

Visualization of Real-time Survivability Metrics for Mobile Networks

T. A. Dahlberg[#], K. R. Subramanian^{*}

[#]Electrical and Computer Engineering, ^{*}Computer Science
University of North Carolina at Charlotte
9201 University City Blvd.
Charlotte, NC 28223, USA
{tdahlber, krs}@uncc.edu

Abstract

In this work¹² we suggest real-time cost and performance metrics that are useful for survivability analysis at the radio layer of mobile networks. We demonstrate implementation of two of these metrics, to monitor blocking rates for adaptive admission control (AAC), using our simulation framework, to which we have added a data visualization (DV) layer. The DV layer uses 3-D animation to illustrate the effects of parameter selection on AAC performance, where the parameters of interest are those which control measurement and use of real-time metrics. Examples highlight the usefulness of the metrics to manage the tradeoff between new-call and handover channel requests and the capability of the DV layer to represent the spatial and temporal behavior of the AAC algorithm in a comprehensible form.

1. Introduction

One of the greatest challenges of future generation mobile networks is extending Quality of Service (QoS) guarantees to the mobile terminal. Normal operating mode for a wireless access network is characterized by bursts of demand from mobile users attempting to gain access to wireless links with varying signal quality. Over-allocation of resources to meet varying demand is not solely a cost issue, but is constrained by limited frequency spectrum. As a result, current research on mobile network architectures and protocols is placing heightened focus on the development of adaptive techniques to scale resource allocation policies in response to changing network conditions.

An overall focus of our work is on developing distributed, adaptive protocols for mobile networks to improve not only system performance, but system survivability as well. This means that we must consider system performance in the wake of failure. Our objective presents the following challenges. First is the identification of real-time metrics which adequately quantify system survivability. A review of survivability metrics given in [1] indicates the need for cost and performance metrics for survivability analysis for fixed networks. Even less work has been done to identify metrics for mobile network survivability. 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.

Consider that the measurement and use of real-time metrics is typically managed by various parameters to specify, for example, sampling rates and threshold values. Careful choice of parameter values is critical for managing the *sensitivity* of adaptive techniques. An inherent tradeoff exists in developing an adaptive protocol that can differentiate between "real problems which must be addressed" and "burst activity that should be ignored".

Understanding algorithm sensitivity within a mobile environment is not straightforward. Examining system-wide averages is of some interest. However, greater insight requires evaluation of time-varying system performance at each cellsite. Furthermore, survivability analysis requires that we consider system performance during the *transient period* immediately following a failure, during the *steady state* failure period, and following recovery [2]. This requires careful consideration of algorithm behavior on a spatial and temporal basis. The daunting result is huge amounts of data that must be analyzed, and hence the need for data visualization tools.

Visualization of scientific and engineering data has been an active area of research since the late 1980s, especially in the fields of biomedical imaging (anatomical reconstructions from CT, MRI, and Ultrasound) and fluid

¹ Supported by NSF grants NCR-9633444, NCR-9973672 and by a UNC Charlotte Faculty Research Grant

² An animated movie to illustrate the data visualization can be found at <http://www.cs.uncc.edu/~krs/mobvis>

flow analysis (vector, tensor fields). The research has resulted in the introduction of new 2D and 3D visualization techniques, volumetric illumination models, and algorithmic solutions to address very large datasets that are common in many applications.

In recent years, visualization has been applied to more abstract types of data, such as telephone databases [3] and multivariate data from various applications. However, application of data visualization (DV) to communication network analysis has not approached its full potential. Work on modeling performance in a telecommunication network is reported in [4]. The emphasis in this work is on visualizing backbone network topology, with some interactive query support for understanding network activity. One of the authors' earlier work on displaying activity of spinal cord neuron populations [5] has many common elements with the work presented here.

In view of this, the work presented herein is as follows. We define survivability as a cost/performance tradeoff and suggest several cost and performance metrics that are useful for measuring survivability at the radio layer (cell site) of a mobile network. We implement two of these metrics to monitor call blocking rates for use in adaptive admission control (AAC) schemes administered at the basestation level. We describe the extension of our simulation-based [6] analysis to include a DV layer. We use 3-dimensional DV animation to effectively display and communicate system level activity of the mobile network in terms of the cost and performance metrics. The DV system allows interactive exploration and rapid analysis of the effects of parameter selection on AAC performance, where the parameters of interest are those which control measurement and use of real-time metrics.

2. System architecture

The network topology includes basestations (BS), basestation controllers, mobile switching centers and associated databases interconnected to the fixed infrastructure as described for a Global Systems for Mobile communications (GSM) network. We use a load-sharing cellsite architecture (meaning that cells overlap) to provide multiple access points to mobile terminals (MT). This cellsite architecture uses two reuse partitions. A group of *long haul* channels is assigned to each BS using a relatively lower reuse ratio and larger coverage area. A group of *short haul* channels is assigned to each BS using a higher reuse ratio and smaller coverage area. Details are given in [6].

3. Survivability metrics

Table 1 lists a few of the many metrics which can be used to estimate system cost and performance for radio-level survivability analysis. Cost metrics can represent

"network infrastructure" costs (e.g. hardware/software complexity) or "user satisfaction" costs (e.g. signal quality, battery life). For example ASH can be used to evaluate network cost incurred by various implementations of a channel allocation protocol. NAP can be used to measure network costs incurred by various cell-site architectures. APD can be used to measure user costs, in terms of expected MT battery life, incurred by various cell-site architectures and CA protocols. CTR, NBR, FTR, and HAR are metrics used to measure the performance and cost of CA and admission control schemes.

Average signaling per handover (ASH)
Average delay per handover (ADH)
Average distance to access point (APD)
Number of access points (NAP)
Carried traffic rate (CTR)
New call blocking rate (NBR)
Forced termination rate (FTR)
Handover activity rate (HAR)

Table 1: Example metrics

Survivability Objective	Cost Metric
Keep BH-FTR < 1%, 99% of the time	BH-NBR
Keep RT-FTR < 3%, 95% of the time	RT-NBR

Table 2: AAC survivability objective

The example that we detail herein uses FTR as a performance metric and NBR as a cost metric to compare the survivability of two adaptive admission control (AAC) protocols. To do this, we define a survivability objective in Table 2. BH-FTR is the number of connections that were prematurely terminated due to a failed handover attempt or a network failure, within a 1 hour timeframe, divided by the number of handover requests that were made during the same timeframe. RT-FTR represents the same ratio over a much smaller timeframe (e.g. 100 sec). An FTR/NBR tradeoff was selected because the AAC algorithms that we implement lower FTR by allowing an increase in NBR. In comparing the schemes, the protocol determined to be the most survivable will be the one which meets survivability objectives at minimum cost.

4. Adaptive admission control

We implement two AAC algorithms, AAC 1 and AAC 2, that are evaluated at each BS. RT-FTR is defined as a sliding window metric. The algorithms sample the value of RT-FTR every κ seconds. At each sampling instant, AAC 1 calculates the value of RT-FTR over the prior $n\kappa$ -second window. E.g., if $\kappa=100$ and $n=3$, then RT-FTR is calculated every 100 seconds using data from the prior 300 second interval. AAC 2 works similarly except that it evaluates the rate of change (derivative) of RT-FTR, denoted $\Delta RT-FTR$.

The algorithms use the value of RT-FTR (or Δ RT-FTR) to adjust the value of a *guard-band* parameter, δ , as follows:

AAC 1

If $RT-FTR > \gamma_{high}$, increment δ
 If $RT-FTR < \gamma_{low}$, decrement δ

AAC 2

If $\Delta RT-FTR > \chi_{high} * n\kappa$, increment δ
 If $\Delta RT-FTR < \chi_{low} * n\kappa$, decrement δ

δ specifies the percentage of channels within a channel group that are not allowed to be used for new-call requests, but are set-aside for necessary handover requests. γ_{high} , γ_{low} , χ_{high} , and χ_{low} denote upper and lower threshold values. Use of a guard-band controls the tradeoff between improved performance, represented by a low forced termination rate, and increased user cost, represented by new-call blocking rates. We use RT-FTR to dynamically modify guardband size to be only as small as is necessary to meet performance objectives. The CA protocol implemented is detailed in [6].

5. Data visualization layer

We have constructed 3D visualization tools that have proven to be indispensable in analyzing the behavior of the real-time survivability metrics. The visualization system has been constructed using the Visualization Toolkit (VTK) [7] and OSF/Motif for the graphical user interface (GUI), and runs on both SGI and Linux workstations. The system is currently capable of displaying the activity (over time) of FTR, δ , and NBR. It is highly useful to compare multiple simulation runs that vary in the choice of adaptive algorithms and in the choice of metric parameters.

The design allows for a spread-sheet style display of an arbitrary number of simulation runs, as shown for two runs in Fig. 1. Each row represents a simulation run, and can accommodate an arbitrary number of real-time values, where each value is mapped onto a cell matrix. For example, the three square images in row 1 of Fig. 1a represent FTR, δ , and NBR (left to right). Each image is a visual representation of the value of a metric, at each of the 105 basestation cells (viewed as a square grid), at one timestep in a simulated run. The second row of three images in the figure represents the same data for a different simulation run, at the same simulation time step.

Fig. 1 illustrates a *height field view*, where the level of activity is represented by raising each cell lattice point in a direction orthogonal to the cell plane by the metric value times a scale factor. Height fields represent a terrain style geometry that dynamically changes over time. The use of a *color map view* and *transparent planes* are shown at <http://www.cs.uncc.edu/~krs/mobvis>. Such simple representations are very effective in allowing the user to

focus on important information, such as transient activity following a failure.

The DV interface also provides VCR style controls, that allow the sequence to be viewed as continuous play, fast-forward, rewind, or single step. As data is stored in main memory, the analysis is highly interactive and the data can be reviewed in a convenient manner. Each of the runs herein spans 5400 simulated seconds with visualization sequences generated every 20 seconds. This provides 270 sequences for each run to support comparisons among the various results.

6. Simulation

A geographical data set, a traffic model, and a mobility model together describe user movements and calling patterns throughout a simulation run. The cell-site architecture modeled has 105 cells, each having 2 reuse partitions, assumed to serve a region of Charlotte, NC. Each short- and long-haul channel group at each of the 105 BS have 11 channels with a reuse ratio of 3 and 13, respectively. Details are in [6].

Note that the sensitivity of the AAC schemes is managed by the following parameters: sampling rate (κ), window size ($n * \kappa$), and the upper and lower threshold values (γ and χ). Table 3 lists the simulation scenarios studied, in which we varied these parameter values. For all cases, system-wide offered load is 50 calls/sec. Call duration follows an exponential distribution with a mean of 120 seconds. The distribution of arriving calls to cells is based on demographic information in the geographical data set. Simulation includes a 30 minute rampup period, then a 1 hour period during which BH data is collected.

Table 4 details each of the failure modes. The 105 cells that model the network are arranged in a square grid. Cell 53 is approximately in the center of that grid. For each failure, all short haul channels and 5 (of the 11) long haul channels are marked unavailable. Connections using a failed channel are forced to terminate and will attempt to reinitiate their connection with probability 0.5.

Busy hour performance for the runs in Table 3 are shown in Table 5.

7. Discussion of results

Here we demonstrate that algorithm sensitivity is influenced by the interpretation of metrics and illustrate the dramatic impact of the DV animations to provide insight into dynamic algorithm behavior.

7.1. Setting a baseline

To emphasize the need for AAC, refer to rows 1-2 in Tables 3, 5. For each of these 5 minute failure scenarios, no AAC is implemented. Every 100 seconds, metrics are

sampled over a 300 second window, but no adjustments are made to the size of the guard band. In the algorithms denoted Ref 1 and 2 (rows 1 and 2 of the tables), guard band size is fixed at zero and one, respectively. BH results show that a BH-FTR of 1.48%, for Ref. 1, is higher than our Table 2 objective. BH results for Ref. 2 show improvement, as BH-FTR is reduced almost to zero. However, the cost incurred, a 68.6% new call blocking probability, is completely unacceptable! If we are to use a guard band to try to improve the FTR/NBR ratio, *it is essential that we implement an adaptive technique.*

Row	Algorithm	κ	$n * \kappa$	$\gamma_{high}, \gamma_{low} (\chi_{high}, \chi_{low})$
5 minute failure:				
1	Ref 1	100	300	δ fixed at 0
2	Ref 2			(δ fixed at 1/11)
5 minute failure:				
3	AAC 1	100	300	0.5, 0.03
4	AAC 1	60	180	
5 minute failure:				
5	AAC 2	100	300	(0.18, -0.18)
6	AAC 2			(0.36, -0.36)
7	AAC 2	60	300	(0.18, -0.09)
8	AAC 2	60	180	
9	AAC 2			
30 minute failure:				
10	AAC 1	60	180	0.5, 0.03
11	AAC 2			(0.18, 0.0)

Table 3: Simulation scenarios

7.2. AAC 1 parameter selection

Rows 3-4 of Tables 3, 5 show performance of AAC 1 using different metric sampling and window sizes. We expect smaller sampling and window sizes to enable the algorithm to detect the onset or removal of a failure condition more quickly. But, this also increases the likelihood that the algorithm will misinterpret normal burst activity as a failure condition. The BH results indicate that we are able to move closer to our survivability objective using smaller sampling and window sizes. The DV results (not shown) indicate that the algorithm operates correctly and appears stable. Let us now look at AAC 2 for which parameter selection proved to be more difficult.

7.3. AAC 2 parameter selection

Refer to rows 5-9 of Tables 3, 5. We experimented with several variations in sampling and window sizes and threshold settings. Looking only at the BH results of Table 5 is confusing. Row 5 shows a reasonable improvement in BH-FTR, but at a cost of BH-NBR = 6.3%. The BH-NBR value seems high given that the guardband size never becomes larger than 1 throughout the run. Row 7 results are comparable to AAC 1 results.

This leads one to believe that the chosen parameters for this scenario result in an improved algorithm.

Failure	Description
5 min.	At time 3000sec, cell 53 loses all short haul and 5 long haul channels. Channels restored 300 sec. After failure.
30 min.	At time 3000sec, cell 53 loses all short haul and 5 long haul channels. Channels restored at time 4800 sec.

Table 4: Failure scenarios

Row	Carried Traffic (Erlangs)	BH-FTR (%)	BH-NBR (%)
1	5836	1.48	2.73
2	1884	0.02	68.60
3	5848	1.28	2.54
4	5854	1.24	2.44
5	5622	1.27	6.30
6	5838	1.42	2.70
7	5847	1.21	2.56
8	5831	1.45	2.81
9	5845	1.37	2.58
10	5850	1.27	2.49
11	5847	1.39	2.54

Table 5: Busy hour performance

For further insight, however, refer to Fig. 1. Top row images are for the run in row 7 which seems to be an improved algorithm. At time 2960, just prior to the failure, the four lighter "bumps" in δ indicate that the guardband has been increased in four cells. In viewing the entire run, however, we see that these increases are made due to bursty activity during the rampup period and remain throughout the run. I.e. δ is increased due to bursts, and then remains "stuck" at a high value. In fact, the occurrence of the failure is not even detected by the row 7 run, as shown by the peak in FTR, in Fig. 1b top row, in the failed cell and the lack of a corresponding peak in δ .

This example dramatically illustrates the DV capabilities. BH results seemed inconsistent and confusing. Specifically, the row 7 results seemed to provide a noticeable improvement in BH results for AAC 2. However, with the DV animations, we were quickly able to understand the defect in algorithm behavior and make the appropriate modifications. The bottom row of Fig. 1 illustrates acceptable AAC 2 performance using parameter settings from Row 9 of the tables. The key parameter that seemed to "fix" AAC 2 was to set χ_{low} to zero so that any small decrease in FTR will pull down δ .

7.4. Additional failure scenario

Next, we compare performance using a different failure scenario. For each algorithm, we chose parameter settings that resulted in the best performance for the 5 minute failure mode. Setup and BH results for the 30 minute failure mode are listed in rows 10-11 of Tables 3, 5. Although AAC 1 performed noticeably better than AAC 2 for the various runs using a 5 minute failure mode, BH results show that the reverse is true for the 30 minute failure mode. For further understanding, refer to Fig. 2.

Prior to the snapshots shown, both algorithms exhibit good intolerance to burst activity. After rampup, the δ map remains dark and flat, indicating no activity in guardband adjustment. At time 3060 (Fig. 2a), both algorithms have detected the failure and increased the guardband (via δ) accordingly. At time 3120 (Fig. 2b), however, AAC 1 increases δ but AAC 2 decreases δ . Why? Recall, that AAC 1 compares the value of FTR to a fixed threshold value. While FTR decreases slightly from time 3060 to time 3120, it still remains higher than the acceptable threshold level. However, AAC 2 is looking at the "rate of change" of FTR. For our "5 min" failure runs described in the previous sections, FTR decreased very slowly following the initial failure burst. Therefore, we needed to make the lower threshold for Δ FTR extremely small to avoid having the guardband "stuck" at a higher than needed value. The result, however is that AAC 2 is too sensitive. The value of δ is pulled down too soon.

To explain, we observe that the *transient failure period* begins at the time of failure, time 3000, and ends around time 3300, when reinitiation attempts have ceased. The *steady state failure period* persists until all channels have been restored at time 4800. Throughout this time, δ move to and remains at zero for AAC 1 and oscillates between 0 and 1/11 for AAC 1, as shown in Fig. 2 c,d.

Because our channel allocation scheme is load-based, an increased guardband is NOT required during the steady state failure period. This is verified by portions of the animation (not shown) that show real-time values of FTR do NOT exceed the RT-FTR survivability objective stated in the last row of Table 2.

The results, again, are dramatic. BH performance implies that AAC 2 is superior to AAC 1 in handling the 30 minute failure. However, the DV animation demonstrates that, in fact, AAC 1 causes guardband size to unnecessarily oscillate throughout steady state failure period resulting in a possibly unstable algorithm.

8. Concluding remarks

We interpret survivability of mobile networks as a cost-performance tradeoff and introduced several metrics that are useful to represent system costs, user costs, and

system performance. We showed how these metrics can be used to create a survivability objective used for comparison of diverse architectures and protocols.

We detailed our implementation of FTR and NBR, as real-time sliding window metrics used for adaptive admission control. We compared survivability of two AAC protocols by viewing FTR as a performance metric, which can be improved at a cost of increased NBR.

Our simulation results demonstrated how the use of FTR and the parameter settings associated with this metric affect the performance and sensitivity of the AAC schemes.

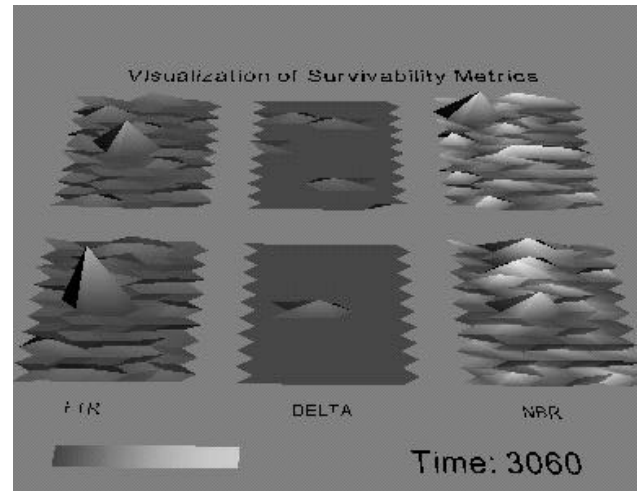
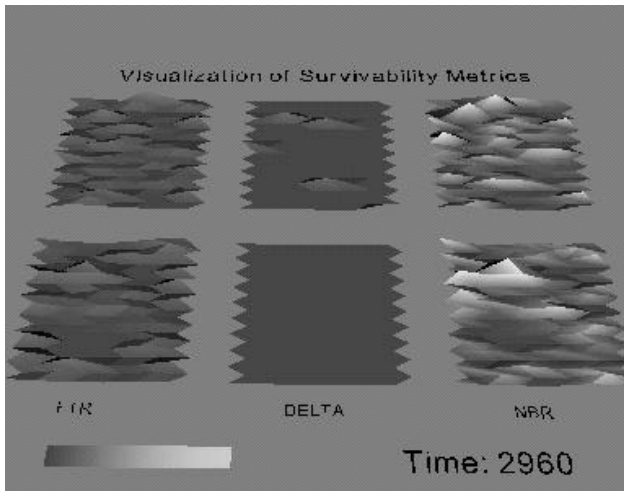
We introduced our newly developed visualization layer to complement our mobile network simulation tool. We showed the capability of the DV tool to represent huge amounts of spatial and temporal data generated by several mobile network simulations in a comprehensible format.

The visualizations proved to be an essential means to illustrate AAC algorithm behavior. In several instances, the busy hour results implied that a particular algorithm or parameter setting for the FTR metric provided a performance improvement. However, the DV animation for these instances demonstrated that, in fact, the algorithm under study was behaving inappropriately. For us to have come to the same conclusions without the use of visualization would have required a huge effort in parsing through volumes of output data.

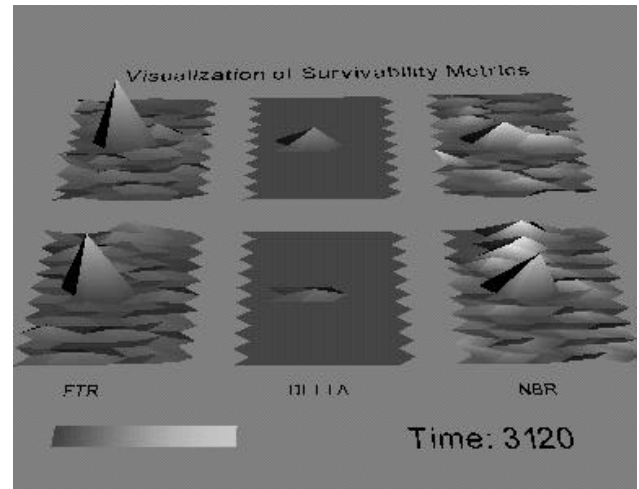
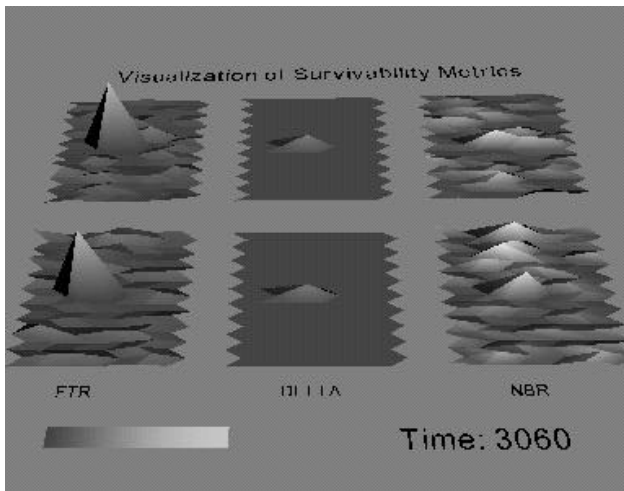
We continue to improve our definition of survivability of mobile networks by integrating multiple cost-performance metrics into one function. We continue to expand the capabilities of our data visualization tool.

References

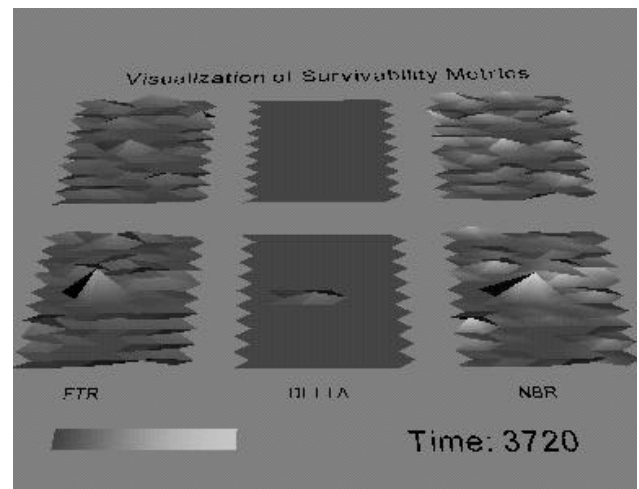
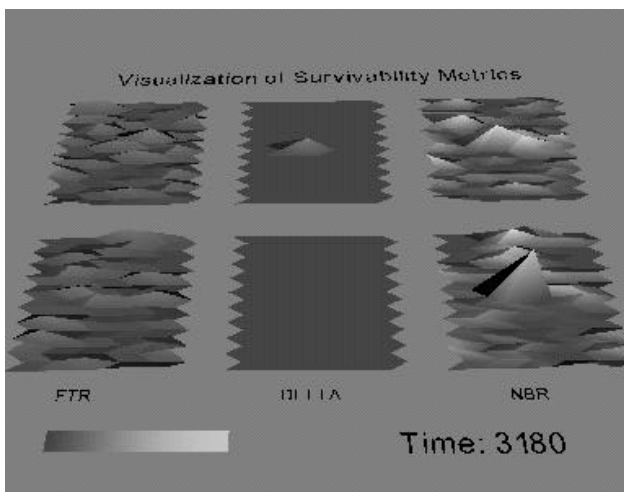
- [1] S. D. Moitra, E. Oki, and N. Yamanaka, "Some New Survivability Measures for Network Analysis and Design", IEICE Trans. Communications, Vol. E80-B, No. 4, April 1997.
- [2] D. Tipper, S. Ramaswamy, T. Dahlberg, "PCS Network Survivability", Proc. Mobile and Wireless Communication Networks conference (MWCN), Sept. 1999, invited paper.
- [3] V. Anupam, S. Dar, T. Leibfried, E. Petajan, "DataSpace: 3D Visualization of Large Databases", Proc. IEEE Information Visualization 95, Oct. 1995, IEEE Computer Society, Los Alamitos.
- [4] E. E. Koutsofios, S. C. North, R. Truscott, and D. A. Keim, "Visualizing Large-Scale Telecommunications Networks and Services", IEEE Visualization 1999 Proceedings, Oct. 1995
- [5] K. R. Subramanian, D. P. Bashor, and W. V. Hasty, "Multilevel Visualization of Spinal Reflex Circuit Simulations and Services", IEEE Computer Graphics and Applications, Vol. 17, No. 3, 1997.
- [6] T. A. Dahlberg and J. Jung, "Survivable Load Sharing Protocols: A Simulation Study", ACM/Baltzer WINET, to appear, <http://www.ece.uncc.edu/faculty/fac/tdahlber..>
- [7] W. Schroeder, K. Martin, and B. Lorenzen, *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, Prentice Hall, Inc., 2nd ed., 1998.



(a) (b)
 Figure 1: Comparing AAC 2 parameter settings (top: row 7, bottom: row 9)



(a) (b)



(c) (d)

Figure 2: Comparing AAC 1 to AAC 2 (top: row 10, bottom: row 11)