Estimating Rates of Extreme Events: Beyond the Threshold

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

Abstract

Some events are characterized by their severity and duration. For example, the seriousness of an ozone alert may depend on its duration and the level of ozone during the alert. In communications, the seriousness of network problems depends on the rate at which messages are blocked (fail to reach their destination) and the persistence of a high blocking rate.

In statistical terms, a random mean μ_t changes with time t but is usually near a background level, such as zero. Occasionally, however, μ_t jumps to a higher level and stays high, but not necessarily constant, for a random length of time before it returns to its background level. When μ_t remains above a severity threshold τ_S for a duration τ_D, then an *event* at thresholds (τ_S, τ_D) is said to occur. The higher the thresholds, the more serious the consequences of the event. To detect events and estimate event rates for a range of severity and persistence thresholds, observations Y_t are taken at times $t_1, t_2, \ldots$. For example, μ_t might be the probability a message sent at time t is blocked and Y_t the completion status of a message sent at time t.

This paper shows how a sequence of Bernoulli data collected throughout time can be used to detect events and to estimate event rates for wide ranges of thresholds on duration and severity. The algorithm for detecting events is simple, but, unlike past detection schemes, does not rely on aggregating the data. Our estimator for rate as a function of severity and persistence is model–free, except for one step in a sampling bias correction, and practical even if there are hundreds of millions of observations. Confidence intervals on rates are built using a kind of partial bootstrapping that is suitable for very large sets of data. An application to reliability of communications networks that motivated our work is discussed throughout the paper. Extensions to more general monitoring problems are also discussed.

Keywords Bridging, Massive Data, Monitoring, Network Reliability, Partial Bootstrap, Two–Stage Sampling.

*Richard A. Becker is Member of Technical Staff with AT&T Research, Linda A. Clark and Diane Lambert are Members of Technical Staff with Bell Laboratories.

1 Background

Some highly reliable systems almost never fail completely. Instead, from time–to–time performance degrades abruptly. The severity and persistence of the degradation vary from one incident to another, and some users are able to tolerate more severe degradation for longer periods than others can.

For example, a communications network routinely passes all messages without error. But, from time to time, some messages are *blocked* and fail to complete. Casual users may be unaware of a 10% blocking rate between two locations, but a heavy user might notice a 10% blocking rate, especially if it persisted. Similarly, casual users may not notice a 90% blocking rate that lasts several minutes, but a heavy user might notice and suffer. To meet the needs of a range of customers, reliability is managed at more than one set of blocking and duration thresholds. More background on reliability of communication networks can be found in the papers by Glossbrenner (1993) and Daneshmand and Savolaine (1993).

In statistical terms, a mean μ_t (the blocking probability between two nodes in a network in our application) evolves randomly with time t. A *failure* at thresholds (τ_S, τ_D) occurs when $\{\mu_t\}$ crosses the severity threshold τ_S and remains above τ_S for at least τ_D hours. Note that the persistence of high μ_t, rather than just a level crossing, defines a failure. If there are multiple sites, as happens in environmental monitoring and communications reliability, a *network event* is declared whenever one or more sites experience a failure. Details are given in Section 2.

Our goals are to make lists of site failures and network events for several pairs of thresholds (τ_S, τ_D) and to provide confidence intervals for the event rates at these thresholds. These are challenging goals. First, the information on events is indirect, because only observations $\{Y_t\}$, which in our application are binary and obtained by two stage sampling (Section 3), rather than the means $\{\mu_t\}$ are observed. Second, there can be hundreds of millions of observations, so relying on complicated statistical models to estimate rates is impractical. Third, even though there is a lot of data, 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.

Even if there were less data and just one site, it would be difficult to reach our goals with parametric statistical models. We would need a stochastic process that has jumps at random start times to describe the onset of failures and random functions to describe the evolution of $\{\mu_t\}$ during and outside of failures. We would also need to derive a formula for the probability that a realization of $\{\mu_t\}$ stays above τ_S for at least τ_D hours and to specify a probability model that relates the observations $\{Y_t\}$ to $\{\mu_t\}$. Many possible parametric models can be imagined, but finding the rate of persistent excursions (failures) under the chosen model, validating the model with data, and estimating the rate of failures parametrically would be difficult at best. Moreover, correlations among sites would require a multivariate rather than univariate $\{\mu_t\}$ to be built. Fortunately, detailed probability modeling can be avoided without giving up on statistical methods entirely.

Our approach has three basic steps to detecting site failures: (i) find a site and period with a possible failure, (ii) smooth the data $\{Y_t\}$ in the period to estimate $\{\mu_t\}$ and (iii) *slice* through $\{\mu_t\}$

at height τ_S for several values of τ_S. The start and stop of each part of the smooth that lies above τ_S define a *slice* at τ_S. Instead of storing the smooth itself, which might be costly when there are hundreds of millions of observations, we keep just a few statistics on the slices that exceed τ_S for at least time τ_D for several values of (τ_S, τ_D). Ignoring a few important details for the moment, the slices at thresholds (τ_S, τ_D) comprise an estimated failure list. The union over all sites of the intervals defined by the slices gives the list of estimated network events. Section 4 describes the algorithm for detecting events in more detail.

The reported events at thresholds (τ_S, τ_D) can be used to estimate the rate of events $\lambda(\tau_S, \tau_D)$ for a given threshold pair (τ_S, τ_D). The simplest rate estimator is the reported number of events divided by the time spent on test. This estimator is attractive, because it is easy to explain to nonstatisticians, but it has three disadvantages. First, events are more likely to be missed than wrongly counted in our application, which implies that the simple estimator is biased. Second, only the events at thresholds (τ_S, τ_D) are used to estimate $\lambda(\tau_S, \tau_D)$. If $\lambda(\tau_S, \tau_D)$ is continuous, $S \approx S'$ and $D \approx D'$, then the reported event list at $(\tau_{S'}, \tau_{D'})$ should help improve the estimate of $\lambda(\tau_S, \tau_D)$. Third, the simple rate estimator can be computed only at the thresholds used to report events. A better rate estimator that is less biased, smoother, and can be computed for any thresholds within the range of the largest and smallest thresholds monitored is introduced in Section 5. Model–free confidence intervals for events rates are found by a kind of partial bootstrapping in Section 6. Some extensions and further implications of our work are discussed in Section 7; a summary is given in Section 8.

We do not claim that our approach — slicing, removing sampling bias, smoothing estimated rates, and "partial bootstrapping" — is optimal in a mathematical sense. But it is practical, it is based on sound statistical principles, and it informally balances statistical efficiency, computational efficiency and ease of interpretation.

2 Site Failures and Network Events

A *site failure* at thresholds (τ_S, τ_D) occurs whenever the mean μ_{it} at site i exceeds τ_S for at least τ_D consecutive hours. More precisely, a failure at thresholds (τ_S, τ_D) starts at time S_i and has a duration $D_i \geq \tau_D$ if μ_{it} satisfies

$$\mu_{it} \begin{cases} \geq \tau_S & \text{for } t \in [S_i, S_i + D_i] \\ < \tau_S & \text{for } t \in [S_i - \epsilon, S_i) \text{ and } t \in (S_i + D_i, S_i + D_i + \epsilon] \end{cases}$$

for some $\epsilon > 0$. That is, the mean is at or above the threshold during the event but not just before or after it. Note that if the data are binary, failure does not imply that all $Y_t = 1$ but only that the probability that $Y_t = 1$ is unacceptably large.

Network events are found by taking unions of site failures, so that site failures at the same thresholds (τ_S, τ_D) that overlap in time are counted as one *network event*. The resulting list of true network events is $\mathcal{E}(\tau_S, \tau_D) = \cup_{ik}[S_{ik}, S_{ik} + D_{ik}]$, where S_{ik} and D_{ik} are the true start and duration of site failure k at site 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 site failure in an event must meet the threshold requirements, so an event at an hour long threshold cannot be built out of six consecutive ten minute failures at different sites, for example. Also note that an event may appear on several lists with different (τ_S, τ_D) values, because an event on a list $\mathcal{E}(\tau_S, \tau_D)$ also appears on any list with a shorter τ_D and a lower τ_S.

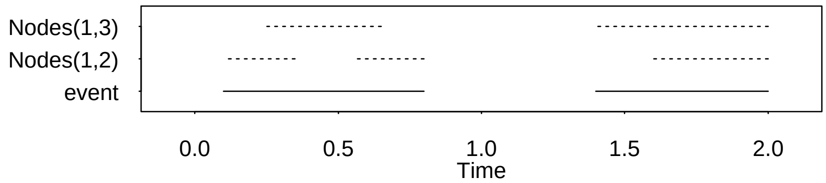

Figure 1: The dotted lines show site failures for two sites, and the solid lines show the corresponding network events.

Site failures that overlap in time are counted as only one network event, regardless of how many sites are involved. This is reasonable, if users are concerned with the network as a whole. It is also a way to ignore the correlations among concurrent site failures, which can be complex. In a communications network, for example, a disruption at node pair (site) (A, B) may cause traffic to be re-routed over a path that includes (C, D). The re-routing is chosen dynamically, so that the alternate route is not known in advance. Nor can the re-routing be deduced from the data. Even if the re-routing disrupts traffic at (C, D), the pattern of blocking at (C, D) may not resemble the pattern of blocking at (A, B), because different routes have different reserve capacities and different baseline traffic patterns. Counting site failures that overlap in time as one network event avoids the need to model such complex interactions.

The idea of counting site failures that overlap in time as only one network event can also be found in the papers by Bloomfield, Royle and Yang (1993) and Smith and Huang (1994) on environmental monitoring. These papers define network events in terms of the maximum measurement observed over a fixed set of network sites in a given unit of time, such as a day.

We define failures and events in terms of the true means $\{\mu_{it}\}$. Others, however, define failures and events in terms of the observations $\{Y_{it}\}$. For example, the papers of Smith (1986, 1989) equate failures with the maximum measurement in a fixed unit of time, such as a day, crossing a severity threshold. Using the true mean instead of the data to define events is preferable, however, because it allows an event to be defined independently of the measurement process. That is, the definition of an event does not change if technology improves and measurements become less biased or less noisy. Using the true mean rather than aggregated measurements to define events also seems more in line with common understanding of environmental regulations (see Fairly, 1989). Finally, the distinction between events defined from the true mean and events defined from the data is unimportant only if measurement error is ignorable, which it is not for binary data. We distinguish *true events*, which are based on the true mean, from *reported events*, which are estimated from the data. It is the list of true events and the rate of true events that are of concern.

As for reported events, we require an estimate $\hat{\mu}_t$ of the mean to exceed τ_S throughout the reported event, while others require an aggregated statistic, such as the mean or maximum over a fixed unit of time, to exceed a threshold. For example, others might report a one day failure when the mean or maximum measurement for an entire day exceeds a severity threshold, and a two day failure if the mean or maximum measurement for each of two consecutive days exceeds the threshold. Of course, a failure for a day based on an aggregated statistic such as the daily mean or daily maximum might be caused by an extremely high level in only one particular hour. If so, the event would have been demoted to an hour long event from a day long event if the unit of aggregation had been an hour instead of a day. Tracking an estimate of the instantaneous mean through time avoids this inconsistency.

Basing failures and events on the mean rather than extreme observations, as is common in environmental applications, may be considered a disadvantage when atypical high levels are more important than typical levels. If total exposure is a concern, however, then an integral of the mean over time, rather than a maximum in a time period, is required. Applications in which damage is inflicted instantaneously as a threshold is crossed rather than accumulated throughout an event are not the topic of this paper, because the notion of joint duration and severity thresholds is central to our work. Nevertheless, care has to be taken to avoid missing events by averaging measurements in periods of low means with measurements in periods of high means.

3 Motivating Application: Communications Networks

In our application, the objective is to compare the rates of events for two different communications networks at arbitrary blocking and duration thresholds. To compare the networks, messages are sent between pairs of nodes on each network throughout time. Considering each network separately for now, the outcome Y_{it} of a message sent at time t for node pair (A_i, B_i) of the network is 0 if the message *completes* and 1 if the message is *blocked*. The blocking rate μ_{it} for the node pair at time t is the probability that a message sent from A_i to B_i at time t is not received at B_i.

Because communications networks are usually highly reliable, most messages complete. To avoid collecting long strings of completed messages, data are obtained by two-stage sampling. *Screening* messages are routinely sent from A_i to B_i about once every 15 minutes. If the screening message from A_i to B_i completes, then no further message is sent on (A_i, B_i) until the next scheduled screening message is due. If the screening message is blocked, then *rapid sampling* begins and messages are sent rapidly over (A_i, B_i) until the next screening message is due. The number N of messages sent in a sampling interval is not fixed. Before the next screening message is sent, the blocking fraction $f = N^{-1} \sum Y_j$ for the sampling interval that just completed is computed for the node pair. If f exceeds a pre-specified threshold, then rapid sampling continues until the next screening message; otherwise, rapid sampling stops. The decision to continue or stop rapid sampling is made separately for each node pair every time a screening message is due at the node pair. To avoid cutting off data collection prematurely during a node pair failure, the threshold to cut off rapid sampling is much lower than the lowest severity threshold τ_S of interest.

Figure 2 shows the data collected during a period of unusually persistent high blocking at one node pair on one network. The dots at a blocking rate of 0 before hour 18 represent completed screening messages. The short horizontal dashes after hour 18 show the fraction of blocked messages in the sampling intervals.

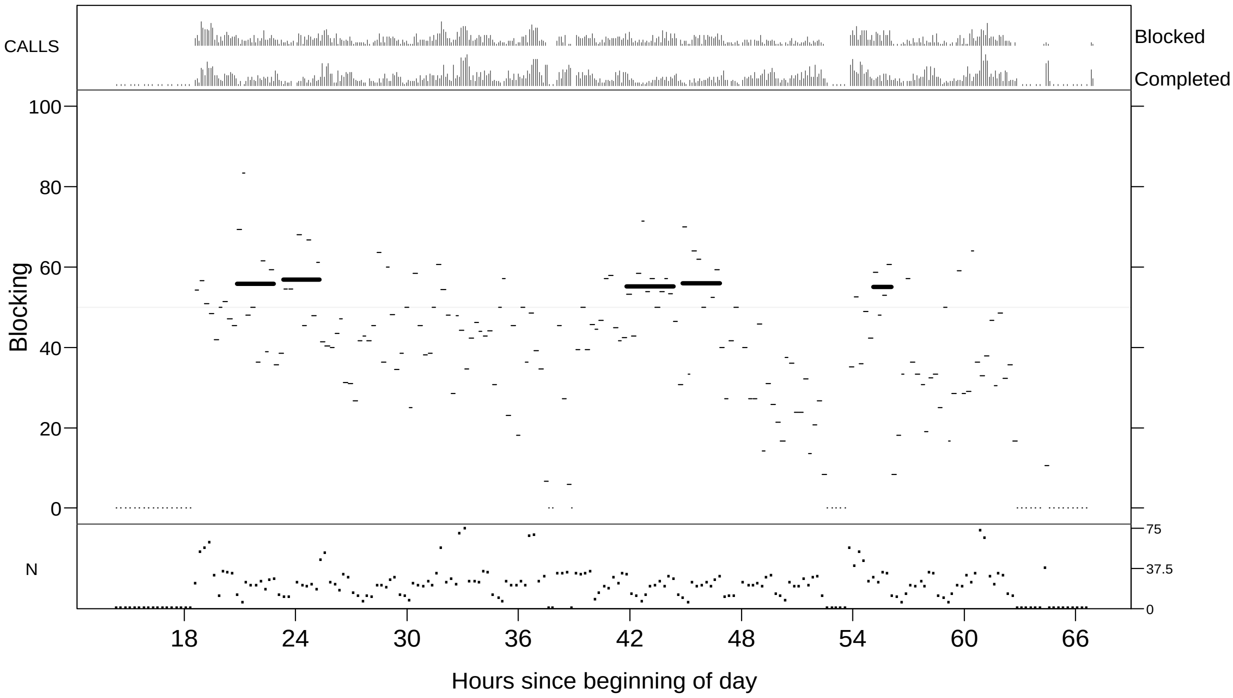

Figure 2: Data collected at a node pair during a period of persistent blocking. The plot is divided into three horizontal strips. The large middle strip uses short, light dashes to show the observed blocking fractions in the sampling intervals; dashes extend from the first to the last message in an interval. Longer, bold dashes show periods defined as a reported failure by an *ad hoc* rule based on the fifteen minute blocking averages. The faint horizontal line at 50% makes comparison to the blocking threshold easier. The numbers of blocked and completed messages throughout time are shown in the top strip; each bin is six minutes wide in this example. The numbers of test messages in the sampling intervals are shown in the lower strip. In all, nearly 4400 messages are represented in this plot.

4 Slicing, Site Failures, and Events

4.1 Finding Failures from Data

Site failures could be reported using the observed means for the sampling intervals, especially if only failures at least as long as the length of a sampling interval are of concern. For example, with a fifteen minute sampling interval, a one hour failure at 50% blocking could be reported whenever each of four consecutive fifteen minute blocking averages exceeds 50%. Five failures defined by a slightly different *ad hoc* rule based on 15 minute averages are shown in Figure 2. Such rules have several undesirable features, though. First, fifteen minutes is an arbitrary, coarse time scale that matches the rate of sampling rather than the rate of change in the true blocking rate. Second, rules based on fifteen minute averages may not have good false positive and false negative rates. Suppose, for example, that there are about 25 observations per sampling interval, a true blocking rate of .6 and thresholds of .5 and one hour. Then about 16% of the fifteen minute blocking fractions will be below .5 and the probability that four consecutive fifteen minute blocking fractions will exceed .5 is only about 50%. Thus, a failure that is one hour long is detected with probability .5, and a failure that lasts several hours is likely to be split into several shorter failures at thresholds (.5, 1 hour). In Figure 2, for example, it is not unreasonable to argue that the first four events found from the fifteen minute averages should be counted as only two events.

Therefore, the next step is to use the raw data instead of the aggregated data to detect site failures. To decide if a site mean has crossed a threshold, first estimate μ_t by smoothing the $\{Y_t\}$. With binary data, smoothing over abrupt starts and stops is avoided by delaying smoothing until an observation of $Y_t = 1$ is obtained, say at time t_0, and then stopping smoothing just before the first $Y_t = 1$ that is followed by a "long enough" string of $Y_t = 0$. 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 a message that influences $\hat{\mu}_t$ can be from t. These periods of smoothing are the only candidates for site failures, because they are the only periods with non-zero estimates of μ_t. (Data that are not binary are discussed briefly in Section 7.)

The smooths are sliced at height τ_S and the starts and stops of the pieces that last at least τ_D hours are recorded as site failures at thresholds (τ_S, τ_D) (see Figure 3). Slicing at every (τ_S, τ_D) 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 t is often adequate for estimating a mean, storing slices for only some thresholds is adequate for estimating events and event rates for a wide range of thresholds. 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, minimum estimated mean, and average estimated mean are kept for each slice. In our application, 23 values of (τ_S, τ_D) were monitored (see Figure 5).

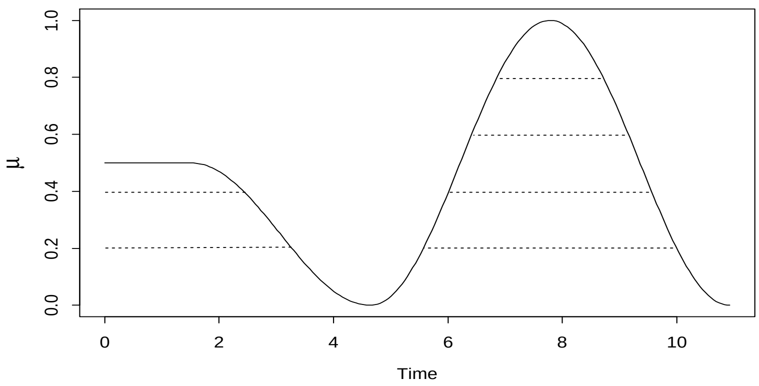

Figure 3: An artificial μ_t and its slices at $\tau_S = (.2, .4, .6, .8)$ and $\tau_D = 1$. At thresholds $(\tau_S, \tau_D) = (.2, 2)$ there would be two failures starting at $t = 0$ and $t = 5.5$, and at thresholds $(\tau_S, \tau_D) = (.4, 4)$ there would be one failure starting at $t = 5.8$.

4.2 Bridging: A Statistical Criterion for Joining Consecutive Failures at a Site

In simple cases, a slice at thresholds (τ_S, τ_D) defines a *reported site failure* at thresholds (τ_S, τ_D). There are difficult cases, however. If the true μ_t is just a little above τ_S for a long time, then measurement error is likely to cause $\hat{\mu}_t$ to wander above and below τ_S producing several slices instead of one long slice. In some applications, such overcounting is avoided by ignoring gaps that are shorter than a fixed, pre-specified duration; *e.g.*, Smith (1986, 1989). In our application, however, event counts can be sensitive to the size of the allowable gap. Instead of trying to choose the size of the allowable gap arbitrarily, we use the following statistical criterion to decide whether a gap is caused by only sampling variability and, therefore, should be ignored. Namely, k consecutive slices at the same site and same severity threshold τ_S with reported starts $\hat{S}_1, \ldots, \hat{S}_k$, durations $\hat{D}_1, \ldots, \hat{D}_k$, and gaps $\hat{G}_1, \ldots, \hat{G}_{k-1}$ between them are *bridged* and counted as one failure with start $\hat{S}_1$ and duration $\sum_{i=1}^{k} \hat{D}_i + \sum_{i=1}^{k-1} \hat{G}_i$ if

1. $\hat{\mu}_{it} \geq \tau_S$ for $t \in [\hat{S}_i, \hat{S}_i + \hat{D}_i]$, and $\hat{\mu}_{it} < \tau_S$ for $t \in (\hat{S}_i + \hat{D}_i, \hat{S}_{i+1}), i = 1, \ldots, k$;
2. $\hat{D}_i \geq \tau_D$ for at least one $1 \leq i \leq k$;
3. a one-sided upper 95% pointwise confidence limit for μ_t exceeds τ_S for $t \in (\hat{S}_i + \hat{D}_i, \hat{S}_{i+1}), i = 1, \ldots, k$

Informally, failures at a site are bridged if we are uncertain that $\mu_t < \tau_S$ during the gap between them. Periods above the threshold that are very far apart in time are not bridged, because the upper confidence limit typically drops below τ_S during long gaps. Conditions (1) and (2) ensure that at least one of the slices qualifies as a site failure on its own. Thus, bridging can add periods of sub-threshold durations onto a qualified site failure, but it cannot introduce a failure where none would be reported otherwise. In theory, $\{Y_t\}$ could be autocorrelated, so that the confidence intervals would be too narrow, but in the examples we have seen there is no noticeable autocorrelation.

Figure 4 shows the node pair failures at 50% blocking for an hour for the data shown in Figure 2. Five events were found using fifteen minute averages, but smoothing and slicing found only two.

The first failure from smoothing starts earlier than the first failure based on 15 minute averages, and it involves some bridging. The fifth failure based on 15 minute averages is not included among the failures found by smoothing. The average blocking rate during a failure is higher in Figure 2 than in Figure 4, because only fifteen minute blocking rates above 50% are included when events are called from 15 minute averages. Figure 4 also suggests that counting events at just one (τ_S, τ_D) may be unwise, because subthreshold problems of nearly the same severity continued after the first failure ended.

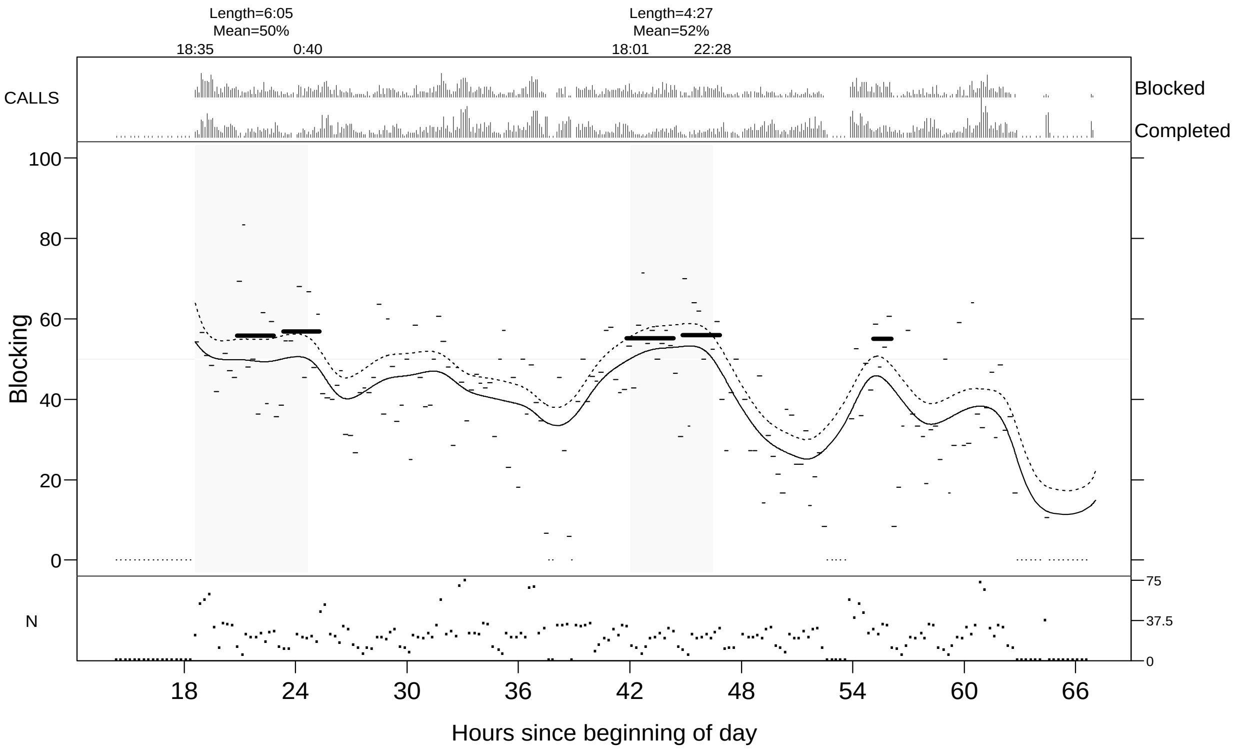

Figure 4: One hour, 50% node pair failures for the period shown in Figure 2. The solid curve shows $\hat{\mu}_t$, and the dotted gives a pointwise one-sided 95% upper confidence limit for μ_t. Shaded areas highlight failures. Note the bridging in the first reported failure when $\hat{\mu}_t$ dips slightly below 50%. The start, stop, duration and average blocking rate $\bar{\mu}$ during the failure are printed above the plot, where $\bar{\mu} = \sum_{i=1}^{M}(\hat{\mu}_{t_i} + \hat{\mu}_{t_{i-1}})(t_i - t_{i-1})/(t_M - t_1)$, and M is the number of observations from the start to the stop of the failure and t_i is the time of the i^{th} observation.

Finally, the list of reported network events $\hat{\mathcal{E}}(\tau_S, \tau_D)$ is found by taking the union of all the site failure intervals, as described in Figure 1.

4.3 Computations

Clearly, monitoring generates a lot of data. For each message in our application, the type of sampling (screening or rapid), origination and destination nodes, date and time, type of network used, and whether the message was complete or blocked is recorded. In twelve months, the two networks considered in this paper generated hundreds of millions of calls, resulting in about .5 gigabytes of raw data.

All computing is done in the S language (Becker, Chambers and Wilks, 1988). Data for different node pairs are saved in compressed files outside of S, and S functions process the data in the following steps.

1. Call Unix tools to uncompress slightly more than a month of data for a set of node pairs (a little of the previous month's and following month's data is added to the current month to avoid prematurely cutting off node pair failures that span the end of a month).
2. Call Unix tools such as grep, awk, and sort to extract and edit data from files outside of S in a pipeline.
3. Scan the edited data into S, smooth, find node pair failures at several threshold pairs (τ_S, τ_D), produce postscript files of plots that show all the information about the node pair failures, and save information about the node pair failures in S objects.

Other S functions process the information about node pair failures to find network events, estimate event rates, and compute confidence intervals.

Note that we do not keep all the data in S at once. Instead, we compute for a manageable set of data and then keep only the summaries of the data that are important for reporting events and estimating event rates. This is akin to the use of sufficient statistics for summarizing the information in a sample, but the decision about what to save is based on data analysis rather than on statistical modeling. More precisely, our analysis of the data leads us to believe that the sampling variation in $\{\hat{\mu}_t\}$ is negligible relative to the variation in the occurrence of events and to the variation among patterns of blocking $\{\mu_t\}$ from one event to another. For that reason, it is enough to keep the pointwise smooths for each node pair failure rather than the raw data. Storage in S can be reduced further, because our reported event lists, estimated event rates, and confidence intervals depend only on summary information about the slices rather than on the pointwise estimates $\{\hat{\mu}_t\}$. In our application, the slices for 23 threshold pairs (τ_S, τ_D) for each of two networks can be stored in less than 0.1% of the space required to store the raw data.

5 Estimated Event Rates

5.1 A Simple Estimator

We define the *event rate* at thresholds (τ_S, τ_D) by

$$\lambda(\tau_S, \tau_D) = E(N)/[T - E(N)E(D|N)],$$

where T is the time on test, $E(N)$ is the expected number of events at thresholds (τ_S, τ_D) in the period and $E(D|N)$ is the expected duration of an event given that there are N events. In words, λ is the mean number of events during the mean uptime at thresholds (τ_S, τ_D). In our application, we take $E(D|N) = E(D)$, because there is no evidence that the duration of events at (τ_S, τ_D) is related to the number of events at (τ_S, τ_D). For example, a plot (not shown) of duration against time-from-the-previous-event shows no pattern. A *simple estimator* of the rate of occurrence at thresholds (τ_S, τ_D) for time-on-test T is then

$$\hat{\lambda}(\tau_S, \tau_D) = N/(T - N\bar{D}), \tag{1}$$

where N is the reported number of events at thresholds (τ_S, τ_D) during the period and $\bar{D}$ is the mean of the N durations $\hat{D}_1, \ldots, \hat{D}_N$.

The simple rate estimator is biased, because N and $\bar{D}$ are biased estimators of the true number of events and average duration. For example, consider the first reported failure in Figure 4. It starts with the onset of rapid sampling at 18:35. The actual failure might have started earlier, however. It might have started after the previous screening message at 18:20 but before 18:35, because the node pair was not monitored for those 15 minutes. Or it might have started even earlier than 18:20. If the true blocking rate is $\mu_t = .6$, then with probability .4 the screening message at 18:20 completes and no further message is sent until 18:35, even though the node pair failure has already begun. Thus, the reported duration may underestimate the true duration. A failure can even be left off a list entirely, if the portion of the event that occurs after the onset of rapid sampling is shorter than the required τ_D. Missing the beginning of an event is the only important source of bias in the simple rate estimator, however. Rapid sampling is in effect at the end of the failure, so there is no reason to expect that the end time is either over or underestimated. The estimated mean during an event is also not obviously biased. (Simulations not reported here support these claims.)

5.2 The Bias in Reported Duration

An event has three important times: its start t_0, the onset of rapid sampling t_R, and its end t_1. Its *true duration* $\delta = t_1 - t_0$ is composed of two parts: the *monitored duration* $D = t_1 - max(t_R, t_0)$ and the *unmonitored duration* $\delta - D$. The *reported duration* $\hat{D} = \hat{t}_1 - \hat{t}_0$ estimates the monitored duration, which may be shorter than the true duration. Our goal is to add the missing portion $\delta - D$ to $\hat{D}$ when $t_0 < t_R$.

Bias correction is based on the following assumptions about events for which $\delta > D$.

A1. Each site i involved in a network event from t_0 to t_R has a constant mean μ_i;

A2. If multiple sites are involved in an event for which $t_0 < t_R$, then each site failure starts at t_0.

A3. A site failure is equally likely to start any time during the unmonitored portion of a sampling interval;

A4. Screening observations are taken simultaneously every $1/r$ hours at all sites involved in an event;

A5. $\delta \sim lognormal(\theta_1, \theta_2)$, where (θ_1, θ_2) is independent of the μ_is for the site failures that make up the event and δ is independent of the μ_is given θ.

A6. The difference $D - \hat{D}$ between the monitored duration and the reported duration is negligible.

All these assumptions are reasonable in our application. Assumption (A1) is approximately true, because events tend to be short. Assumption (A2) holds because node pair failures are rare, and overlapping events that start abruptly in several node pairs are often thought to share a common cause. Assumption (A3) is difficult to verify, but is plausible and supported by an analysis of the reported failures that started after the onset of rapid sampling (*i.e.*, the failures that may start during a sampling interval). Assumption (A4) is consistent with the sampling design for our application. As for (A5), normal scores plots (not shown) of reported log durations for various ranges of average estimated blocking rate do not contradict lognormality or independence of blocking rate and (θ_1, θ_2). (As will be seen, adjustments to duration tend to be short relative to the length of events, so uncorrected duration is a reasonable if imperfect way to judge the distribution of true duration). Independence of severity and duration is plausible because all network events are considered to be serious, and they are sufficiently rare that each is ended as soon as possible no matter what its blocking rate. The accuracy of (A6) depends on the quality of $\hat{\mu}_t$, which in turn depends on the sampling rate during rapid sampling.

For now, suppose that only one site is involved in an event and the data are binary. Then two conditions must be satisfied to obtain a monitored duration of $D < \delta$ with a screening sampling rate of r calls per hour. (i) The first $M = \lfloor r(\delta - D) \rfloor$ screening observations during the event must be zero, so that the event is not detected until the $(M+1)^{th}$ screening observation, where $\lfloor x \rfloor$ is the largest integer in x. (ii) $E = \delta - r^{-1}M - D$ hours must pass before the first screening observation during the event. Because E is uniformly distributed on $[0, r^{-1}]$ and M has a geometric$(1-\mu)$ distribution, it follows that

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

When only durations longer than δ_0 are kept after slicing, the density of monitored duration is

$$f(D|\delta, \mu, D \geq \delta_0) = f(D|\delta, \mu) / \int_{\delta_0} f(x|\delta, \mu) dx \qquad if \quad \delta \geq D \geq \delta_0.$$

Bayes Theorem then gives the following posterior for δ given a monitored duration of D:

$$f(\delta|\mu, D, \theta_1, \theta_2) = \frac{f(D|\delta, \mu, D \geq \tau_{D_0}) P(D \geq \tau_{D_0}|\delta, \mu) h(\delta|\theta_1, \theta_2)}{\int_{\tau_{D_0}} f(D|x, \mu, D \geq \tau_{D_0}) P(D \geq \tau_{D_0}|x, \mu) h(x|\theta_1, \theta_2) dx} \tag{3}$$

if $\delta \geq D \geq \tau_{D_0}$ and $t_0 < t_R$. Note that if the reported event starts during rapid sampling ($t_0 \geq t_R$), then

$$f(\delta|\mu, D) = \begin{cases} 1 & \text{if } \delta = D \geq \tau_{D_0} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Finally, by (A6), equations (3) and (4) hold approximately if D is replaced by $\hat{D}$.

When more than one site is involved in an event, equation (3) holds if μ is replaced with $1 - \prod(1-\mu_k)$, where μ_k is the mean for the k^{th} failed site. This is because all the failures start simultaneously (A2) and all the sites in a failure take screening observations simultaneously (A4).

The next step is to estimate the parameters (θ_1, θ_2) of h, which, by assumption, do not depend on μ. Our estimates are based on the list of reported *base events* $\hat{\mathcal{E}}_0$ corresponding to the lowest duration and severity thresholds (τ_{S_0}, τ_{D_0}), because this gives the largest set of reported events. (If not, then the assumption that the μ_is are constant and independent of (θ_1, θ_2) is wrong.) Note that we require that the *base threshold pair* (τ_{S_0}, τ_{D_0}) be monitored where τ_{S_0} is the minimum τ_S monitored and τ_{D_0} is the minimum τ_D monitored. The estimates $(\hat{\theta}_1, \hat{\theta}_2)$ are found by maximizing

$$\begin{aligned} L(\theta_1, \theta_2) &= \sum_{\hat{\mathcal{E}}_0} log \left(\int\int f(\hat{D}_i |\, \hat{\mu}_i, \delta_i, \theta_1, \theta_2) f(\hat{\mu}_i, \delta_i | \theta_1, \theta_2)\, d\delta_i d\mu_i \right) \\ &= \sum_{\hat{\mathcal{E}}_0} log \left(\int_{\tau_{D_0}} f(\hat{D}_i |\, \hat{\mu}_i, \delta_i) h(\delta_i |\, \theta_1, \theta_2)\, d\delta_i \right) + constant \end{aligned}$$

where $f(\hat{D}_i |\, \hat{\mu}_i, \delta_i)$ is given by equation (3), the constant is independent of (θ_1, θ_2) and $\hat{\mu}_i$ is the average estimated blocking rate during the i^{th} base event. It is straightforward to maximize L by replacing the integral with a sum over a fine grid on $\delta_i = exp(z_i)$. Once $(\hat{\theta}_1, \hat{\theta}_2)$ is computed, the bias-corrected duration of an event with $\hat{t}_0 = t_R$ and reported duration $\hat{D}$ then corresponds to a density $f(\delta | \hat{D}, \hat{\mu}, \hat{\theta}_1, \hat{\theta}_2)$ of true durations.

5.3 A Bias Corrected Rate Estimator

Given a threshold pair (τ_S, τ_D), there are three kinds of events of interest: (i) reported events at (τ_S, τ_D) for which the estimated start time occurs after rapid sampling begins ($\hat{t}_0 > t_R$), (ii) reported events at (τ_S, τ_D) for which $\hat{t}_0 = t_R$, and (iii) reported events at (τ_S, τ_{D_1}) for which $\hat{t}_0 = t_R$ and $\tau_{D_1} \le \hat{D} < \tau_D$. In words, (i) includes reported events whose duration does not need to be corrected, (ii) includes reported events that are longer than required but might have been even longer had rapid sampling started earlier, and (iii) includes reported events that are too short to qualify at τ_D but might qualify after correcting for missing the beginning of the event. Given a pair of thresholds (τ_S, τ_D), these three lists are mutually exclusive. Events of type (iii) are found from the list of reported events at $(\tau_S, \tau_{D'})$ where $\tau_{D'}$ is the shortest persistence threshold monitored for severity threshold τ_S. If τ_D is the shortest persistence threshold monitored at τ_S, then events of type (iii) are found from the list of reported events at (τ_{S_1}, τ_{D_1}) where τ_{S_1} is the smallest severity threshold monitored that is larger than τ_S and τ_{D_1} is the smallest duration threshold monitored at τ_{S_1}. Of course, at (τ_{S_0}, τ_{D_0}) there are no events of type (iii) because no shorter events are retained. This is not a problem in our application, because the base thresholds were chosen to allow bias correction, not because $\lambda(\tau_{S_0}, \tau_{D_0})$ is itself interesting.

Let N be the number of events of type (i) at (τ_S, τ_D) that are not subject to bias correction, and suppose these have average duration $\bar{D}$. Also suppose that there are a total of M events of type (ii) and type (iii) that are subject to length correction, and the i^{th} of these has observed duration

$\hat{D}_i$, average blocking rate $\hat{\mu}_i$, and posterior density $f(\delta|\hat{D}_i, \hat{\mu}_i, \hat{\theta}_1, \hat{\theta}_2)$ for the true duration. Then the *bias-corrected rate estimator* $\tilde{\lambda}(\tau_S, \tau_D)$ is defined as

$$\begin{aligned}
\tilde{\lambda}(\tau_S, \tau_D) &= \frac{N + \sum_{i=1}^{M} P(\delta \geq \tau_D|\hat{D}_i, \hat{\mu}_i, \hat{\theta}_1, \hat{\theta}_2)}{T - N\bar{D} - \sum_{i=1}^{M} E(\delta|\delta \geq \tau_D \hat{D}_i, \hat{\mu}_i, \hat{\theta}_1, \hat{\theta}_2) P(\delta \geq \tau_D|\hat{D}_i, \hat{\mu}_i, \hat{\theta}_1, \hat{\theta}_2)} \\
&= \frac{N + \sum_{i=1}^{M} \int_{\delta \geq \tau_D} f(\delta|\hat{D}_i, \hat{\mu}_i, \hat{\theta}_1, \hat{\theta}_2) d\delta}{T - N\bar{D} - \sum_{i=1}^{M} \int_{\delta \geq \tau_D} \delta f(\delta|\hat{D}_i, \hat{\mu}_i, \hat{\theta}_1, \hat{\theta}_2) d\delta}
\end{aligned}$$

where T is the total time on test. In the numerator, each of the N reported events that exceeds the required thresholds is given a score of 1 and each of the M reported events that could exceed the thresholds after bias correction is given a score of the estimated probability that it exceeds (τ_S, τ_D). That is, $\tilde{\lambda}(\tau_S, \tau_D)$ estimates the expected number of events at thresholds (τ_S, τ_D) during the expected uptime at thresholds (τ_S, τ_D).

In our application, $\lambda(\tau_S, \tau_D)$ was estimated at 23 threshold pairs for each of two networks over a twelve month period. Bias correction increased the estimated event rates, as expected, but sometimes only slightly. The median, third quartile, and maximum of the relative increases at the 23 threshold pairs were 1.05, 1.16 and 1.38 for network I and 1.04, 1.08 and 1.14 for network II. The corrections are smaller for network II because its events tend to have higher blocking, and, other things being equal, the larger the blocking rate during an event, the smaller the chance that a screening observation is zero, and the smaller the correction for sampling bias.

Recall that the objective in our application is to compare the reliability of networks I and II. Figure 5 shows the ratio of uncorrected estimates and the ratio of bias-corrected estimates for the two networks. The bias-corrected ratios are smoother, because some of the events reported at a lower duration are counted fractionally in the bias-corrected estimate, lessening the distinction between thresholds.

We have not corrected for missing events completely, although it would be possible to include the expected number of events with $\hat{D} < \tau_{D_0}$ and $\delta > \tau_D$. In our application, the probability of such events is so low that correcting for them is unimportant. This was fortunate, because there was a reluctance to include an inflation factor for events for which there was no reported evidence.

The only remaining step is to estimate the event rates for arbitrary thresholds; *i.e.*, to estimate the surface $\lambda(\tau_S, \tau_D)$. To do this, we assume that $log(\lambda(\tau_S, \tau_D))$ is locally linear in (τ_S, τ_D) and $variance(\hat{\lambda}_0(\tau_S, \tau_D)) \propto \lambda(\tau_S, \tau_D)$. To be more specific, a degree 1 loess model is fit to $log(\hat{\lambda}(\tau_S, \tau_D))$ using τ_S and τ_D as predictors, and then the loess model is re-fit using the fitted values from the first loess as weights in the second loess. (See Cleveland and Devlin (1988) for a discussion of loess smoothing 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.) Only a few minutes are required to compute $(\hat{\theta}_1, \hat{\theta}_2)$, $f(\delta|\hat{D}_i, \hat{\mu}_i, \hat{\theta}_1, \hat{\theta}_2)$ and the *smooth bias-corrected estimate* $\tilde{\lambda}_S(\tau_S, \tau_D)$ after the lists of reported site failures are found.

Figure 5 shows the ratio of smooth, bias-corrected estimates for the data considered in this paper. Network II's event rates are at least twice as large as Network I's at all thresholds considered. Roughly speaking, the ratio becomes more unfavorable for II as duration increases for a fixed

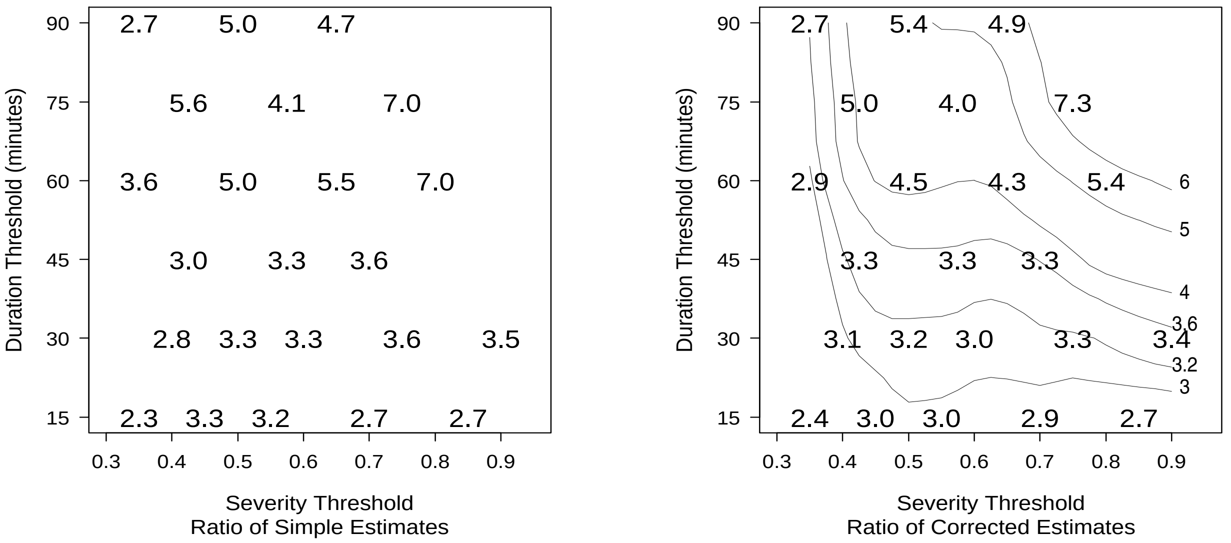

Figure 5: The left panel shows the ratio of simple estimates $\hat{\lambda}_0(\tau_S, \tau_D)$ for network II relative to network I at the threshold pairs monitored in our application. The right panel shows the ratio of bias-corrected estimates $\tilde{\lambda}(\tau_S, \tau_D)$ as points and the ratio of smooth bias-corrected estimates $\tilde{\lambda}_S(\tau_S, \tau_D)$ as contours.

blocking rate or as blocking increases for a fixed duration threshold, especially as blocking surpasses .6.

6 Standard Errors and Confidence Intervals for Event Rates

Nonparametric confidence intervals for $\lambda(\tau_S, \tau_D)$ 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 sampling variability are negligible with hundreds of millions of observations, so re-sampling the raw data is not necessary. For example, in our application, rapid sampling is fast enough that the blocking rate during rapid sampling and the end of a reported event are estimated with little error. Therefore, the important sources of variability in estimating $\tilde{\lambda}(\tau_S, \tau_D)$ and $\tilde{\lambda}_S(\tau_S, \tau_D)$ are the randomness in the occurrence of events and the randomness in the start of rapid sampling, not the error in estimating event properties once the event is monitored. Therefore, 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.

Here, we assume that the waiting times from the end of one base event (event at the lowest thresholds (τ_{S_0}, τ_{D_0})) to the start of the next base event are independent and identically distributed, and we re-sample these to simulate the time periods when no events occur. Not having stored the complete blocking function $\hat{B} = \{\hat{\mu}_t\}$ during an event, we cannot simulate $\hat{B}$ by re-sampling from the set of observed $\hat{B}_1, \hat{B}_2, \ldots$. But any base event has an associated list of nested events at all higher, monitored thresholds, as in Figure 3, and that "nest" approximates the true blocking rate. Therefore, re-sampling from the set of base events provides a natural way to re-sample events at all thresholds simultaneously. Also, re-sampling base events gives a sample of the binary variables that determine whether the event starts before or after the onset of rapid sampling.

If event rates are estimated from a year of data, then bootstrapping proceeds by alternately sampling first the time between base events and then the list of base events to form new year long histories. More precisely, start the bootstrap history by randomly choosing a W_1^* from the set of observed waiting times between base events. 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 event of the year by randomly choosing a base event from the observed list of base events. This gives a base duration of D_1, an indicator for whether rapid sampling started before or after the onset of the event, and possibly a list of associated events at higher thresholds. Next, choose the time W_2 from the end of the first event until the start of the second event by randomly choosing one of the observed waiting times between base events. Choose the properties of the second event by randomly choosing one of the base events, which again has an associated nest of events at higher levels. Continue sampling times-between-events and event properties 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 waiting time (rather than an event) is in progress at the end of the year (midnight of December 31 of the bootstrap year), then the bootstrap year ends on schedule and the excess waiting time beyond midnight December 31 is used as the first waiting time for the next bootstrap year. If an event is in progress at the end of the year, then the year is not ended until the event is over. This policy is consistent with current practice, but it does mean that 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 are not affected by calendar boundaries.

Once the bootstrapped events are generated, it is straightforward to compute the smooth bias-corrected estimates for each bootstrapped year for the thresholds (τ_S, τ_D) of interest. The sample standard deviation of the bootstrapped, yearly values of $\tilde{\lambda}_S(\tau_S, \tau_D)$ estimates the standard error of $\tilde{\lambda}_S(\tau_S, \tau_D)$, for example. Simple 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 our application are shown in Figure 6. These were obtained by taking a bootstrap year for each network separately, taking the ratio of the rate 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, $\tilde{\lambda}(\tau_S, \tau_D)$ are shown in Figure 7. The contours of the ratios of $\tilde{\lambda}(\tau_S, \tau_D)$ are noisier and less interpretable than the contours based on $\tilde{\lambda}_s(\tau_S, \tau_D)$. The ratios at the 23 monitored (τ_S, τ_D)

are numerically similar, however; the smoothed and unsmoothed ratios at any of the 23 thresholds are within 25% of each other. As would be expected, the confidence intervals based on $\tilde{\lambda}_s(\tau_S, \tau_D)$ tend to be shorter, with the greatest reduction for τ_S between .40 and .50.

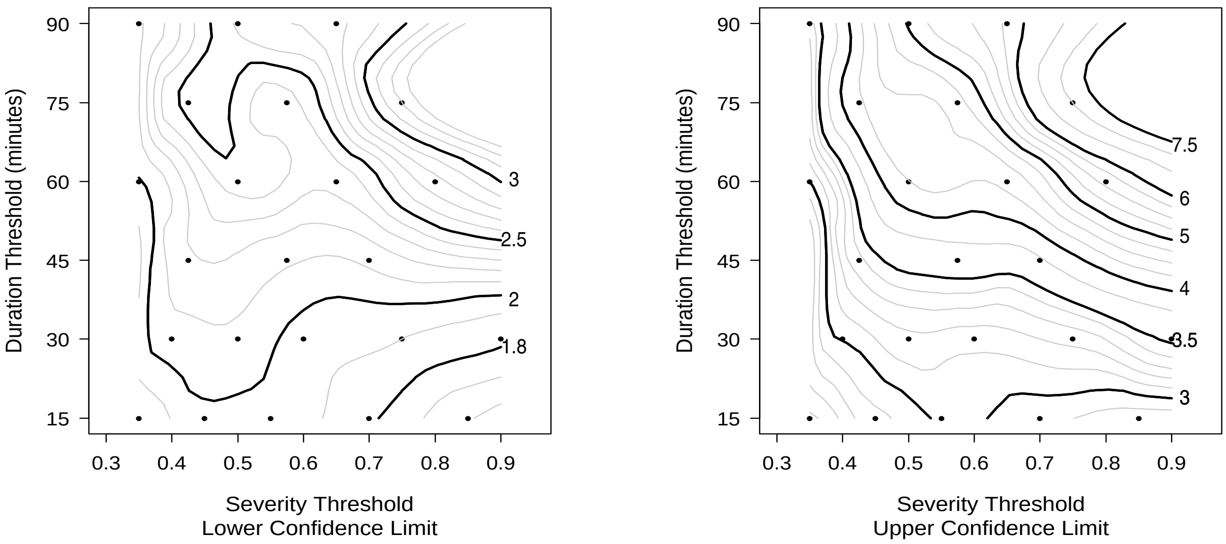

Figure 6: The left panel shows the lower 90% pointwise confidence limit for network II's event rate relative to network I's based on the bias-corrected, smooth estimates. The right panel shows the upper 90% pointwise confidence limit. Points show the thresholds (τ_S, τ_D) at which the simple estimates shown in Figure 5 are computed.

7 Possible Extensions

The methods proposed in this paper were developed for communications reliability data, but we believe that severity and duration thresholds are important 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. The damage caused by flooding may also depend on the height (severity) of the flood and its persistence.

In our application, the data are binary and the background measurement is zero, making it easier to spot the beginning and end of events. The background may not be zero in other applications, but environmental engineers and hydrologists, for example, may reduce the data to a binary variable that indicates whether the maximum measurement in a day exceeds a threshold and to a continuous variable that reports the amount (zero or positive) by which the threshold is exceeded. Variants of our approach can be applied instead by replacing each raw measurement X_t throughout time (rather than just the maximum in a day) with a binary variable Y_t that is zero if a minimal threshold τ_{S_0} is not exceeded and 1 otherwise. Smoothing of the X_t's to estimate the mean μ_t during an event should start when the corresponding Y_t becomes nonzero and stop when a string of $Y_t = 0$ is

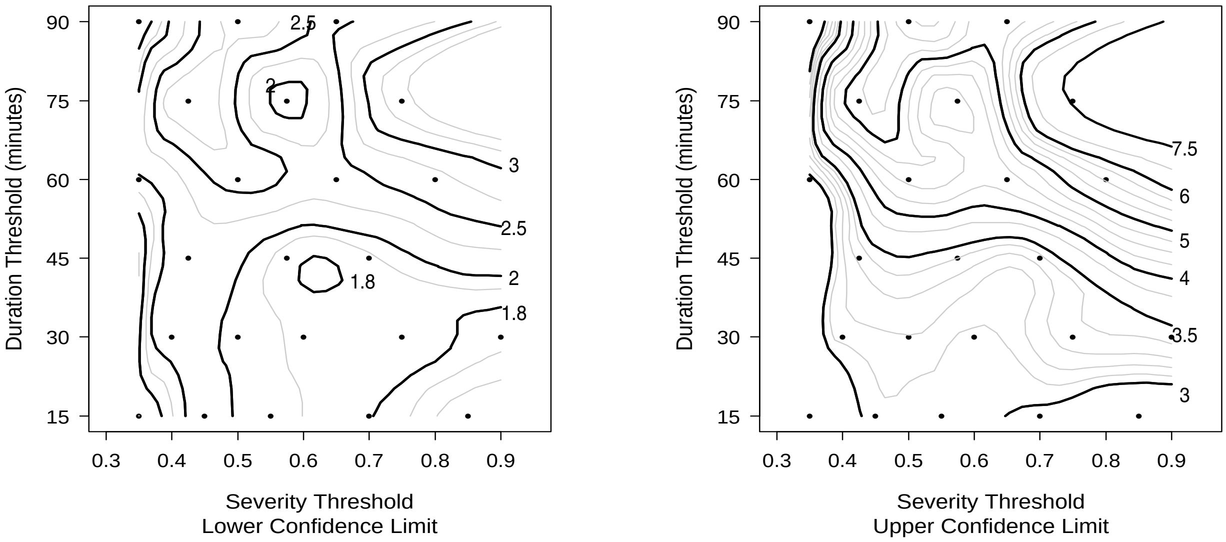

Figure 7: The left panel shows the lower 90% pointwise confidence limit for network II's event rate relative to network II's based on the bias-corrected estimates without smoothing; the right shows the upper 90% pointwise confidence limit. Points show the thresholds (τ_S, τ_D) at which the simple estimates shown in Figure 5 are computed.

observed. The periods of smoothed means can then be sliced, as in our application.

Covariates are lacking in our application but may be important in others. There are many ways to include covariates linearly, additively or more generally when smoothing against time; for example, the functions *gam* and *loess* in S and Splus allow smoothing with covariates. In principle, therefore, there is no reason that covariates cannot 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, however.

8 Summary

We have described a way to detect periods (events) in which a mean exceeds a specified level for a specified duration or longer. Events defined by joint severity and duration thresholds are implicitly of interest when total exposure rather than instantaneous peak exposure is the concern. Our approach does not require the data to be aggregated, but instead works with the raw measurements. This allows the duration of the event to be reported in the time units of the raw data rather than in the time units used to aggregate the data. We advocate computing a continuous estimate of the mean and slicing that estimate at a collection of different severity thresholds. The start and stop times that result from slicing are a compact representation of the events. Such lists of events are required in some applications.

We have shown how the lists of events for a set of thresholds can be used to provide point estimates and confidence intervals for the rate of events at any threshold within the span of thresholds monitored. Partial bootstrapping of events, rather than full bootstrapping of the raw data, is an efficient way to estimate standard errors with very large sets of data.

Slicing at different thresholds allows the analyst to explore the sensitivity of the lists of reported events to the severity and duration thresholds. For example, regulations may require a particular set of severity and duration thresholds to be monitored, but it may still be important to understand how small changes in the thresholds affect the reported list of events and the estimated event rate. Smoothing event rates as a function of severity and duration threshold gives an estimated surface that shows event rates for all combinations of severity and duration within the span of monitored thresholds. This surface allows us to explore the sensitivity of event rates to the choice of threshold.

Whenever the mean is close to a severity threshold, sampling variation may cause the estimated mean to wander above and below the severity threshold, producing several shorter events rather than one long event. We show that this undesirable behavior can be avoided, or at least reduced, by using an upper confidence limit for the mean to determine whether a gap between two reported events should be ignored. As long as the upper confidence limit is above the threshold, then consecutive periods above the threshold are joined together (bridged) into one long event.

We have also shown how to correct for bias when mean levels are estimated via two-stage sampling. Two-stage sampling is often used to detect rare events. That is, measurements are taken infrequently when the estimated mean is close to its background level and more intensively when the mean may be out of control, which happens rarely. For example, only one manufactured part in every hundred may be routinely examined for a rare defect, but when the examined part is defective then every part may be examined until the process is back in control. Bias affects event durations with two-stage sampling when the rate of routine sampling is too low to pinpoint the start of an event.

Communications monitoring, and environmental engineering all generate 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 is generally a statistically inefficient way to reduce the data. Summarizing by slices preserves more of the information that is important for understanding the character and rate 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, but by removing sources of variability that are unimportant for the statistical goals rather than by aggregating before statistical analysis begins. That is, analysis of variance in the broad sense is a paradigm that allows statisticians to understand rich, complex sets of data such as those that arise in network reliability.

Finally, we have described a way to think about extending our techniques to continuous data. Much work needs to be done to understand the limitations of our approach for continuous data with covariates, but we are hopeful that some extension in those directions is possible.

Acknowledgements We 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.

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.