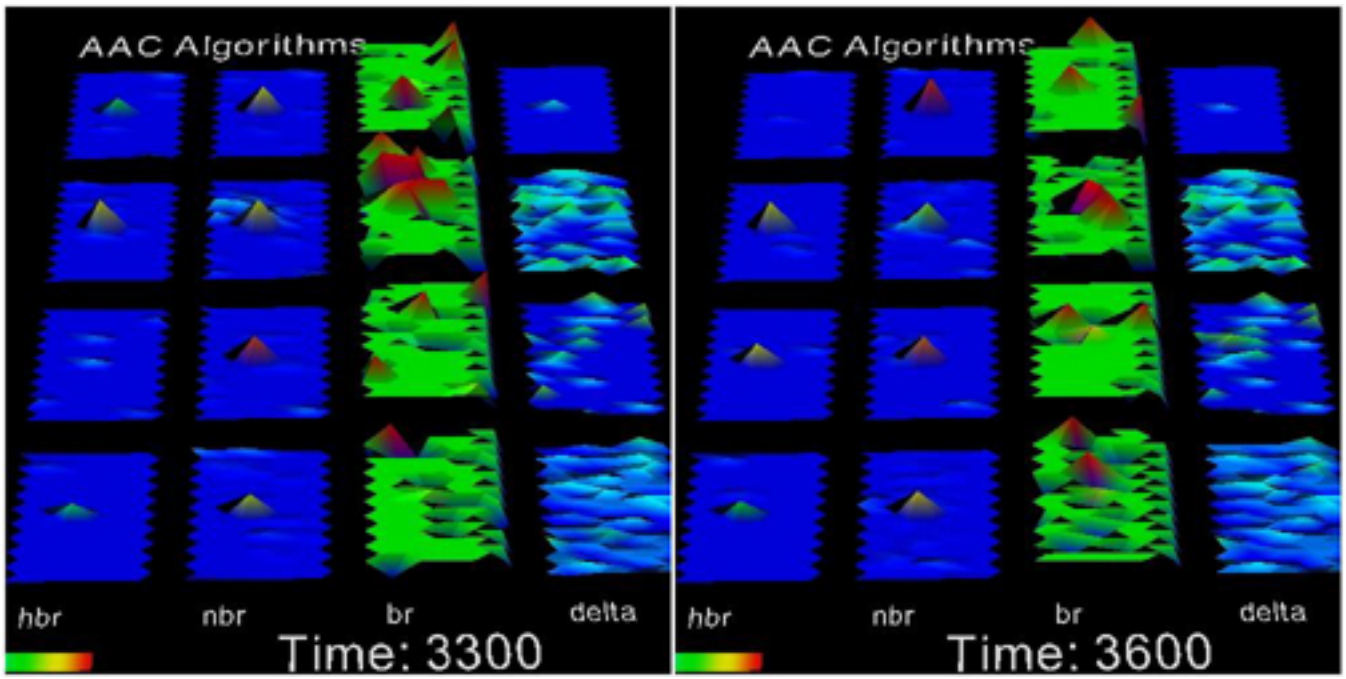

Figure 9. Comparing AAC 1, 2, 4, and 5

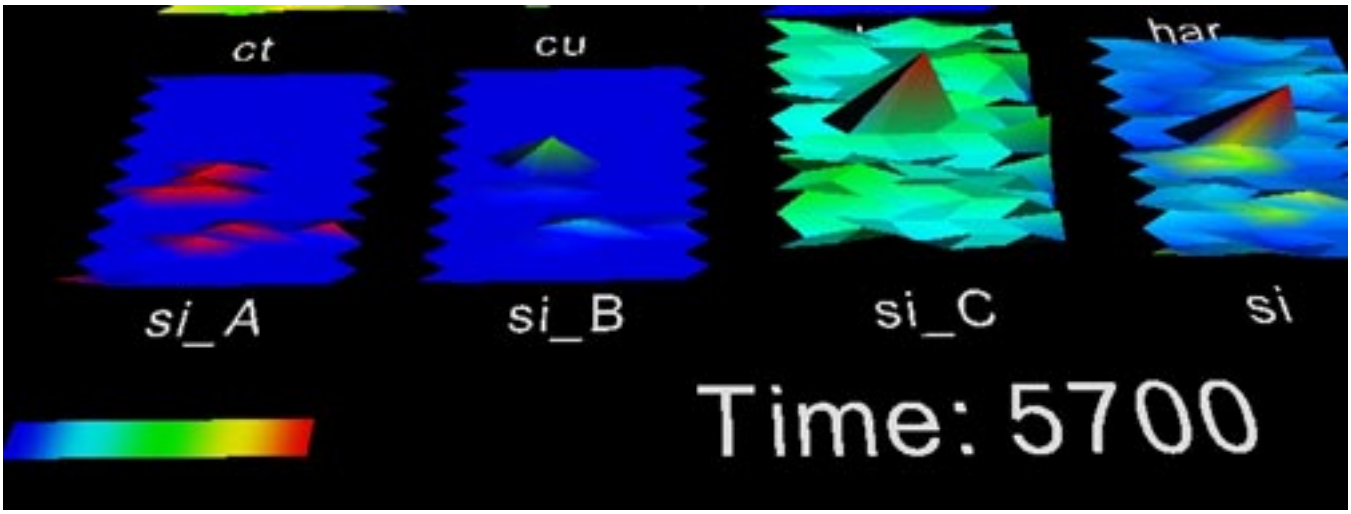

Figure 10. Survivability Index Components (si_A, Si_B, and si_C) and Function (si)

stand algorithm behavior across a large number of states. Visualization permits us to abstract the essence of large amounts of data, which is an inevitable outcome of using real-time metrics over long simulation runs.

We have introduced a number of real-time metrics that have been found be useful in influencing adaptive algorithms and for network monitoring. In particular, we have made progress in identifying and characterizing critical states of the network in terms of these metrics, and these will form the basis to a sound understanding of survivability of mobile networks.

Our experiments already revealed a problem with our channel access algorithm, which simply would not have been possible without the use of real-time metrics and interactive visualization. Such insights, coupled with domain expertise and the assistance of visualization, can make the analysis process very efficient. We also demonstrated how various adaptive admission control strategies can be quickly compared and contrasted with the help of the visualization system and better understanding of the influence and sensitivity of each metric on performance and cost.

In addition to continuing our investigation into tuning the metrics proposed in this work and designing better adaptive algorithms, the work on state characterization will be followed by specific algorithms to permit automatic feature detection and invocation of adaptation policies. Work on better understanding of the survivability index requires further investigation, especially for comparative analysis across multiple layers.

While the use of visualization tools in and of itself has proved to be of immeasurable value, work on building visualizations that are custom designed to the challenges facing mobile networks will be pursued in the future. The current visualizations display metric values in isolation. It is also important to explore spatial and temporal relationships in a more explicit fashion, for instance, between cells within a certain neighborhood or look at multiple metrics in combination.

8. References

[1] Boumerdassi, S. and Beylot, A.-L. "Adaptive channel allocation for wireless PCN," *MONET*, Vol. 4, 1999, pp 111-116.

[2] Lee, K. "Adaptive network support for mobile multimedia." *Proc. Of Intn'l Conf. on Mobile Computing and Networking*, Nov. 1996, pp 62-74.

[3] Lu, S. and Bharghavan, V. "Adaptive resource management algorithms for indoor mobile computing environments," *Proc. SIGCOMM*, Aug. 1996.

[4] Modiano, E. "An adaptive algorithm for optimizing the packet size used in wireless ARQ protocols," *WINET*, Vol. 5, 1999, pp 279-286..

[5] Oliveira, C., Kim, J.B. and Suda, T. "An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks," *IEEE JSAC*, Vol. 16, No. 6, Aug. 1998, pp 858-874.

[6] Priscoli, F.D. and Sestini, F. "Fixed and adaptive blocking thresholds in CDMA cellular networks," *IEEE Personal Communications*, April 1998, pp 56-63.

[7] Eckhardt, D.A. and Sttenkiste, P. "A trace-based evaluation of adaptive error correction for a wireless local area network," *MONET*, Vol. 4 (1999), pp 273-287.

[8] Iera, A., Marano, S. and Molinaro, A. "Transport and control issues in multimedia wireless networks," *WINET*, Vol. 2, 1996, pp 249-261.

[9] Anastasi, G., Lenzini, L., Mingozzi, E., Hettich, A. and Kramling, A. "MAC protocols for wideband wireless local access: Evolution toward wireless ATM," *IEEE Personal Communications*, Oct. 1998, pp 53-64.

[10] Harpantidou, Z. and Paterakis, M. "Random multiple access of broadcast channels with pareto distributed packet interarrival times," *IEEE Personal Communications*, April 1998, pp 48-55.

[11] Ryu, B. "Modeling and simulation of broadband satellite networks—Part II: Traffic modeling," *IEEE Communications Mag.*, July 1999, pp 48-56.

[12] Talukdar, A.K., Badrinath, B.R. and Acharya, A. "Integrated services packet networks with mobile hosts: Architecture and performance," *WINET*, Vol. 5, 1999, pp 111-124.

[13] Tipper,D., Ramaswamy, S. and Dahlberg, T.A. "PCS Network Survivability," *Proc. of the Mobile and Wireless Communication Networks Conference* (WCNC), Sept. 1999.

[14] Dahlberg, T. and Jung, J. "Survivable Load Sharing Protocols: A Simulation Study," *ACM/Baltzer Wireless Networks Journal (WINET)*, to appear. Retrieve from: http://www.cs.uncc.edu/~tdahlber/tdahlberbio.html#pub

[15] Dahlberg, T.A., Subramanian, K.R. "Visualization of Real-time Survivability Metrics for Mobile Networks," *Proc. of the Modeling and Simulation of Wireless and Mobile Systems* (MSWiM), Aug. 2000.

[16] Subramanian, K.R., Dahlberg, T. "Congestion Control in Mobile Networks," *IEEE Symposium on Information Visualization 2000* (InfoVis 2000), Oct. 2000, IEEE Computer Society.

[17] Cleveland, W.S. *The Elements of Graphing Data*, Wadsworth Inc., 1985

[18] LeBlanc, J., Ward, M.O., and Wittels, N. "Exploring N-dimensional databases," *Proceedings of Visualization 1990*, San Francisco, CA., Oct., 1990, pp 230-237.

[19] Feiner, S. and Beshers, C. "Automated Design of Virtual Worlds for Visualizing Multivariate Relations," *Proceedings of Visualization 1992*, Boston, MA, Oct. 1992, pp 283-290.

[20] van Wijk, J.J., van Liere, R. "HyperSlice: Visualization of Scalar functions of many variables," *Proceedings of Visualization 1993*, San Jose, CA, Oct. 1993, pp 119-125.

[21] Buja, A., McDonald, J.A., Michalak, J. and Stuetzle, W. "Interactive Data Visualization Using Focusing and Linking," *Proceedings of Visualization '91*, San Diego, CA, Oct. 1991, pp 22-25.

[22] Henderson, D.A. and Card, S.K. "Rooms: The use of multiple virtual workspaces to reduce space contention in a window based graphical user interface," *ACM Transactions on Graphics*, Vol. 5, No. 3, pp 211-243.

[23] Inselberg, A., and Dimsdale, B. "Parallel Coordinates: A Tool for visualizing Multi-dimensional Geometry," *Proceedings of Visualization '90*, San Francisco, CA., Oct. 1990, pp 361-375.

[24] Inselberg, A. "Multidimensional Detective,"*Proceedings IEEE Information Visualization*, Phoenix, AZ., Oct. 1997, pp 100-107.

[25] Ward, M.O. "XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data," *Proceedings of Visualization 1994*, Washington, DC., Oct. 17-21, 1994, pp 326-334.

[26] Fua, Y.H., Ward, M.O. and Rundensteiner, E.A. "Structure-Based Brushes: A Mechanism for Navigating Hierarchically Organized Data and Information Spaces," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 6, No. 2, 2000, pp 150-159.

[27] Jain, R. The Art of Computer Systems Performance Analysis, John Wiley & Sons, 1991.

[28] Schroeder, W., Martin, K. and Lorensen, B. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics,* Prentice Hall, Inc., Second Edition, 1998.

[29] Moitra, S.D., Oki, E. and Yamanaka, N. "Some New Survivability Measures for Network Analysis and Design," *IEICE Trans. Communications,* Vol. E80-B, No. 4, April 1997.

Teresa A. Dahlberg (tdahlber@uncc.edu) is an assistant professor of Computer Science in the College of Information Technology at the University of North Carolina at Charlotte. She holds a B.S.E.E. from the University of Pittsburgh and M.S. and Ph.D. degrees in Computer Engineering from North Carolina State University. She previously worked in the ECE department and was with IBM Corporation for several years. Her research, within the Multimedia Computing and Networking laboratory, is supported by the NSF. Her current funded projects include wireless access network survivability, interactive visualization of mobile network simulations, development of a secure, multimedia, wireless network testbed, and integrated solutions for fault tolerant middleware and networking.

Kalpathi Subramanian is an associate professor in the department of computer Science at the University of North Carolina at Charlotte. He obtained a B.E(Honors) in Electronics and Communication Engineering from the University of Madras in 1983, followed by MS (1987) and PhD (1990) in Computer Science at the University of Texas at Austin. Between 1991 and 1993 he was a post-doctoral member of technical staff at AT&T Bell Laboratories in Murray Hill, New Jersey. He was a visiting NIH faculty fellow during the summers of 2000 and 2001. His research interests include large scale scientific and information visualization, data representations and models, and medical data imaging/visualization.