Events Defined by Duration and Severity, With an Application to Network Reliability

Richard A. Becker
Linda A. Clark
and
Diane Lambert*

Abstract

Communications networks are highly reliable and almost never experience widespread failures. But, from time-to-time performance degrades and the probability that a call is *blocked* or fails to reach its destination jumps from nearly zero to an unacceptable level. High, but variable, blocking may then persist for a noticeable period of time. Extended periods of high blocking or *events* can be caused by congestion in response to natural disasters, fiber cuts, equipment failures, and software errors, for example. Because the consequences of an event depend on the level of blocking and its persistence, lists of events at specified blocking and duration thresholds, such as 50% for 30 minutes or 90% for 15 minutes, are often maintained. Reliability parameters at specified blocking and duration thresholds, such as the mean number of events per year and mean time spent in events, are estimated from the lists of reported events and used to compare network service providers, transmission facilities, or brands of equipment, for example.

This paper shows how data obtained with two-stage sampling can be used to estimate blocking probabilities as a function of time. The estimated blocking probabilities are then used to detect and characterize events and to estimate reliability parameters at specified blocking and duration thresholds. Our estimators are model–free, except for one step in a sampling bias correction, and practical even if there are hundreds of millions of observations. Pointwise confidence intervals for reliability parameters as a function of blocking and duration thresholds are built using a kind of "partial bootstrapping" that is suitable for very large sets of data. The performance of the algorithm for event detection and the estimators of reliability parameters are explored with simulated data. An application to comparison of two network service providers is given in this paper, and possible adaptations for other monitoring problems are sketched.

Keywords Bridging, Event Detection, Monitoring, Network Reliability, Partial Bootstrap, Thresholds, Two–Stage Sampling.

---

*Richard A. Becker is on the technical staff of AT&T Laboratories and Linda A. Clark and Diane Lambert are on the technical staff of Bell Laboratories, Lucent Technologies. They are grateful to Cheng Ge Cheng, Karel Ehrlich, Jim Frauenthal, Carol Rigler, Bob Sung, and Eze Wills for their patient explanations of network testing and their help in obtaining and understanding the data. They also thank the referees for helpful advice.

1 Background

Communications networks are highly reliable, but from time–to–time performance degrades abruptly and some calls are *blocked* (fail to complete). When the fraction of calls from an origination (such as a city) to a destination (perhaps another city) is high enough for a long enough period of time, an event is recorded. The level of blocking and its duration vary from event to event, and the consequences of an event vary among users. For example, casual users are unlikely to notice a 25% blocking probability, but a heavy user may notice a 25% blocking probability, especially if it persists. Thus, reliability of communication networks is monitored over a range of blocking and duration thresholds. More background on network reliability can be found in the papers by Glossbrenner (1993) and Daneshmand and Savolaine (1993).

The completion status of each call placed on a network is recorded, and, in principle, this information can be used to compute blocking probabilities, which in turn can be used to detect events at different blocking and duration thresholds. But information about the completion status of calls is highly proprietary, and it is not shared among network service providers or equipment manufacturers. Thus, studies that compare the reliability of competing networks may place test calls over each network of interest. The outcomes of the test calls are used to estimate the blocking probabilities on the different networks, and the estimated blocking probabilities used to detect extended periods of high blocking (events, in our terminology). The detected events are then used to find point and interval estimates of network reliability parameters that are the basis of competitive claims.

The outline of the steps that lead from test calls to reliability comparisons is clear, but filling in the details is challenging. First, a list of detected events must be based on test calls, rather than all calls, placed on the network. Thus, the list of detected events may be erroneous, omitting some events that occurred and including others that are false alarms. Second, hundreds of millions of test calls may be placed in a year, making computations difficult. Third, there is a lot of data, but only a small part of it (that obtained near an event) directly helps to detect and characterize any particular event. Thus, we are data rich and information poor.

Moreover, realistic parametric models for this application are not easy to work with. A stochastic process with jumps at random start and stop times is needed to describe the timing of events, and random functions that vary with time are needed to describe the evolution of blocking probabilities during events. Formulating such a model is not necessarily difficult. Section 2 describes a simple model with constant blocking during events and exponential durations and waiting times between events, for example. But, finding the probability that a random blocking probability exceeds a threshold π for at least δ hours under the model — *i.e.*, the probability that an event occurs — is not easy for realistic probability models. Finding tractable estimators for the parameters of the model and associated reliability parameters as a function of blocking and duration thresholds may also be difficult for realistic probability models. Standard errors are also needed to understand the precision of the estimates and the strength of competitive claims based on the data. Finally, the choice of parametric models must be validated with data. (The simplest model, for example, is inappropriate for our application because waiting times and event durations are far from exponentially distributed.) In light of the difficulties presented by parametric modeling, we propose statistical methods that do not require detailed probabilistic analysis.

This paper is organized as follows. Section 2 defines a statistical model for generating events. The sampling algorithm and an example of the kind of data that is available for detecting events are described in Section 3. An algorithm for detecting events from the data and characterizing detected events is given in Section 4. Estimation of network reliability parameters from lists of reported events is described in Section 5. The performance of our estimators on simulated data is summarized in Section 6. Partial bootstrap confidence intervals and standard errors for estimates of network reliability parameters are developed and applied to real data in Section 7. Some extensions and summary conclusions are discussed in Section 8.

2 A Statistical Framework

Consider an origination and destination pair (C_1, C_2). The origination and destination may be customers, cities, urban areas, or regions, but to be specific we call them cities. A call sent at time t from C_1 to C_2 is blocked with a probability μ_t that is affected by holiday congestion, natural disasters, equipment failures, software errors, day of week, and time of day. The blocking probability may also depend on traffic between other cities, because traffic routes can be dynamically re-configured to optimize network resources. A *city pair event* at thresholds (π, δ) occurs at (C_1, C_2) when μ_t remains at or above a blocking threshold π for t in an interval at least δ hours long.

A simple model of city pair events of varying severity and duration can be built by assuming that the blocking probability μ_t for the city pair jumps back and forth between zero (no blocked calls) and a random, positive value (positive probability of blocked calls) at random times. If B_i is the i^{th} level that the blocking probability assumes and $B_i > 0$, then after remaining a random duration T_i at B_i, the blocking probability returns to zero for a random duration T_{i+1} before jumping to another positive level B_{i+2}. The initial blocking probability is taken to be $B_1 = 0$ and T_i and B_i are assumed to be independent. Independence of severity and duration is plausible in our application, because we consider only thresholds high enough that corrective action is taken as soon as possible. This model can be written as

$$
\begin{aligned}
B_1 &= 0 \\
B_i &\sim \begin{cases} I_0 & \text{if } B_{i-1} > 0 \\ F_B & \text{if } B_{i-1} = 0 \end{cases} \\
T_i &\sim \begin{cases} F_0 & \text{if } B_i = 0 \\ F_1 & \text{if } B_i > 0, \end{cases}
\end{aligned}
\tag{1}
$$

where I_0 is the distribution that assigns probability one to zero, F_B, F_0 and F_1 are distributions on the positive real line, and $B_2, B_3, \ldots, T_1, T_2, \ldots$ are independent. More flexible models can be built by modeling the nonzero blocking probabilities as polynomials or other curves, rather than as constants, and assigning distributions to the parameters of the curves. We do not use the model (1) or its variants to derive estimates of network reliability parameters and standard errors, but it may suggest the statistical context of our application.

A *city pair event* at severity and duration thresholds (π, δ) starts at time $S_k = T_1 + \ldots + T_{k-1}$ if $B_k \geq \pi$ and $T_k \geq \delta$. A *network event* at thresholds (π, δ) is declared whenever there is at least one city pair event at thresholds (π, δ). Thus, the list of network events $\mathcal{E}$ is a union of the lists of city pair events: $\mathcal{E}(\pi, \delta) = \cup_{ik}[S_{ik}, S_{ik} + D_{ik}]$, where S_{ik} and D_{ik} are the start and duration of event k for city pair i. The endpoints of the intervals in $\mathcal{E}$ are the starts and stops of the network events (see Figure 1). Note that by definition each city pair event in a network event must meet the threshold requirements, so a network event at a threshold of one hour cannot be built from consecutive ten minute city pair events for six different cities, for example. This is reasonable in our application, because the goal is to measure impact on customers, and often the same customer is not affected by city pair events in different locations. The idea of counting city pair events that overlap in time as one network event can also be found in the papers by Glossbrenner (1993) on network reliability and Bloomfield, Royle and Yang (1993) and Smith and Huang (1994) on environmental monitoring.

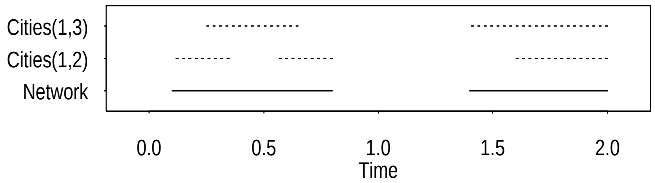

Figure 1: The dotted segments show the start to the end of five city pair events for two city pairs. The solid segments show the corresponding two network events.

It is important to note that we define events in terms of the blocking probabilities $\{\mu_t\}$, not in terms of the data. In contrast, other papers (e.g., Smith 1986, 1989) declare events when the maximum measurement in a fixed unit of time, such as a day, crosses a severity threshold. While we could use the fraction of blocked test calls in a fixed unit of time to define an event, using the blocking probabilities instead is preferable, because it allows an event to be defined independently of sampling and measurement processes. Defining events in terms of blocking probabilities, rather than in terms of the outcomes of test calls, also seems more in line with common understanding of the intent of regulations (see Fairly, 1989). Of course, the distinction between events defined from $\{\mu_t\}$ and events defined from the data is unimportant if measurement error is negligible, which it is not if the data are binary or if μ_t is not monitored continuously.

3 Sampling and Data

In our application, two or more networks are monitored under the same conditions, in the sense that the same city pairs are used for all tested networks and test calls are placed at nearly the same time over all tested networks. Thus, all networks tested experience the same natural disasters and holidays, for example. The test calls are a tiny fraction of the traffic handled by each network, so they do not cause congestion or otherwise affect network reliability.

On the networks of interest, the blocking probability μ_t between any city pair at any time t is usually zero. Therefore, to reduce sampling costs during uninteresting periods (periods with nearly zero blocking probabilities), data are collected by two-stage sampling. *Screening* calls are routinely sent from city C_1 to city C_2 once every 15 minutes. If the screening call from C_1 to C_2 completes, then no further call is made from C_1 to C_2 until the next screening call is due. If the screening call is blocked, then *rapid sampling* begins and calls are placed rapidly from C_1 to C_2 until the next screening call is due. The number N of calls sent in a sampling interval is not fixed, but often does not vary greatly. Before the next screening call from C_1 to C_2 is made, the estimated blocking fraction $f = N^{-1} \sum Y_j$ for the sampling interval that just completed is computed. If f exceeds a pre-determined sampling threshold, then rapid sampling continues in the next sampling interval; otherwise, rapid sampling stops. The decision to continue or stop rapid sampling is made separately for each city pair every time a screening call is due. To avoid cutting off data collection prematurely during an event, the threshold for cutting off rapid sampling is much lower than the lowest blocking threshold π of interest.

The outcome of a test call Y_t placed at time t is a Bernoulli(μ_t) random variable. All calls are assumed to be independent given the blocking probabilities $\{\mu_t\}$, because there is no reason to believe that test calls interact.

Figure 2 shows the data collected during a period of unusually persistent high blocking between two cities on one network. The plot is divided into three horizontal strips. The numbers of test calls in the sampling intervals are shown in the lower strip. In all, nearly 4400 calls were sent in the period shown. The large middle strip uses short, light dashes to show the observed blocking fractions in the sampling intervals, with dashes extending from the first to the last call in an interval. The points at a blocking probability of zero before the start of the event represent completed screening calls. The numbers of blocked and completed calls in short bins of time are shown in the top strip; each bin is six minutes wide in this example. The bins are chosen to match the resolution of the printer and are unrelated to the sampling intervals.

4 Rules for Reporting Events from Test Calls

Network events are found by taking unions of city pair events. We consider two types of rules for detecting city pair events: standard rules based on blocking fractions computed for the sampling intervals and new rules based on the smoothed binary outcomes of the test calls.

4.1 Event Detection from Sample Blocking Fractions

Using blocking fractions for fifteen minute sampling intervals, for example, a one hour city pair event at 50% blocking is reported whenever the observed blocking fraction in each of four consecutive

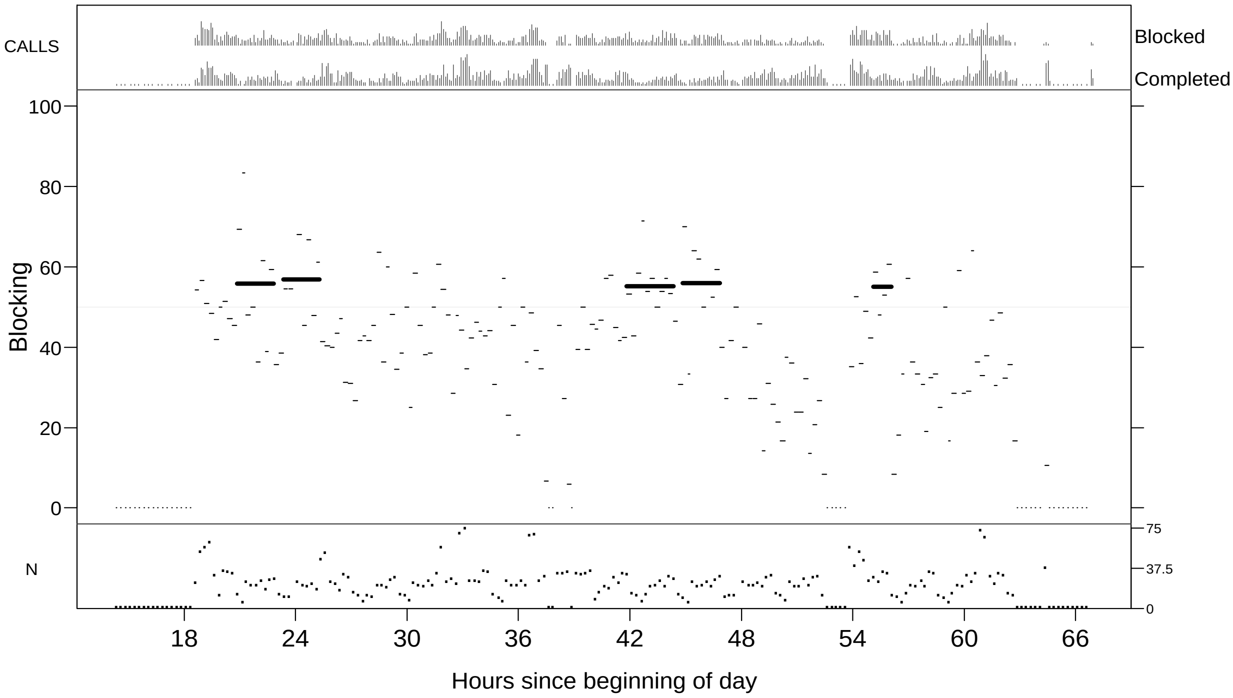

Figure 2: Data for a city pair during a period of unusually persistent blocking. The points at zero before hour 18 show completed screening calls. The short, light dashes show the observed blocking fractions in the sampling intervals during rapid sampling. The bold dashes show periods that qualify as a reported event at thresholds of (50%, one hour) under a rule based on fifteen minute blocking fractions. The numbers of blocked and completed calls throughout time are shown in the top strip; the numbers of test calls in the sampling intervals are shown in the lower strip.

fifteen minute sampling intervals exceeds .5. This rule can be refined to allow the blocking fraction to dip below .5 for one of the intermediate sampling intervals, as long as the overall average blocking fraction for the entire reported event stays above .5. This refinement offers some protection against missing events due to sampling variability when the blocking probability μ_t is near the blocking threshold π, but at the expense of more false alarms. Five reported city pair events, each with an estimated blocking probability of at least .5 for at least an hour, that were detected using the refined rule are shown in Figure 2.

Rules based on aggregated data (the observed blocking fractions for the sampling intervals in our application) are simple to apply, at least if no refinements are made, but they can be undesirable. First, the coarsened time scale is often chosen in terms of the sampling interval, which is determined by financial constraints rather than by the rate of change in μ_t. Second, aggregation obscures the beginning and end of an event, so rules based on aggregated data may not have good false positive and false negative probabilities. If an event stops in the middle of a sampling interval, for example, then the blocking fraction for the interval as a whole may fall below the blocking threshold, causing the end of the event or even the entire event to be missed (a false negative). Or, the entire interval may be included in the reported event, overestimating the end of the event (possibly leading to a false positive). If there are about 25 observations per sampling interval and $\mu_t = .6$ throughout the interval, then only about 50% of the sequences of four consecutive fifteen minute blocking fractions exceed .5. Thus, a city pair event with 60% blocking for one hour that starts at the time of a screening call is likely to be missed at thresholds of (.5, one hour) because it is only detected with probability 0.5. Similarly, a city pair event with 60% blocking for several hours is likely to be reported as several shorter city pair events, rather than as one long event. Returning to Figure 2, it is not unreasonable to argue that the first four events found from the fifteen minute averages should be counted as only two events, and that the last event, which consists of four sampling intervals, one of which is below the blocking threshold, is a false alarm.

4.2 Event Detection from Smoothed Data

We prefer to detect city pair events by smoothing the raw binary data $\{Y_t\}$ to estimate $\{\mu_t\}$ when there is evidence that $\mu_t > 0$. Smoothing over abrupt starts and stops is avoided by delaying smoothing until an observation of $Y_t = 1$ is obtained, and then stopping smoothing before a "long enough" string of $Y_t = 0$. The estimated blocking probability during periods with no smoothing is zero. Hence, only the periods with smoothing can contain reported city pair events.

We smooth with a running weighted logistic regression of Y_t on t with weights determined according to a symmetric gaussian kernel of fixed width. (See Hastie and Tibshirani (1986) for a discussion of binary smoothing.) Fixed width kernels are used to control how far in time a call that influences $\hat{\mu}_t$ can be from t. The periods of smoothing are the only candidates for reported city pair events, because they are the only periods with non-zero estimates of μ_t. Each smooth is *sliced* at height π and the starts and stops of the pieces that exceed π for at least δ hours are recorded as city pair events at thresholds (π, δ) (see Figure 3). Slicing at every possible (π, δ) would preserve all the information in a smooth, because a curve can be deduced from its threshold crossings, but that is unnecessary. Just as storing the smooth at only a subset of observation times and interpolating for intermediate times is often adequate for estimating a curve, storing slices for only some thresholds is often adequate for detecting events and estimating reliability parameters. Storing the slices takes less space than storing the smooth if sampling is rapid and only a few thresholds are monitored, because only the start, stop, and minimum, maximum and average of $\hat{\mu}_t$ during the reported event are kept for each slice. In our application, 23 values of (π, δ) were monitored (see Figure 5), and storing the start, stop, average and peak estimated blocking probabilities, and a few other statistics for the slices requires less than 0.1% of the space that would be required to store all the smooths at each observed time t.

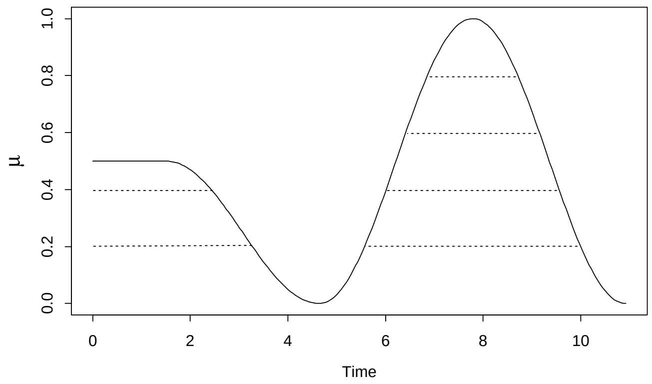

Figure 3: An artificial μ_t and its slices, shown as dotted line segments, at blocking thresholds $\pi = (.2, .4, .6, .8)$ and duration threshold $\delta = 1$. At thresholds $(\pi, \delta) = (.2, 2)$ there would be two city pair events starting at $t = 0$ and $t = 5.5$, and at thresholds $(\pi, \delta) = (.4, 4)$ there would be only one city pair event starting at $t = 5.8$ Note that μ_t can be re-constructed if the starts and stops of the slices are known for all thresholds.

4.3 Bridging Small Gaps Between Events

In simple cases, a slice at thresholds (π, δ) defines a reported city pair event at thresholds (π, δ). That is, one slice corresponds to one reported event. Sometimes, counting each slice as one reported event is undesirable, though. If μ_t is just a little above π for a long time, then sampling variability is likely to cause $\hat{\mu}_t$ to wander above and below π producing several shorter slices above π instead of just one long slice. When the number of reported events is used in competitive comparisons, over-counting is undesirable. We could avoid overcounting by re-defining a slice to allow the estimated blocking fraction to fall below π for a short time (see Smith (1986, 1989), for example). In our application, however, event counts are sensitive to the choice of an allowable gap. Besides, because the "false" gaps are a consequence of statistical variability in $\hat{\mu}_t$, the length of gap that is ignored should depend on the uncertainty about μ_t rather than on an arbitrary time constant. Therefore, we join or *bridge* slices when a pointwise, upper 95% one-sided confidence interval for μ_t exceeds π throughout the gap between the slices. We further require that at least one bridged slice meet the

(π, δ) thresholds to avoid introducing a city pair event where none would be reported otherwise. This requirement is not necessary, but in our application engineers were unwilling to incur more false alarms.

More precisely, suppose there are k consecutive slices for a city pair, each of which exceeds a blocking threshold π. Suppose the slices start at times $\hat{S}_1, \ldots, \hat{S}_k$, have durations $\hat{D}_1, \ldots, \hat{D}_k$, and the gaps between them have lengths $\hat{G}_1, \ldots, \hat{G}_{k-1}$. Then the k slices are *bridged* and counted as one city pair event with start $\hat{S}_1$ and duration $\hat{D}_{total} = \sum_{i=1}^{k} \hat{D}_i + \sum_{i=1}^{k-1} \hat{G}_i$ if

1. $\hat{\mu}_{it} \geq \pi$ for $\hat{S}_i \leq t \leq \hat{S}_i + \hat{D}_i$, $i = 1, \ldots, k$, and $\hat{D}_i \geq \delta$ for at least one i, $1 \leq i \leq k$;
2. a one-sided, upper 95% pointwise confidence limit for μ_t exceeds π for $\hat{S}_1 \leq t \leq \hat{S}_1 + \hat{D}_{total}$.

Condition (1) ensures that at least one of the slices qualifies as a city pair event without bridging. Condition (2) states that we are uncertain that the city pair event stopped during the gaps. Slices above the blocking threshold that are far apart in time are typically not bridged, because the upper confidence limit drops below π during long gaps.

Figure 4 shows the city pair events at thresholds of (.5, one hour) for the data shown in Figure 2. Five events are found using fifteen minute blocking averages, but only two are found using smoothing and bridging. The first city pair event found by smoothing starts earlier than the first city pair event found with 15 minute averages, and it includes bridging. The fifth city pair event found with 15 minute averages is not included among the city pair events found by smoothing. Note that Figure 4 suggests that counting events at just one (π, δ) may be unwise, because subthreshold problems of nearly the same severity continued after the first event ended.

To summarize, Table 1 shows the steps of our procedure for detecting network events from binary outcomes of test calls.

Table 1: Steps for detecting network events from test calls.

- For each city pair, find periods of positive blocking probability that might qualify as city pair events
  After a string of $Y_t = 0$, find the first $Y_t = 1$.
- Estimate the blocking probability μ_t during a period of positive blocking
  Smooth the Y_t's after the first $Y_t = 1$ until just before a string of $Y_t = 0$.
- Find periods with μ_t above the blocking threshold π
  Slice through the estimated $\hat{\mu}_t$.
  Bridge gaps between slices if an upper confidence limit on μ_t stays above π during the gap.
  Keep bridged slices with total length longer than δ.
- Report bridged slices as a city pair event.
- Report unions of reported city pair events as network events.

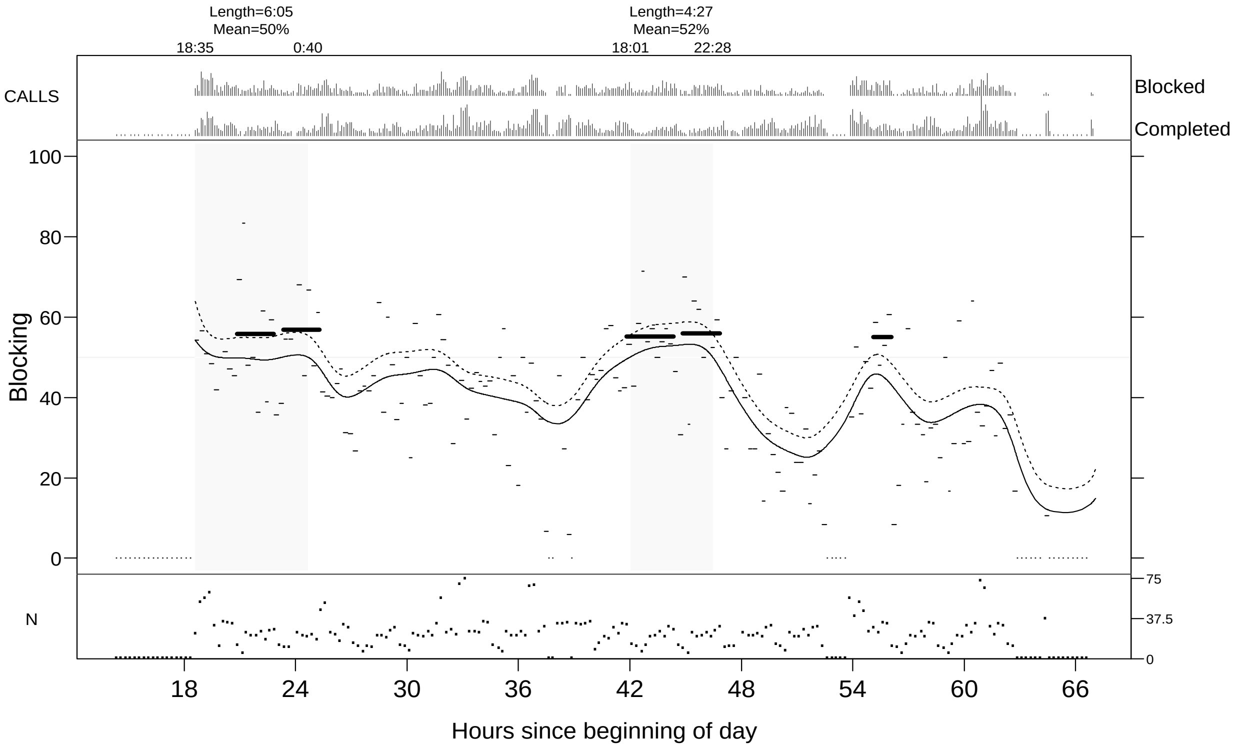

Figure 4: City pair events at thresholds(50%, one hour) for the data shown in Figure 2. The solid curve shows $\hat{\mu}_t$, and the dotted gives a pointwise one-sided 95% upper confidence limit for μ_t. The shaded vertical strips highlight reported events. The start, stop, duration and average blocking probability $\bar{\mu}$ during a reported city pair event are printed above the plot, where $\bar{\mu} = \sum_{i=1}^{M}(\hat{\mu}_{t_i} + \hat{\mu}_{t_{i-1}})(t_i - t_{i-1})/(t_M - t_1)$, M is the number of test calls from the start to the stop of the event and t_i is the time of the i^{th} call.

4.4 Computations

Monitoring generates a lot of data. For each call in our application, the type of sampling (screening or rapid), city pair names, date and time, type of network used, and whether the call was complete or blocked is recorded. In twelve months, the two networks considered in this paper generated hundreds of millions of calls.

All computing was done in version 3 of the S language (Becker, Chambers and Wilks, 1988). Data are saved by city pair in compressed files outside of S, and S functions process the data by

1. calling Unix tools to uncompress slightly more than a month of data for a set of city pairs (a little of the previous month's and following month's data is added to the current month to avoid prematurely cutting off city pair events that span the end of a month),
2. calling Unix tools such as grep, awk, and sort to extract and edit data from files outside S in a pipeline,
3. scanning the edited data into S and then smoothing. City pair events are detected at several threshold pairs (π, δ), postscript files of plots like Figure 4 are written, and information about the detected city pair events is saved in S objects.

Other S functions process the information about reported city pair events to find network events, estimate reliability parameters, and compute confidence intervals.

Note that we do not keep all the data in S or compute with it all at once. Instead, we compute for a manageable set of data and save only the summaries needed for reporting events and estimating network reliability.

5 Reliability Parameters and Estimators

Given sets of reported events for several blocking and duration thresholds for two networks over a year, an obvious question is which network should be judged more reliable? In line with the probability model described in Section 2, we assume that events happen to each network according to a random process, the duration of an event and blocking during an event are random, and simple summaries of these processes and distributions can be used to compare the reliabilities of competing networks. For example, for a threshold pair (π, δ), the mean number $E(N)$ of events that occur in a year, the mean duration $E(D)$ (in hours) of an event, and the mean total time $E(T) = E(N)E(D|N)$ spent in events in a year are parameters that describe network reliability. We also consider a measure of *impairment* $\lambda(\pi, \delta)$ defined as the mean number of events per year divided by the mean event-free hours per year:

$$\lambda(\pi, \delta) = E(N)/[365 \times 24 - E(T)].$$

"Event" might refer to a city pair event or a network event, but city pair events are so rare that we consider only network events. Also, we take $E(D|N) = E(D)$, because there is no evidence that the duration and number of events at (π, δ) are related in our application. For example, a plot (not shown) of reported duration against time from the end of the previous reported event shows no pattern.

For a threshold pair (π, δ), the number of reported events $\hat{N}$ at (π, δ) estimates $E(N)$, their total reported duration $\hat{T} = \sum_i \hat{d}_i$ estimates $E(T)$ and their average reported duration $\hat{N}^{-1}\hat{T}$ estimates $E(D)$, where $\hat{d}_i$ is the reported duration of reported network event i in hours. Also, $\hat{\lambda}(\pi, \delta) = \hat{N}/(365 \times 24 - \hat{T})$ estimates $\lambda(\pi, \delta)$.

These estimators are simple, but likely to be biased downwards, because events are more likely to be missed than wrongly counted and durations are more likely to be underestimated than overestimated in our application. For example, the first reported city pair event in Figure 4 starts with the onset of rapid sampling at 18:35, but μ_t might have crossed the threshold $\pi = .5$ earlier. It might have crossed after the previous screening call at 18:20 but before 18:35, because the city pair was not monitored for those 15 minutes. Or it might have started even earlier than 18:20, if by chance the screening call at 18:20 completed. In either case, the early part of the event is not monitored and its duration cannot be included in the reported duration. If enough of the event is not monitored, then the city pair event may not be reported at all. On the other hand, sampling bias affects only the beginning of an event, not the end, because the end of an event typically occurs during rapid sampling.

Sections 5.1 and 5.2 show how to correct the estimates of mean duration and mean number of events for possibly missing the beginning of events. Section 5.1 first shows which reported durations $\hat{d}$ are likely given an (unobserved) event duration of D and then shows how to find the conditional density of interest, which is the conditional density of D given a reported duration $\hat{d}$. For example, a reported duration of $\hat{d}$ corresponds to a mean event duration of $E(D|\hat{d})$, and the probability that a reported duration $\hat{d} < \delta$ corresponds to an event with duration $D \geq \delta$ is $P(D \geq \delta|\hat{d})$. Section 5.2 then shows how to correct estimated reliability parameters for bias. Readers who are not concerned with the details of bias correction can skip Sections 5.1 and 5.2 and turn directly to the example in Section 5.3.

5.1 Densities Related to Event Duration

First, consider city pair events instead of network events. Each city pair event has a start t_0 and an end t_1. The *duration* $D = t_1 - t_0$ of an event is composed of two parts: the *monitored duration* d during which rapid sampling occurs and the *unmonitored duration* $D - d$. The monitored duration may be zero, if there is no rapid sampling during the event. In any case, the *reported duration* $\hat{d} = \hat{t}_1 - \hat{t}_0$ estimates the monitored duration d rather than D. Our goal is to add an estimate of $D - d$ to $\hat{d}$ when the beginning of an event may have been missed.

Suppose $D > d$, so rapid sampling is not in progress when the event starts. Then the unmonitored duration $D - d$ depends on (i) the number of screening calls that complete during the event before the first failed screening call (if any) during the event and (ii) the time between μ_t crossing π and the first screening call during the event. Because events are unrelated to sampling, the latter time is uniformly distributed over the sampling interval (.25 hours in our application).

Thus, an event with duration D is monitored for $0 < d \leq D$ hours if and only if

(i) the first $M = \lfloor r(D-d) \rfloor$ screening calls during the event complete and screening call $M+1$ is blocked, where $\lfloor x \rfloor$ is the largest integer in x, and

(ii) the time until the first screening call during the event is $D - r^{-1}M - d$.

When μ_t is constant, the rapid sampling rate is r calls per hour, and the event starts before rapid sampling,

$$f(d|D,\mu) = r\mu(1-\mu)^M \qquad if \quad D \geq d > 0 \quad and \quad t_0 < t_R. \tag{2}$$

This density does not integrate to one, because there is a positive probability that every screening call during the event completes, and, hence, $d = 0$. The higher μ, the less likely it is that all screening calls during the event complete, and the more likely it is that the first screening call during the event is blocked, so that the monitored duration d is close to the event duration D.

A more general formula for $f(d|D,\mu)$ that allows μ_t to vary with t can be derived; all that is needed is the value of μ_t at the time of each screening call. The more general formula is not used here, however, because we are interested in blocking thresholds so high that the missed portion of an event is short.

Following the convention of labeling densities by their arguments, the density $f(D|d,\mu)$ of interest can be obtained from equation (2) and the unconditional density $f(D|\theta)$ of event duration using Bayes theorem. (In the notation of model (1), $f(D|\theta)$ is the density of F_1.) In our application, we take $f(D|\theta)$ to be approximately lognormal with parameter $\theta = (\theta_1, \theta_2)$ that is independent of the parameters that govern the distribution of μ_t. So,

$$f(D|d,\mu,\theta) = \frac{f(d|D,\mu)f(D|\theta)}{\int f(d|x,\mu)f(x|\theta)dx} \tag{3}$$

if the event starts before rapid sampling.

Equation (3) specifies which event durations D are likely when the monitored duration is d and the event starts before rapid sampling. In our application rapid sampling is fast enough that the end of an event is usually estimated well. Therefore, we also use equation (3) for $f(D|\hat{d},\mu)$, conditioning on reported duration rather than monitored duration.

So far, we have considered the duration of a city pair event. If only one city pair is involved in a network event, then equation (3) applies to the network event. If two or more city pairs are involved in a network event, then they are likely to be related because city pair events are so short and so rare that overlapping independent events are unlikely. Moreover, the earliest city pair in a network event usually has the highest blocking probability, because the secondary congestion due to re-routing traffic is usually less than the primary congestion. Thus, equation (3) for the city pair event with the highest blocking probability in the network event applies to the network event.

Because $E(D|\hat{d},\mu,\theta)$ describes the average event duration corresponding to a reported duration of $\hat{d}$, the duration of a reported event can now be corrected for bias in the steps shown in Table 2.

Table 2: Steps for correcting the reported duration of an event for the bias introduced by two stage sampling.

• Compute the reported duration $\hat{d}$ as the difference of the reported stop and start of the event.
• Estimate the parameter θ of the marginal density $f(d\|\theta)$ of monitored duration d, as in Section 5.2.
• Compute the density of d given event duration D and blocking probability $\hat{\mu}$ as in equation (2).
• Use Bayes Theorem to compute the density $f(D\|d = \hat{d}, \hat{\mu}, \hat{\theta})$ of interest.
• Compute the bias-corrected estimated duration $E(D\|\hat{d}, \hat{\mu}, \hat{\theta})$ corresponding to the reported duration.
• Compute $P(D \geq \delta'\|\hat{d}, \hat{\mu}, \hat{\theta})$, the probability that the reported event was longer than δ' hours.

5.2 Bias Corrected Reliability Estimates

The bias corrected estimators of $E(D)$ and $E(N)$ depend on the unknown parameter θ of the unconditional density $f(D|\theta)$ of monitored duration. In our application, we take $f(D|\theta)$ to be lognormal and estimate $\theta = (\theta_1, \theta_2)$ from the reported durations at the thresholds (π_0, δ_0), where π_0 is the smallest π monitored and δ_0 is the smallest δ monitored. These thresholds are used because estimates based on more events are likely to be better, and more events occur at lower thresholds. (Note that the algorithm for detecting events must be run at (π_0, δ_0).)

The estimated $\hat{\theta}$ is found by maximizing the log density of the $\hat{N}_0$ reported durations at (π_0, δ_0):

$$\begin{aligned} L(\theta) &= \sum_{i=1}^{\hat{N}_0} log(f(\hat{d}_i|\hat{\mu}_i, \theta)) \\ &= \sum_{i=1}^{\hat{N}_0} log\left(\int f(\hat{d}_i|\, D_i, \hat{\mu}_i, \theta) f(D_i|\hat{\mu}_i, \theta)\, dD_i\right) \\ &= \sum_{i=1}^{\hat{N}_0} log\left(\int f(\hat{d}_i|\, \hat{\mu}_i, D_i) f(D_i|\, \theta)\, dD_i\right), \end{aligned}$$

where $\hat{\mu}_i$ is the average reported blocking probability, $\hat{d}_i$ is the reported duration of the i^{th} reported event at (π_0, δ_0), and $f(\hat{d}_i|\hat{\mu}_i, D_i)$ is given by (2). The maximizing $\hat{\theta}$ can be found by replacing the integral with a sum over a fine grid, for example. Then the estimated density for the duration of an event with reported duration $\hat{d}$ and reported average blocking $\hat{\mu}$ is $f(D|\hat{d}, \hat{\mu}, \hat{\theta})$, where f as in equation (3).

Bias corrected estimates $\hat{N}_{cor}$ and $\hat{T}_{cor}$ of $E(N)$ and $E(T)$ at thresholds (π, δ) for the past year, say, can now be computed from the lists of reported events for the past year as follows. First, note that reported events that are shorter than δ but have reported blocking above π must be considered, because there is a chance that these shorter reported durations would exceed δ after bias correction.

1. Find the $\hat{N}_R$ reported events at (π, δ) that started during rapid sampling and their total reported

duration $\hat{T}_R$. These events have no unmonitored duration, so $\hat{N}_R$ and $\hat{T}_R$ are not corrected for bias.

2. Find the $\hat{N}_S$ reported events at (π, δ) that started with a screening call and their reported durations $\hat{d}_1, \ldots, \hat{d}_{\hat{N}_S}$. The reported durations of these events may be too short. The bias corrected estimated duration of the event corresponding to $\hat{d}_s$ is $E(D|\hat{d}_s, \hat{\mu}_s, \hat{\theta}) = \int Df(D|\hat{d}_s, \hat{\mu}_s, \hat{\theta})dD$.

3. Suppose that thresholds $(\pi, \delta_1), \ldots, (\pi, \delta_K)$ were also monitored, where $\delta > \delta_1 > \ldots > \delta_K$.

   (a) Find the reported events at thresholds (π, δ_1) with reported duration shorter than δ. These reported events were not included in Step 2. Suppose that there are $\hat{M}_1$ such reported events. For each, compute $f(D|\hat{d}_{1i}, \hat{\mu}_{1i}, \hat{\theta})$ where $\hat{\mu}_{1i}$ is the average reported blocking during the i^{th} such reported event and $\hat{d}_{1i}$ is the reported duration for the i^{th} such reported event. Then the estimated probability that the duration D of the event exceeds δ is

   $$P(D > \delta|\hat{d}_{1i}, \hat{\mu}_{1i}, \hat{\theta}) = \int_{D>\delta} f(D|\hat{d}_{1i}, \hat{\mu}_{1i}, \hat{\theta})dD$$

   and the bias corrected estimated duration of the event given that it is longer than δ is

   $$E(D|D > \delta, \hat{d}_{1i}, \hat{\mu}_{1i}, \hat{\theta}) = \frac{\int_{D>\delta} Df(D|\hat{d}_{1i}, \hat{\mu}_{1i}, \hat{\theta})dD}{P(D > \delta|\hat{d}_{1i}, \hat{\mu}_{1i}, \hat{\theta})}.$$

   (b) If $K > 1$, for k in $2, \ldots, K$, find the $\hat{M}_k$ reported events at thresholds (π, δ_k) with reported duration less than δ_{k-1}, and correct their reported durations as in step (3a).

4. When δ is the smallest duration threshold paired with π, Step 3 cannot be implemented. Instead, replace π with the smallest π' larger than π and follow Steps 3a and 3b with (π', δ) replacing (π, δ). (Events at (π', δ) would also have been reported at (π, δ), if that combination of thresholds had been monitored.)

Finally, the bias corrected estimator of $E(N)$ is

$$\hat{N}_{cor} = \hat{N}_R + \hat{N}_S + \sum_{k=1}^{K} \sum_{i=1}^{\hat{M}_{ki}} P(D \geq \delta|\hat{d}_{ki}, \hat{\mu}_{ki}, \hat{\theta}),$$

and the bias corrected estimator of $E(T)$ is

$$\hat{T}_{cor} = \hat{T}_R + \sum_{i=1}^{\hat{N}_S} E(D|D > \delta, \hat{d}_i, \hat{\mu}_i, \hat{\theta}) + \sum_{k=1}^{K} \sum_{i=1}^{\hat{M}_k} E(D|D > \delta, \hat{d}_{ki}, \hat{\mu}_{ki}, \hat{\theta}) P(D > \delta|\hat{d}_{ki}, \hat{\mu}_{ki}, \hat{\theta}).$$

In effect, each reported event in the first two terms on the right exceeds (π, δ) and hence is scored as 1. Each reported event in the double summation on the right could exceed (π, δ) after bias correction and hence is scored as the estimated probability that the event duration exceeds δ. Together, $\hat{N}_{cor}$ and $\hat{T}_{cor}$ define $\hat{\lambda}_{cor}(\pi, \delta)$. Also, the bias corrected estimate of average duration at (π, δ) is $\hat{T}_{cor}/\hat{N}_{cor}$.

Note that we do not correct for missing events completely; that is, for failing to report events whose estimated duration is shorter than the shortest duration threshold δ_0 monitored. It is possible to include the expected number of events with $D \geq \delta$ and $\hat{d}$ less than the minimum duration threshold δ_0, but in our application the probability of such events at the δ's of interest is so low that the correction is unimportant. This is fortunate, because there was a strong reluctance to correct for events for which there was no reported evidence. But it does mean that $\hat{N}$ and $\hat{T}$ cannot be corrected for sampling bias at the smallest monitored duration threshold and that δ_0 should be chosen in light of its ability to improve estimates at higher thresholds.

Finally, Table 3 summarizes the steps for estimating reliability parameters.

Table 3: The major steps to estimate reliability parameters at thresholds (π, δ) for a fixed period, correcting for under-reporting the duration of reported events.

1. Find the reported events at a set of thresholds $\{(\pi_j, \delta_j)\}$.
2. Estimate the number $E(N)$ of events at (π, δ) as the sum of
   - the number $\hat{N}$ of reported events at (π, δ) and
   - $\sum_A P(D > \delta|\hat{d}_i)$ where A is the set of reported events at (π, δ') for $\delta' \leq \hat{d}_i < \delta$.
3. Estimate the mean duration $E(D)$ at (π, δ) as the sum of
   - the reported duration $\hat{d}_i$ of each event that started after rapid sampling,
   - $E(D|\hat{d}_i)$ for each reported event i at (π, δ) that started before rapid sampling, and
   - $E(D > \delta|\hat{d}_i)P(D > \delta|\hat{d}_i)$ for each reported event i shorter than δ

   and divide their sum by the estimated number of events from Step 2.

5.3 An Example

Returning to our data, the impairment parameter $\lambda(\pi, \delta) = E(N)/(365 * 24 - E(T))$ was estimated at 23 threshold pairs for each of two networks, network A and network B, using data from a twelve month period. Bias correction increased the estimated impairment $\hat{\lambda}(\pi, \delta)$ of each network, but sometimes only slightly. The median, third quartile, and maximum percentage increases at the 23 threshold pairs were 5%, 16% and 38% for network A and 4%, 8% and 14% for network B. The corrections are smaller for network B because its events tend to have higher blocking, and the larger the blocking probability during an event, the smaller the chance that a screening call during the event completes, and the smaller the correction for sampling bias.

Figure 5 shows the ratio of uncorrected network impairment estimates and the ratio of bias-corrected estimates for the two networks. (Recall that the objective in our application was to compare the reliability of networks A and B.) The bias-corrected ratios are smoother, because some of the events reported at a lower duration are counted fractionally as $P(D \geq \delta|\hat{d}_i, \hat{\mu}_i)$ in the bias-corrected estimate, lessening the distinction between thresholds.

We can also estimate $\lambda(\pi, \delta)$ for arbitrary (π, δ). To do this, assume that $log(\lambda(\pi, \delta))$ is locally linear in (π, δ) and that $variance(\hat{\lambda}(\pi, \delta)) \propto \lambda(\pi, \delta)$ and fit a locally linear model with weights

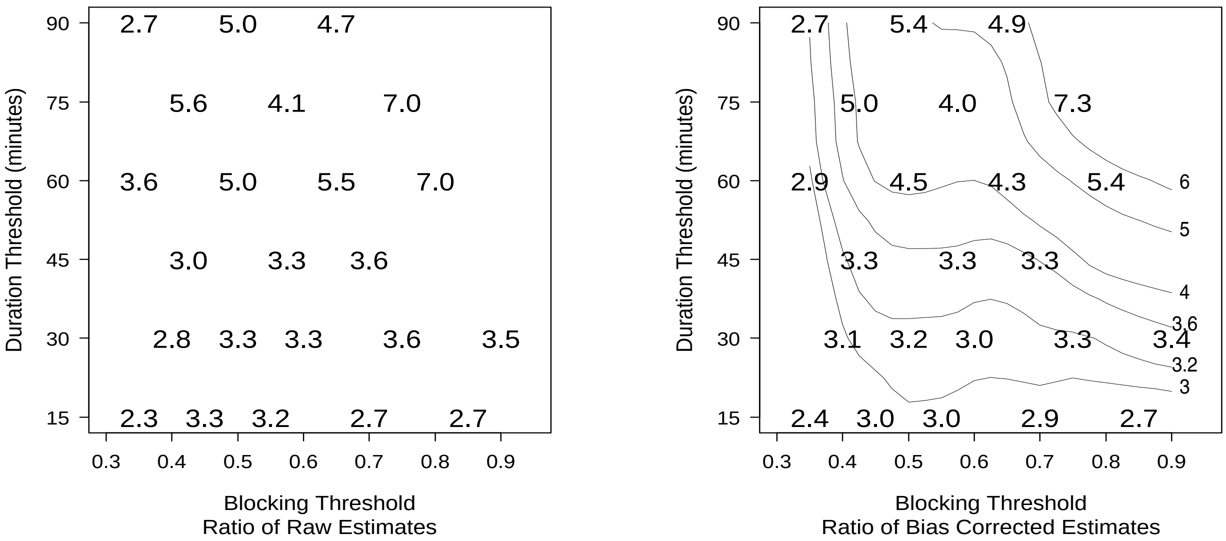

Figure 5: The left panel shows the ratio of raw estimates $\hat{\lambda}(\pi, \delta)$ for network B relative to network A at the threshold pairs monitored in our application. The right panel shows the ratio of bias-corrected estimates $\hat{\lambda}_{cor}(\pi, \delta)$ as numbers and the ratio of smooth bias-corrected estimates $\hat{\lambda}_S(\pi, \delta)$ as contours.

proportional to $\lambda(\pi, \delta)$. For example, fit an unweighted degree 1 loess model to $log(\hat{\lambda}(\pi, \delta))$ as predictors, and then re-fit the loess model using the fitted values from the first loess fit as weights in the second call to loess. (See Cleveland and Devlin (1988) for a discussion of loess in two dimensions, Cleveland and Grosse (1991) for a discussion of the algorithm for computing loess, and Chambers and Hastie (1991) for a discussion of the S function "loess".) Only a few minutes are required to compute $\hat{\theta}$, $f(D|\hat{d}_i, \hat{\mu}_i, \hat{\theta})$ and the *smooth bias-corrected estimate* $\hat{\lambda}_S(\pi, \delta)$ after the lists of reported pair events are found.

Figure 5 shows the ratio of smooth, bias-corrected estimates for our data. Network B's impairment estimates are at least twice as large as Network A's at all thresholds considered. Roughly speaking, the ratio becomes more unfavorable for B as δ increases for a fixed π or as π increases for a fixed δ, especially for $\pi > .6$.

6 Results of a Simulation

There are several sources of variability in our estimators: the random waiting time until an event starts, the random duration of an event, the random blocking probability $\{\mu_t\}$ during an event, the random outcome of a screening call, and the random outcomes of calls during rapid sampling. Add to this the effects of smoothing, bridging, and bias correction, and the end result is a procedure for reporting events and estimating their properties that is difficult to analyze mathematically.

Therefore, to assess its performance we resort to simulation in the statistical language S (Becker, Chambers and Wilks, 1988).

The simulation is structured as follows. The number of events at thresholds ($\pi = .45, \delta = .5\,hours$) in a simulated year follows a negative binomial distribution with mean 50 and variance 100. In 1000 simulated years, there were a total of 50,170 events, with a minimum of 26 events, median of 50, quartiles of 43 and 56, and maximum of 87 events per year. Each event has a random duration (in hours) that follows a lognormal(.5, .5) distribution conditioned to be at least .5, and a random, constant blocking probability that follows a uniform$(.45, 1)$ distribution. The minimum, median and maximum of the generated durations were .50, 1.66 and 12.61 hours, and the quartiles were 1.19 and 2.31 hours. Each event starts at a random time t_0 that is uniformly distributed over the sampling interval $(0, .25$ hours). The outcomes of screening calls follow a Bernoulli(μ) distribution. For each interval with rapid sampling, 25 Bernoulli outcomes, equally spaced in time were generated. An outcome at time t has parameter μ (the simulated blocking probability) if the event is in progress at time t and parameter 0 if the event has ended by time t. The sequence of binary outcomes is padded with zeros before the start of the first screening call and after the end of the event, to simulate a process in which the blocking probability is usually zero. The data generated before, during and after one event are then fed to our algorithm for detecting events. Events were reported at 15 thresholds; the lowest thresholds were $(\pi_0, \delta_0) = (.35, .25)$. The reported events at thresholds (π_0, δ_0) for a simulated year (which might be different from the simulated events for the year) were used to estimate the lognormal parameter θ for the year. The estimated $\hat{\theta}$ was then used to correct the total number of events per year and the durations of reported events for sampling bias.

At any threshold, a simulated event could correspond to 0 reported events (if the event was not detected), 1 reported event, or more than 1 reported event (if the detection algorithm failed to recognize that only one continuous event was in progress and split the one event into two or more pieces). For each simulated event, the simulated blocking probability μ, simulated time from the start of the event to the next screening call, and the simulated duration D were recorded. For each simulated event, the total number of reported events (which was more than 1 in only a few iterations and only at short duration thresholds), an indicator for bridging, and the average reported blocking probability and the reported duration (if the event was reported) were recorded. Finally, the mean number of events per year and mean time spent in events per year were estimated at fifteen thresholds for each of the 1000 simulated years.

Table 4 summarizes the ability of the algorithm to detect events. At low duration thresholds, about 95% of the simulated events were detected, but the percentage decreases to only 75% at high thresholds. The higher detection at low duration thresholds is an artifact, because all simulated events were at least .5 hours long, making it easier to detect events at the $\delta = .25$ threshold. There were no more than 5% false alarms (non-events reported as events) at all thresholds except (π, δ)=(.65, .25), for which there were 13% false alarms. It is not clear why the false alarm probability is higher at (.65, .25); for example, average reported blocking for false alarms at (.65, .25) is not much closer to .65 than average reported blocking for false alarms at (.6, .5) is to .6.

Table 4: Performance of the event detection algorithm. N is the number of simulated events with blocking higher than π lasting longer than δ hours. % detect is the percentage of simulated events detected at the specified thresholds. % false is the percentage of false alarms or detected events for which the simulated parameters did not exceed the specified thresholds. Bias refers to the mean estimated difference of the reported statistic and the value for the simulated event; root mean squared error (rmse) is the square root of the mean squared difference of estimated and simulated value for the event.

(π, δ)	N	% bridged	% detected	% false	$\hat{\mu}$ % bias, rmse	$\hat{D}$ (mins) bias, rmse	$\hat{D}_{cor}$ (mins) bias, rmse
.35, .25	50,171	3	97	0	.5, 3.8	-14, 19	-.4, 12
.45, .25	50,171	12	94	0	.8, 3.8	-16, 22	-.3, 12
.55, .25	41,073	18	95	9	1.4, 4.4	-19, 33	-.2, 11
.65, .25	32,099	22	96	13	1.8, 4.6	-21, 37	-.2, 9
.50, .50	45,646	10	88	3	.9, 3.6	-16, 24	.2, 11
.60, .50	36,613	13	88	5	1.1, 3.6	-16, 27	.1, 9
.45, .75	47,629	5	81	0	.6, 3.3	-14, 19	.5, 11
.55, .75	38,976	8	81	2	.8, 3.3	-14, 23	.4, 9
.65, .75	30,471	11	81	3	.9, 3.1	-14, 24	.2, 8
.50, 1.00	38,599	5	77	1	.6, 3.0	-13, 20	.7, 10
.60, 1.00	30,980	6	77	2	.7, 2.9	-13, 21	.4, 8
.45, 1.25	35,895	3	74	0	.5, 2.9	-13, 18	1.0, 10
.55, 1.25	29,351	4	75	1	.6, 2.8	-13, 19	.7, 9
.35, 1.50	29,046	0	79	0	.3, 2.7	-12, 17	1.3, 10
.50, 1.50	26,424	2	73	1	.5, 2.7	-13, 18	.9, 9

Table 4 also summarizes the ability of the event detection algorithm to characterize detected events accurately. As expected, the blocking probability is estimated well, but event duration is underestimated. The mean difference between simulated and reported blocking probability is positive but no more than .018 at all thresholds monitored. The bias is positive because detected events with blocking below π are not included in the estimate of μ. Generally, event duration is underestimated by about 15 minutes, except for the threshold (.65, .25) where it is underestimated by 21 minutes. Correcting reported durations for sampling bias, as described in Section 5.1 reduces the bias to less than 1 minute at duration thresholds shorter than 1.25 hours and to no more than 1.3 minutes for all other duration thresholds. Thus, the bias correction is both necessary and effective.

Table 5 shows the performance of the estimates of number of events per year and total hours spent in events per year. Without bias correction, the number of events per year is underestimated by about 25% at higher thresholds. The bias corrected estimate is off by at most 10% on average. In all cases, the bias corrected estimate is no worse than the uncorrected estimate (to which it is identical for $\delta = .25$), and in some cases bias correction reduces the bias by a factor of 5 or more and the rmse by a factor of 3. The results for the estimated total time spent in events are similar, with biases ranging from 6% to 27% for uncorrected total event time (most are at least 15%) and substantially smaller biases and rmses for bias corrected estimates at most thresholds. The worst performance occurs at (.65, .25) and (.65, .75), the latter because it includes information from the (.65, .25) reported events. At other thresholds, the bias corrected estimates are better than the raw estimates – sometimes much better.

Table 5: Performance of estimated reliability parameters for 1000 simulated years. N and T are the number of simulated events and their total simulated duration for the year. $\hat{N}$ and $\hat{T}$ are the number of reported events and their total reported duration for a simulated year; $\hat{N}_{cor}$ and $\hat{T}_{cor}$ are the estimated number of events and estimated total duration after correction for sampling bias. The number of events is not corrected for length bias at the lowest duration threshold. Here bias is the mean of $(estimate/parameter - 1) \times 100\%$; rmse is the mean of $(estimate/parameter - 1)^2 \times 100\%$.

(π, δ)	N mean, s.d.	$\hat{N}$ %bias, rmse	$\hat{N}_{cor}$ %bias, rmse	T (hours) mean, s.d.	$\hat{T}$ (hours) %bias, rmse	$\hat{T}_{cor}$ (hours) %bias, rmse
.35, .25	50, 10.1	-3, 4	-3, 4	94, 20	-14, 14	-2, 3
.45, .25	50, 10.1	-6, 6	-6, 6	94, 20	-16, 17	-3, 4
.55, .25	41, 8.6	5, 8	5, 8	77, 17	-9, 10	9, 11
.65, .25	32, 7.3	10, 13	10, 13	60, 15	-6, 9	14, 17
.50, .50	46, 9.3	-10, 11	-4, 7	86, 18	-16, 16	-0, 5
.60, .50	37, 8.0	-7, 10	-1, 6	69, 16	-14, 15	4, 8
.45, .75	48, 9.7	-19, 20	-7, 8	92, 20	-22, 22	-4, 5
.55, .75	39, 8.2	-17, 18	5, 8	76, 17	-19, 20	8, 11
.65, .75	30, 6.9	-16, 18	11, 14	59, 15	-18, 20	14, 17
.50, 1.00	39, 8.2	-22, 23	-4, 7	80, 18	-23, 24	-1, 5
.60, 1.00	31, 7.0	-22, 23	1, 8	64, 15	-22, 23	5, 9
.45, 1.25	36, 7.9	-26, 27	-10, 11	81, 18	-26, 27	-7, 9
.55, 1.25	29, 6.7	-25, 26	2, 8	66, 16	-25, 26	4, 8
.35, 1.50	29, 6.7	-21, 22	-3, 7	71, 17	-21, 22	-1, 5
.50, 1.50	26, 6.2	-26, 28	-5, 9	65, 16	-27, 28	-3, 7

7 Standard Errors and Confidence Intervals

Nonparametric confidence intervals for $E(N)$, $E(D)$ and $\lambda(\pi, \delta)$ at a particular threshold pair (π, δ) are obtained by a kind of partial bootstrapping. Bootstrapping must be partial because we cannot afford to bootstrap the raw data with hundreds of millions of observations. Fortunately, some sources of variability are negligible with so many observations, so re-sampling the raw data is not necessary. For example, in our application, rapid sampling is fast enough that the blocking probability during rapid sampling and the end of a reported event are estimated with little error. Therefore, the important sources of variability are the randomness in the onset of events and the onset of rapid sampling, not the error in estimating event properties once the event is monitored. So, it is enough to bootstrap the random process of generating events and the start of rapid sampling. The random process of generating observations during events after the onset of rapid sampling can be ignored.

We assume that the waiting times from the end of one event at the lowest thresholds (π_0, δ_0) to the start of the next event at (π_0, δ_0) are independent and identically distributed, and we re-sample these to simulate the distribution F_0 of time between events. Not having stored the blocking function $\hat{B} = \{\hat{\mu}_t\}$ during an event, we cannot simulate $\hat{B}$ by re-sampling the set of observed $\hat{B}_1, \hat{B}_2, \ldots$. But any event at the lowest thresholds has an associated list of nested events at all higher, monitored thresholds, as in Figure 3, and that "nest" approximates the blocking probability during the reported event. Therefore, re-sampling from the set of events at the lowest blocking and duration thresholds (π_0, δ_0) provides a natural way to re-sample events at all thresholds simultaneously. Also, re-sampling events at the lowest thresholds gives a sample of the binary variables that determine whether the event starts before or after the onset of rapid sampling.

If reliability parameters are estimated from a year of data, then bootstrapping proceeds by alternately sampling the times between reported events at the lowest thresholds (π_0, δ_0) and the reported durations at (π_0, δ_0) to form new year long histories. More precisely, start the bootstrap history by randomly choosing a W_1^* from the set of waiting times between reported events at (π_0, δ_0). Because the first year starts at an arbitrary time, take the time from January 1 of the first bootstrap year to the first event to be $W_1 = U\, W_1^*$ where U is a uniform(0,1) random variable. Next, choose the properties of the first bootstrapped event of the year by randomly choosing a reported event at thresholds (π_0, δ_0). This gives a bootstrapped duration of D_1 at (π_0, δ_0), an indicator for whether rapid sampling started before or after the onset of the bootstrapped event, and possibly a list of associated bootstrapped events (with possibly shorter durations) at higher thresholds. Next, choose the time W_2 from the end of the first bootstrapped event until the start of the second bootstrapped event by randomly choosing one of the waiting times between reported events at (π_0, δ_0). Choose the properties of the second bootstrapped event by randomly choosing one of the reported events at (π_0, δ_0), which again has an associated nest of events at higher thresholds. Continue sampling times-between-reported-events and reported events at (π_0, δ_0) with replacement until the total time sampled $\sum(W_i + D_i)$ is at least as long as the total number of bootstrapped years required. In our application, we bootstrapped 2000 years. This history is then chopped into year long pieces. If a bootstrapped waiting time (rather than a bootstrapped event) is in progress at the end of the bootstrapped year (midnight of December 31 of the bootstrapped year), then the bootstrapped year ends on schedule and the excess waiting time beyond midnight December 31 is used as the first waiting time for the next bootstrapped year. If a bootstrapped event is in progress at the end of the bootstrapped year, then the year is not ended until the bootstrapped event is over. This policy is consistent with current practice, but it does mean that bootstrapped years may differ slightly in length. Note that re-sampling observed periods such as weeks or years would be unsatisfactory, because events and waiting times do not respect calendar boundaries.

Once the bootstrapped events are generated, it is straightforward to compute estimates for each bootstrapped year for the thresholds (π, δ) of interest. The sample standard deviation of the bootstrapped, yearly values of $\hat{\lambda}_S(\pi, \delta)$ estimates the standard error of $\hat{\lambda}_S(\pi, \delta)$, for example. Lower and upper $1 - \alpha$ confidence limits can be obtained by taking the α and $1 - \alpha$ quantiles of the bootstrapped estimates. (Efron and Tibshirani (1993) discuss these and more sophisticated confidence intervals.)

The 90% confidence limits for the smooth, bias corrected network impairment parameter $\lambda_S(\pi, \delta)$ for our application are shown in Figure 6. These were obtained by taking 2000 bootstrap years for each network separately, taking the ratio of the bias corrected impairment estimates for the two networks for each bootstrap year, and then finding the .05 and .95 quantiles of the 2000 such ratios. For comparison, the upper and lower limits based on the bias-corrected estimates without smoothing $\hat{\lambda}_{cor}(\pi, \delta)$ are shown in Figure 7. The contours of the ratios $\hat{\lambda}_{cor}(\pi, \delta)$ without smoothing are noisier and less interpretable than the contours for the smoothed ratios $\hat{\lambda}_S(\pi, \delta)$, but the numerical values of the estimated ratios with and without smoothing are similar; the smoothed and unsmoothed ratios at any of the 23 thresholds are within 25% of each other. As expected, the confidence intervals based on $\hat{\lambda}_S(\pi, \delta)$ tend to be shorter, with the greatest reduction for π between .40 and .50.

8 Discussion

We have described a way to detect and characterize periods (events) in which a mean μ_t exceeds a specified level for a specified duration or longer. Events defined by severity and duration are of interest when cumulative exposure rather than instantaneous peak exposure is of concern. Our procedure has three major steps: from lists of reported events, (i) detect events, (ii) characterize detected events, (iii) estimate reliability parameters such as the expected number of events and expected duration of an event.

Starts of events are detected by finding a blocked call after a sequence of completed calls. Blocking probabilities during the period of nonzero blocking are then estimated by smoothing the outcomes of the subsequent test calls. Periods in which the smooth stays above a threshold π for at least δ hours, ignoring gaps if a pointwise confidence band stays above π throughout the gap, are reported as events. We note that our approach works by smoothing the raw data rather than averaging the raw data in fixed time intervals. Smoothing allows the duration of the event to be reported in the original time units rather than in aggregated time units.

Reported events are characterized by the average smooth blocking probability during the event and the reported duration (or time from the start to the stop of the event). In our application,

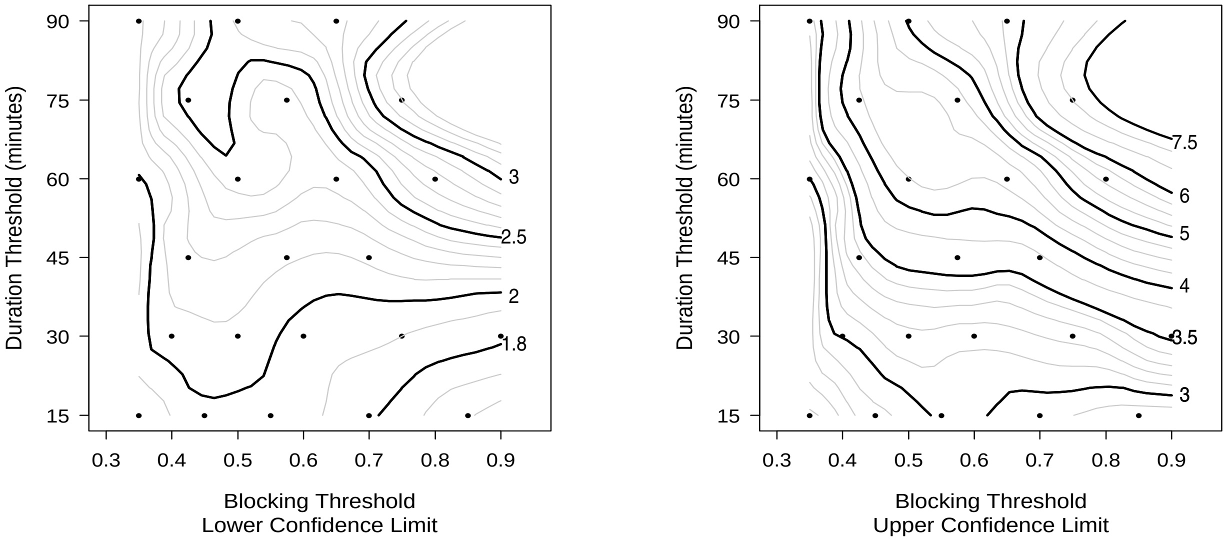

Figure 6: The left panel shows the lower two-sided 90% pointwise confidence limit for network B's impairment relative to network A's based on the bias-corrected, smooth estimates. The right panel shows the upper two-sided 90% pointwise confidence limit. Points show the thresholds used in Figure 5.

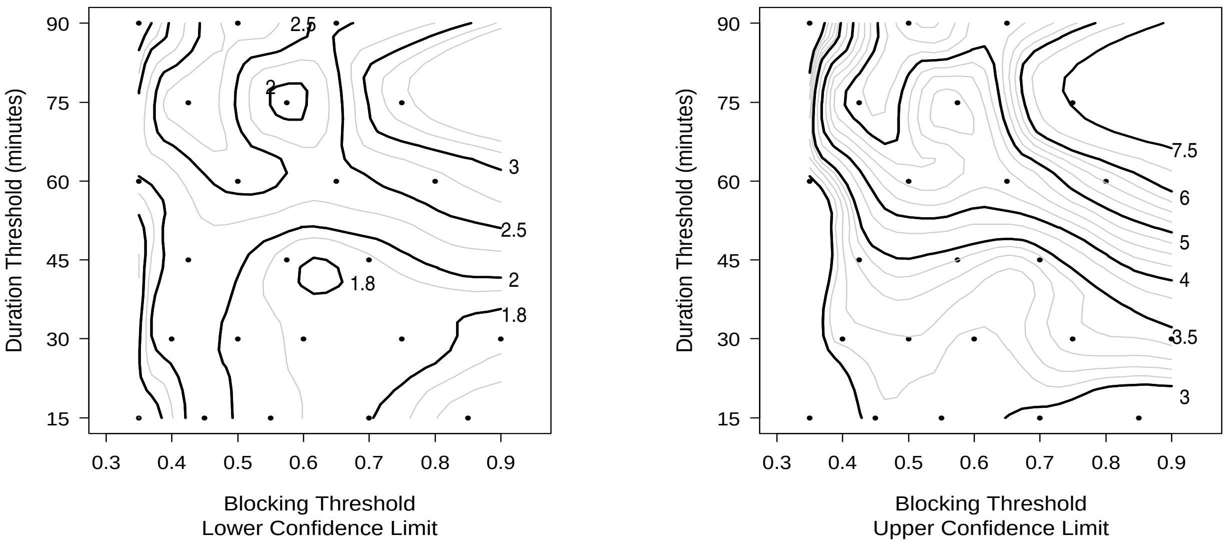

Figure 7: The left panel shows the lower two-sided 90% pointwise confidence limit for network B's impairment relative to network A's based on the bias-corrected estimates without smoothing; the right shows the upper 90% two-sided pointwise confidence limit. Points show the thresholds used in Figure 5.

reported durations are affected by sampling bias, because measurements are taken infrequently when $\hat{\mu}_t$ is close to zero and rapidly when $\hat{\mu}_t$ is high. We correct the reported duration for bias by estimating the density of event duration D given a reported duration $\hat{d}$ and then computing $E(D|\hat{d})$. Evidence from a simulation suggests that the sampling correction can be necessary and effective.

Lists of reported events at different thresholds can be used to compute point estimates and confidence intervals for network reliability parameters such as the expected number of events per year and the expected time spent in events per year. Partial bootstrapping from lists of events, rather than full bootstrapping from the raw data, is an efficient way to estimate standard errors with very large sets of data. We do not claim that our approach to finding events and estimating network reliability is optimal in a mathematical sense. But it is practical, based on sound statistical principles, and supported by simulations. Informally, it also balances statistical efficiency, computational efficiency and ease of interpretation.

Lists of reported events exclude periods in which $\hat{\mu}_t$ is below the blocking threshold π or above π for too short a time to quality as an event. Dichotomizing periods of non-zero $\hat{\mu}_t$ as either "reportable" or "not reportable" can lead to ignoring periods with subthreshold but nonetheless serious problems. These effects can be mitigated by reporting events at more than one threshold pair (π, δ). Smoothing reliability estimates as a function of severity and duration thresholds also suggests sensitivity of reliability estimates to the choice of thresholds.

Communications monitoring generates large sets of data, if not massive sets of data by current standards. Large sets of data have to be summarized to be manageable, but aggregating over a fixed unit of time to reduce computing costs can be statistically inefficient. Summarizing curves in time by thresholding rather than by averaging over time may preserve more of the information that is important for understanding the character and occurrence of events. More generally, "data-mining" and "knowledge discovery" tools are gaining widespread acceptance for analyzing massive sets of data. Our work shows that statisticians can also analyze large/massive sets of data with standard statistical tools like the bootstrap by ignoring sources of variability that are unimportant for the statistical goals.

Although the methods proposed in this paper were developed for communications reliability data, we believe that applications with severity and duration thresholds occur more generally. For example, high concentrations of ozone affect the health of young children, the elderly and the chronically ill. The more persistent and higher the concentration of ozone, the more serious the event. In such applications, the goal is to find extended periods of high μ_t where μ_t is the parameter of interest, which may be observed with measurement error and sampling bias. The background may not be zero in such applications, but variants of our approach may be applied by defining a binary variable Y_t that is zero if a measurement X_t at time t is below a minimal threshold π_0 and one otherwise. Smoothing of the X_t's to estimate the mean μ_t during an event begins when $Y_t = 1$ and continues until a string of $Y_t = 0$ is observed. The smooths can be sliced, as in our application.

Covariates other than location and network name are lacking in our application but may be important in others. Covariates can be included linearly, additively or more generally; for example, the functions *gam* and *loess* in S and Splus allow smoothing with covariates. In principle, therefore, covariates can be included in an estimate of the mean during events. More work is needed to understand how the smooth surface as a function of time and other covariates should be sliced and how the dependence of slices on covariates should be recorded and modeled, but we are hopeful that some extension in those directions is possible.

References

[1] Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988), *The New S Language: A Programming Environment for Data Analysis and Graphics*, Pacific Grove, CA: Wadsworth & Brooks.

[2] Bloomfield, P., Royle, A. and Yang, Q. (1993), "Accounting for Meteorological Effects in Measuring Urban Ozone Levels and Trends," National Institute of Statistical Sciences Technical Report 1.

[3] Chambers, J.M. and Hastie, T. (eds.) (1991), *Statistical Models in S*, Pacific Grove, CA: Wadsworth & Brooks/Cole.

[4] Cleveland, W. S. and Devlin, S. J. (1988), "Locally Weighted Regression: An Approach to Regression Analysis By Local Fitting," *J. Am. Statist. Assoc.*, 83, 596-610.

[5] Cleveland, W. S. and Grosse, E. (1991), "Computational Methods for Local Regression," *Statistics and Computing*, 1.

[6] Daneshmand, M. and Savolaine, C. (1993), "Measuring Outages in Communications Switched Networks," *IEEE Communications Magazine*, June 1993, 34-38.

[7] Efron, B. and Tibshirani, R. J. (1993), *An Introduction to the Bootstrap*, New York, N. Y.: Chapman & Hall.

[8] Fairley, D.. (1989), "Comment on 'Extreme Value Analysis of Environmental Time Series: An Application to Trend Detection in Ground-Level Ozone'," *Statistical Science*, 4, 381-383.

[9] Glossbrenner, K. C. (1993), "Availability and Reliability of Switched Services," *IEEE Communications Magazine*, June 1993, 28-32.

[10] Hastie, T. and Tibshirani, R. J. (1986), " Generalized Additive Models (with discussion)," *Statistical Science*, 1, 297-310.

[11] Smith, R. L. (1986), "Extreme Value Theory Based on the r-Largest Annual Events," *J. Hydrology*, 86, 27-43.

[12] Smith, R. L. (1989), "Extreme Value Analysis of Environmental Time Series: An Application to Trend Detection in Ground-Level Ozone (With Discussion)," *Statistical Science*, 4, 367-393.

[13] Smith, R. L. and Huang, S. L., (1994), "Modeling High Threshold Exceedances of Urban Ozone," National Institute of Statistical Sciences Technical Report 6.