[3] K. A. Doksum and B. S. Yandell, *Handbook of Statistics*, vol. 4. Elsevier, 1984.

[4] W. Feller, *Introduction to Probability Theory*, vol. II. New York: John Wiley and Sons, 1972.

[5] J. Gray, "Why do computers stop and what can be done about it?," Tech. Rep. 85.7, Tandem Computers, June 1985.

[6] J. Gray, "A census of Tandem system availability between 1985 and 1990," Tech. Rep. 90.1, Tandem Computers, Jan. 1990.

[7] K. Harrenstien, M. K. Stahl, and E. Feinler, "NICNAME/WHOIS," Tech. Rep. RFC-954, USC Information Sciences Institute, Oct. 1985.

[8] D. D. E. Long, J. L. Carroll, and K. Stewart, "Estimating the reliability of regeneration-based replica control protocols," *IEEE Transactions on Computers*, vol. 38, pp. 1691–1702, Dec. 1989.

[9] P. V. Mockapetris, "Domain names — concepts and facilities," Tech. Rep. RFC-1034, USC Information Sciences Institute, Nov. 1987.

[10] P. V. Mockapetris, "Domain names — implementation and specification," Tech. Rep. RFC-1035, USC Information Sciences Institute, Nov. 1987.

[11] S. Mourad and D. Andrews, "The reliability of the IBM/XA operating system," in *Proceedings* 15th *Annual International Symposium on Fault-tolerant Computing*, IEEE, June 1985.

[12] J.-F. Pâris, "Voting with witnesses: A consistency scheme for replicated files," in *Proceedings* 6th *International Conference on Distributed Computing Systems*, (Cambridge), pp. 606–612, IEEE, 1986.

[13] J. B. Postel, "User datagram protocol," Tech. Rep. RFC-768, USC Information Sciences Institute, Aug. 1980.

[14] J. B. Postel, "Internet control message protocol," Tech. Rep. RFC-792, USC Information Sciences Institute, Sept. 1981.

[15] J. B. Postel, "Internet protocol," Tech. Rep. RFC-791, USC Information Sciences Institute, Sept. 1981.

[16] Sun Microsystems, Incorporated, "RPC: Remote procedure call protocol specification version 2," Tech. Rep. RFC-1057, USC Information Sciences Institute, June 1988.

[17] K. S. Trivedi, *Probability & Statistics with Reliability, Queuing and Computer Science Applications*. Englewood Cliffs, New Jersey: Prentice-Hall, 1982.

In the case where the total up-time X is exponentially distributed, the distribution of the sampled up-time V matches X in both shape and mean. It is reasonable to assume that the shape of X approximates that of V when the distributions are approximately exponential. This implies that in typical cases, a sampling of up-times can be treated as a sampling of times-to-failure.

The pattern of failures of many classes of hosts were clearly not exponential. This is not surprising when the data contains points that are obviously invalid, but many classes still failed the test for exponentiality even when such noise is factored out. It is generally true that larger sample sizes provide more evidence against the exponential hypothesis. Massive amounts of data can more easily highlight minute deviations from exponential behavior. The sample comprised of all Internet responses conclusively failed the exponentiality test, pointedly underscoring this tendency. For moderately-sized samples, it was often not possible to exhibit the deviation from exponentiality, lending credence to the common practice of assuming that MTTF is exponentially distributed.

Availability was difficult to estimate accurately using the Internet. This was due to the many possible reasons for a host not responding to a request, most of which are indistinguishable to the polling process. Among these are the host being down, the host not implementing the polling protocol, and both hard and soft network failures.

The measured availabilities, MTTF and MTTR estimates reported here correspond with common experience and are well within in the range of reasonable values.

Acknowledgements

The authors are grateful to all those who contributed data directly to this study and to the hundreds of system administrators who allowed their sites to be polled. The direct contributors were: R. Appleton, J. Bloom, B. Bolosky, D. Briscoe, L. Cova, S. Cowan, L. van Doorn, F. Douglis, P. Dyson, B. Ehrmantraut, A. El Abbadi, P. Galvin, R. Golding, R. Guy, B. Hafner, J. Harris, D. Keppel, D. Kotz, K. Lam, R. LaRowe, M. Lee, S. Macrakis, S. Marks, S. Mehta, R. Michiels, C. Pu, A. Radics, P. Reiher, T. Robinson, S. Schwartz, C. Shub, C. Siebenmann, D. Smith, J. Thomas and D. Wallace. Jehan-François Pâris, Richard Golding, and Mary Long also contributed through their thoughtful comments.

Several of the closed-form solutions were found with the aid of *Maple*, a symbolic algebra program developed by the Symbolic Computation Group at the University of Waterloo.

References

[1] J. L. Carroll and D. D. E. Long, "The effect of failure and repair distributions on consistency protocols for replicated data objects," in *Proceedings of the* 22^{nd} *Annual Simulation Symposium*, (Tampa), pp. 47–60, IEEE, Mar. 1989.

[2] D. R. Cox and P. A. Lewis, *The Statistical Analysis of Series of Events*. London: Chapman and Hall, 1966.

The MTTR for the Sun 4/330 server is not included in the table since the misleading 100% availability implies an equally erroneous estimated zero MTTR. Similarly, the Sun 4/65 has an exceptionally low MTTR caused by the high estimated availability. A more accurate estimate can be derived either by using a larger sample or by repeated polling over a long period.

The Sun 3/50 has a very high MTTR. This is due to the low availability reported in table 5. The reasons for the high MTTR are the same as for the low availability. The MTTR of all Sun systems is impacted by the high MTTR of the many Sun 3/50s.

Table 7: Mean-Time-to-Repair for Sun Systems

Model	n	MTTR (days)	Model	n	MTTR (days)
4/60	1047	1.2435	3/50	1486	3.1258
4/65	38	0.4321	3/60	876	1.4113
4/110	136	1.1755	3/80	199	1.8785
4/280	144	0.4963	3/180	125	0.2155
4/390	24	0.5360	3/280	206	0.7570
All 4s	1583	1.1259	All 3s	3960	2.0209

8 Summary

The data gathered for this study were collected over a period of several months. The first phase, which lasted about seven days used Sun RPC to request statistics including the length of time the host had been up. Several months later, these hosts were again polled to determine how many were currently up.

A list of top-level domains was obtained from the network information center. The domain servers for each of these top-level domains were then queried for lists of hosts at each site. Lists of secondary domains were difficult to obtain, and so only hosts known to the primary domain servers for each site were considered. Many the domain servers for several large sites would not provide lists of hosts. Even so, the lists provided by cooperative domain servers contained over $100,000$ hosts. Responses were received from almost $13,000$ of these, providing a wealth of data for statistical analysis. In the process of gathering system statistics, an error in `rpc.statd` that causes it to occasionally return up-time estimates of more than $7,000$ days was encountered. These responses were clearly erroneous, but were rather common.

The domain servers were also used to determine the type and operating system of each host. This information was useful in analyzing the system status information returned by each operational host. Some sites did not provide any host-specific information. Since such hosts cannot be classified, the information provided by these hosts was of little value. A more challenging problem stemmed from the many ways a system administrator may describe a host, making it difficult to precisely classify the host.

solution to this system of equations is given by

$$p(t) = \frac{\mu}{\lambda + \mu} + \xi(t),\; q(t) = \frac{\lambda}{\lambda + \mu} - \xi(t) \tag{3}$$

with

$$\xi(t) = \frac{\lambda e^{-t(\lambda+\mu)}}{\lambda + \mu}.$$

When λ and μ are both greater than 0, $\lim_{t\to\infty} \xi(t) = 0$, and this limit converges at an exponential rate. For example, if $1/\lambda = 15$ days and $1/\mu = 2$ days, then after 30 days, $\xi(30) < 0.0000001$.

Since a typical host has been installed longer than 30 days, it is reasonable to assume that it has reached steady-state. In that case, the host can be modeled with the system of linear equations derived by taking the limit in equation 3 as t approaches infinity, yielding

$$\lambda p = \mu q,\; p + q = 1.$$

The solution to this system of equations is

$$p = \frac{\mu}{\lambda + \mu},\; q = \frac{\lambda}{\lambda + \mu}. \tag{4}$$

Since the MTTF of host can be estimated as in §5, the MTTR can be found given the steady-state probability p of the host being operational and by solving equation 4 for μ. The resultant equation is

$$\mu = \frac{\lambda p}{1 - p}.$$

Since MTTR $= \frac{1}{\mu}$ and MTTF $= \frac{1}{\lambda}$, the formula for MTTR is

$$\text{MTTR} = \frac{\text{MTTF}(1 - p)}{p}. \tag{5}$$

While many of the samples collected failed the rigorous test for exponentiality, their strong resemblance to exponential distributions increases the likelihood that equation 5 will provide an accurate estimate of the MTTR. The estimate provided should be a close approximation to the actual MTTR since it has been shown [1] that for Markov models virtually identical to that presented above, and for some models that are significantly more complex, the distribution has little effect on the results obtained.

The average host availability p was derived in §6. The results summarized in table 7 are obtained by combining this information with the MTTF derived in §5.

The estimates of MTTR derived for Sun systems are are summarized in table 7. The servers, such as the Sun 4/280, 4/390, 3/180 and 3/280 have a lower MTTR than the work stations. This is to be expected since servers are a critical resource and are usually maintained by support staff and are kept under a service contract. By contrast, work stations are less critical and may be serviced less frequently[7] than servers.

[7] Two common approaches are weekly service by a technician and mail-in service where faulty parts are replaced through the mail.

Table 6: Availability of Various Systems.

Model	n	Responses	% Available
Sun 386i	173	146	84.39
Sun 4	1583	1477	93.30
Sun 3	3960	3555	89.77
Suns	6520	5868	90.00
VAX (RPC)	169	163	96.45
VAX	5278	3482	65.97
Sequent	33	31	93.94
Pyramid	71	60	84.51

Sun RPC. Of these, 163 answered the ICMP echo request yielding an availability of 96.45%. There were 5,278 hosts in the sample that could be identified as VAXen. Of these, 3,482 answered the ICMP echo request. The resultant availability of 65.97% is in sharp contrast to the availability of 96.45% reported for hosts that answered the Sun RPC request. However, if the data are examined closely, some sites are found to report a very large number of hosts that did not respond[6] to the ICMP echo request. There are several explanations for this discrepancy. It may be that a large number of hosts are "hidden" behind a bridge that does not forward the ICMP packets. A second possibility is that these hosts do not exist, but are planned to be installed at some later date, and so entries have been allocated for them in the name space. A third possibility is that some of these hosts may have been decommissioned and replaced with newer equipment. This underscores the desirability of polling only those hosts that responded to the initial sampling: such hosts are known to have been operational in the recent past and so are unlikely to be on isolated networks or have been decommissioned.

7 Estimating Mean-time-to-repair

The MTTR of a host can be estimated using information derived in previous sections. In particular, if the MTTF and the availability are known, then MTTR can be estimated using the dependencies derived from a simple steady-state Markov model [17]. A rigorous application of this model requires exponential failure and repair distributions, and the steady-state assumption must be justified.

A host can be modeled by the explicit set of differential equations,

$$\frac{dp(t)}{dt} = \mu q(t) - \lambda p(t), \frac{dq(t)}{dt} = \lambda p(t) - \mu q(t)$$

with initial conditions $p(0) = 1, q(0) = 0$. Here $p(t)$ is the probability of the host being in an operational state at time t, and $q(t)$ is the probability of it being in a failed state. The

[6] For example, 444 VAXen from MIT.EDU did not respond and 191 from Berkeley.EDU did not respond.

Table 4: Availability of Specific Sun 4 Systems.

Model	n	Responses	% Available
4/60	1047	975	93.12
4/65	36	34	94.44
4/110	136	126	92.65
4/280	144	139	96.53
4/330	38	38	100.00
4/390	24	23	95.83

explained by the age of these systems, which are nearing the end of their useful life. They may be becoming more prone to failure, relegated to tasks that minimize the need to quickly restore them to service, or may even have been taken out of service during the months between the first and second polling phases. By examining the raw data, it was noted that several large clusters of these systems were down, as was their corresponding server. The Sun 3/80 systems also show a surprisingly low availability. This may be attributable to the small sample size, and repeated polls should be made before any strong conclusions are made.

Table 5: Availability of Specific Sun 3 Systems.

Model	n	Responses	% Available
3/50	1486	1273	85.67
3/60	876	815	93.04
3/80	199	177	88.94
3/180	125	122	98.40
3/280	206	197	95.63

Other systems were also considered and the results are summarized in table 6. Significantly different availability figures are obtained when systems other than those that responded to the initial sampling are considered. In some cases, this is due to the small number of hosts in the sample. There were 33 Sequent[5] hosts polled. Of these, 31 answered the ICMP echo request yielding an availability of 93.94%. There were 71 Pyramid systems polled. Of these, 60 answered the ICMP echo request. Of the 11 that did not respond, 5 were at one site. The resultant 84.51% availability is probably a low estimate since these hosts may have been permanently taken out of service or could have become isolated from the network.

The VAX systems provide a more telling example. There were 169 VAX systems that responded to the initial sampling. The number is small since few VAX system support

[5] The Sequent host `uunet.uu.net` appeared as a name server for 93 sites but was only counted once.

Table 3: Mean-Time-to-Failure for Various Systems.

Model	n	min (days)	max (days)	$\bar{x}$ (days)	s
Sun 386i	176	0.070	107.395	19.341	21.150
Sun 4	1735	0.009	233.864	15.679	22.499
Sun 3	4475	0.002	330.833	17.734	25.145
Suns	7205	0.002	330.833	17.115	24.109
VAX	122	0.017	160.534	13.018	20.527
VAX (VMS)	21	0.415	46.282	8.553	12.989

6 Availability

The availability of a host is an important measure, indicating the probability that a host will be accessible. Some significant differences were noted for some classes of hosts.

The initial sequence of queries using Sun RPC was used to construct a list of known hosts. Several months after the initial sampling, all of the responding hosts were again polled using the ICMP echo protocol. This two-phase method guarded against incorrectly attributing the absence of a response to a failure when the host might be permanently unreachable or even non-existent.

The two phases are important, but unfortunately this method is slightly biased against hosts with poor availabilities, as such hosts were more likely to be unnoticed during the initial sampling. This bias can be minimized by extensive polling at various intervals during the first phase, to ensure that most existing hosts are marked for participation in the second phase. In this study, the first phase was limited to one week due to time constraints and the considerable network traffic it generated. The large time interval between the first and second phases is necessary to ensure that there is negligible correlation between a host being up in the first phase and being up in the second phase.

The servers, such as the Sun 4/280, 4/330, 4/390, 3/180 and 3/280 showed a uniformly higher availability than the work stations. This is to be expected since servers are more likely to be maintained by a staff person, and less likely to be shut down when the user leaves in the evening.

The results for Sun 4 systems are summarized in table 4. The largest sample is for the Sun 4/60, which yields an availability of 93.12%. The inaccuracy of small samples is best illustrated by the Sun 4/330 servers. Only 38 were polled in the second phase, and all of them responded, yielding a misleading 100% availability. When the available population is small, the only recourse is to repeatedly poll those hosts over an extended period of time until the desired degree of confidence can be obtained. Due to time constraints, this could not be undertaken in this study: to ensure independent samples, the time between polling attempts must be very large.

The results from Sun 3 systems are summarized in table 5. The largest sample is for the Sun 3/50, but the availability reported is unexpectedly low. This may be partially essentially a Sun 4/60 with a faster clock rate. The reason for this low value is most likely the small sample size. With only 39 hosts, a few recently initialized hosts can greatly reduce the average.

The small sample size does not seem to affect the MTTF for the servers such as the Sun 4/280, 4/330 and 4/390. The reason for this is most likely because servers are independent. When a server is disabled, often the work stations that it serves are disabled as well. In contrast, disabling one server usually does not imply the disablement of other servers at that site.

Table 1: Mean-Time-to-Failure for Specific Sun 4 systems.

Model	n	min (days)	max (days)	$\bar{x}$ (days)	s
4/60	1116	0.009	233.864	16.831	24.259
4/65	39	0.164	31.627	7.340	6.773
4/110	159	0.070	90.363	14.818	17.752
4/280	162	0.019	112.150	13.806	17.921
4/330	41	0.247	153.856	14.811	26.965
4/390	31	0.025	100.295	12.317	21.576

The results for specific Sun 3 systems are reported in table 2. The largest samples reported are for the Sun 3/50 and Sun 3/60. Again, a suspiciously large value of over 330 days is reported for a Sun 3/50. While this value could not be proven false, the frequency of such values was low enough to have little impact on the average.

Table 2: Mean-Time-to-Failure for Specific Sun 3 systems.

Model	n	min (days)	max (days)	$\bar{x}$ (days)	s
3/50	1730	0.036	330.833	18.687	27.445
3/60	1026	0.006	245.134	18.866	24.164
3/80	242	0.056	148.041	14.707	21.122
3/180	136	0.014	120.378	13.255	19.439
3/280	223	0.017	118.952	16.565	21.130

The results for various systems are summarized in table 3. There were many hosts that could not be precisely classified. For example, there were $4,475$ hosts that could be identified as Sun 3 systems, but only $3,375$ of these could be classified by specific model.

A small number of VAXen responded to the Sun RPC request. In general, the MTTF reported closely matches the values reported by the Sun systems. Of these VAXen, 21 were running the VMS operating system. The MTTF reported by these systems is approximately 50% of that reported by the other systems, although this may be attributable to the small sample size.

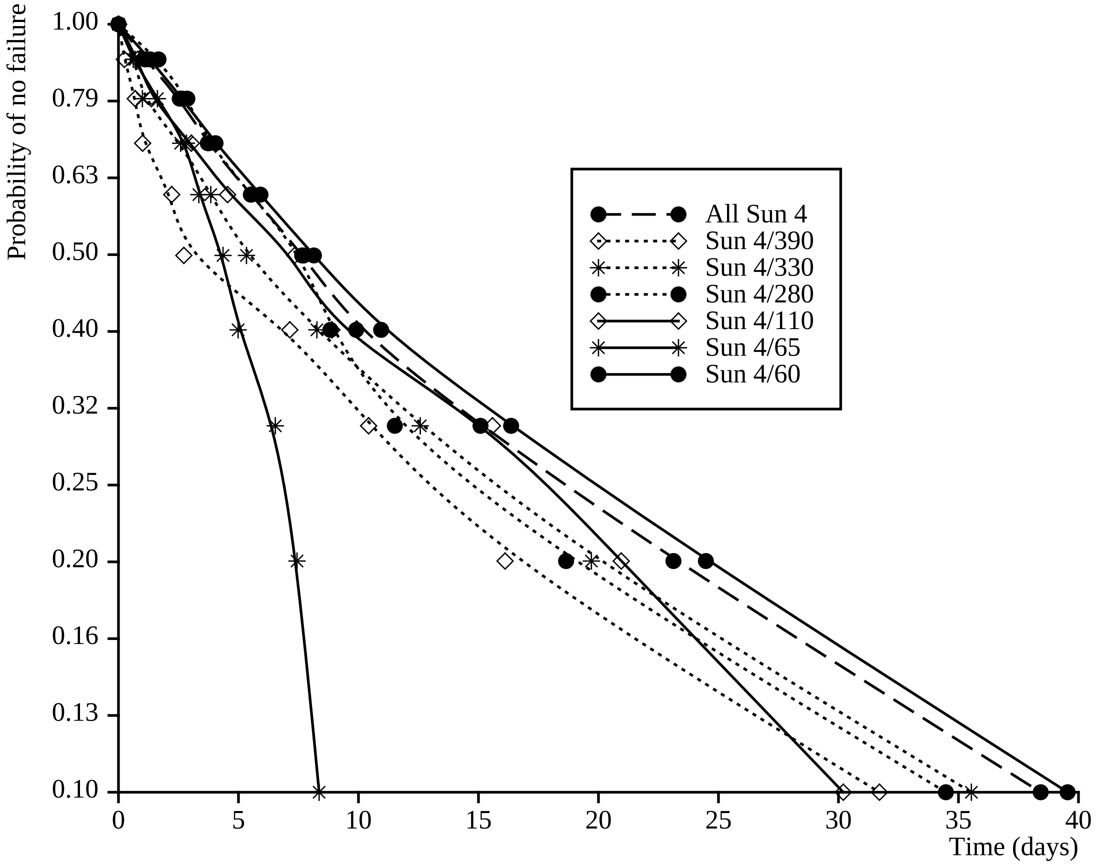

Figure 3: Semi-logarithmic Graph of Up-times for Sun 4 Systems.

the actual MTTF and closely match those seen in practice[4] by system administrators.

A summary of the results are given in tables 1 through 3. The columns give the model, the size of the sample, the minimum and maximum reported values, the mean and the standard deviation.

Table 1 summarizes the results for hosts that could be identified as a specific model of Sun 4. The Sun 4/60 data points comprise the largest sample and should produce the most accurate estimate of MTTF. The large standard deviation can be traced to the large up-time values reported by a few hosts. While is it unlikely that any Sun 4/60 has been continuously operating for 234 days, this value could not be proven false. However, there were many cases where the value reported could be proven false. In particular, an apparent error in `rpc.statd` sometimes causes it to report an up-time on the order of 7,000 days, while Sun Microsystems has been in business less than half that length of time.

The MTTF reported for the Sun 4/65 is suspiciously low, since the Sun 4/65 is

[4] Several system administrators were contacted and shown the results. All agreed that the values were close to what they expected, although some thought that they were slightly too low, while others thought that they were slightly too high.

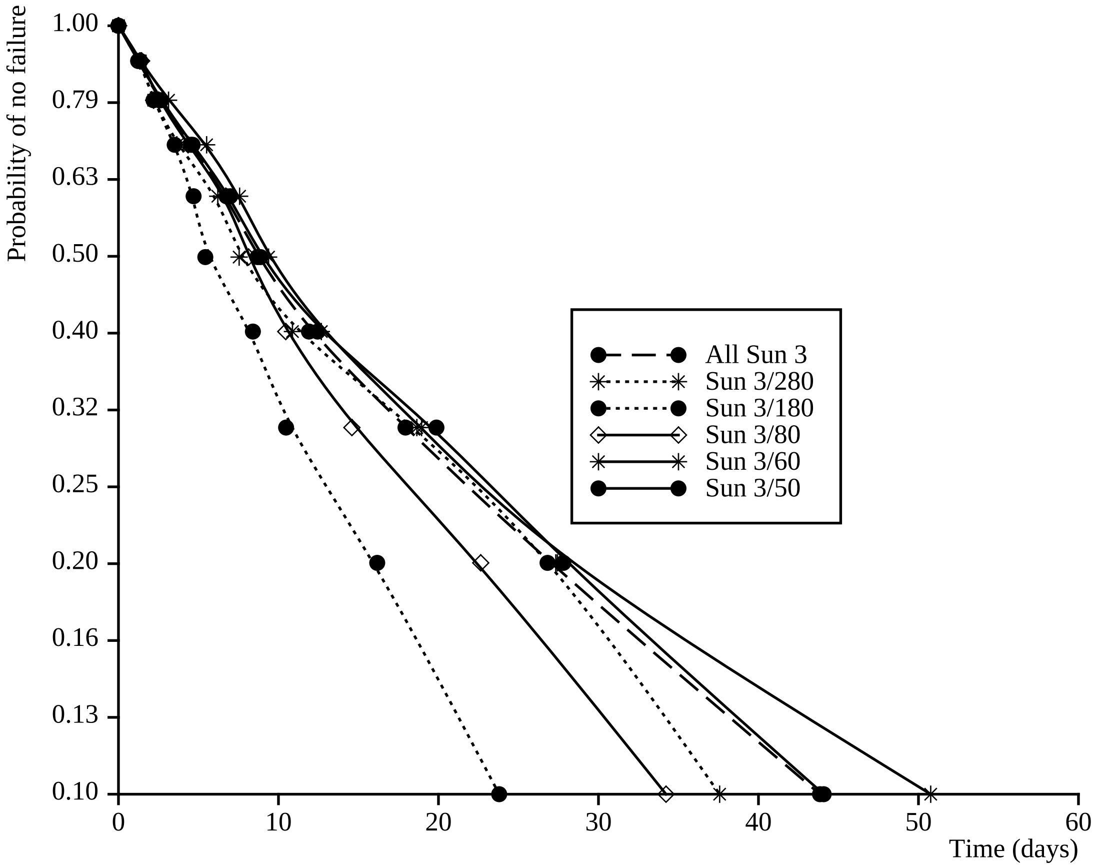

Figure 2: Semi-logarithmic Graph of Up-times for Sun 3 Systems.

all data that could be associated with a model designation. The extremely sparse data in the last decile is not shown in the figure. On this semi-logarithmic scale, a perfect exponential curve would follow a straight line. Sample sizes for the Sun data on which these figures are based are shown in tables 1 and 2. Some of the sample populations are too small to accurately reflect the shape of the underlying distribution, but even the curves of the larger samples are not straight. The test statistic of each of these is sufficiently large to confidently reject the hypothesis that the samples are drawn from exponential distributions.

5 Estimating Mean-time-to-failure

As discussed in §3, when the time to failure is exponentially distributed, its distribution agrees in both shape and mean with that of the up-time reported by Sun RPC. When the failure distribution is approximately exponential, the mean reported by Sun RPC provides an approximation of the MTTF. Although it is unlikely that the sample was drawn from an exponential distribution, the averages obtained are a reasonable approximation of

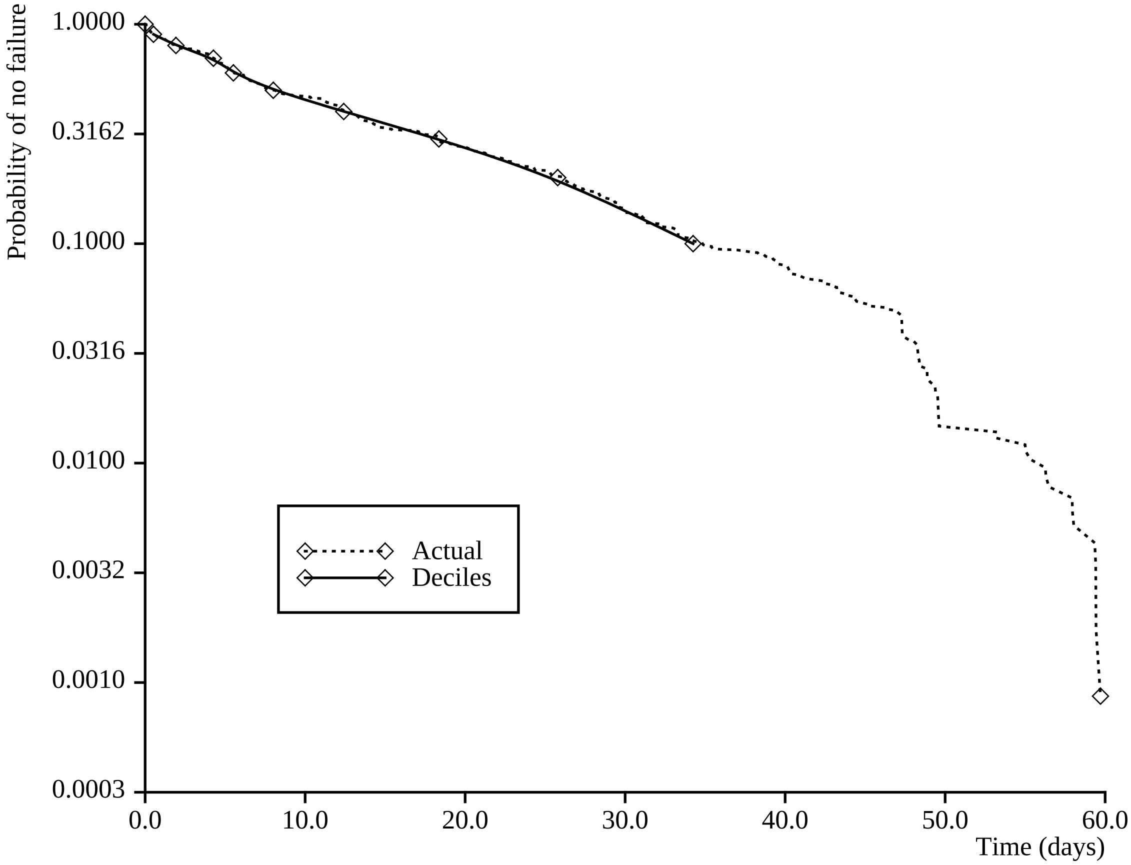

Figure 1: Semi-logarithmic Graph of Up-times for 1,154 Random Hosts.

for several reasons. The data includes some reports of hosts being up longer than their underlying hardware has been in existence; this seems unlikely.

There is reason to suspect that some of the extremely large numbers reported are attributable to a defect in Sun RPC. A curiously large number of hosts reported up-times close to 20 years. These hosts were institutionally and geographically diverse, and hence the evidence points to an anomaly in the software that generates those numbers.

More importantly, the collection is not comprised of truly independent samples, as it is quite common to find entire sets of clients that are reinitialized within minutes of each other. This naturally occurs as a result of the failure of a common server.

In an attempt to ensure independent samples, one set of test data was built by taking at most one datum from each second-level domain. The resultant sample populations had test statistics that indicated a higher probability of exponential behavior, but most could still be distinguished from true exponential distributions.

As shown in §5, the MTTF of the different classes of hardware are distinct. This would suggest that the behavior of the hosts comprising the Internet might best be modelled as a sum of exponentials. Such a hyperexponential distribution would also exhibit straight-line behavior on a semi-logarithmic scale, similar to that in figure 1.

Figures 2 and 3 illustrate the distribution shapes of various Sun models, based on

4 Testing the Exponential Hypothesis

The cumulative distribution determined from the initial sampling suggested that the underlying distribution is either exponential or a mixture of exponentials. As shown in figure 1, the graph of the logarithm of these values is remarkably straight. By designing an appropriate test statistic, the hypothesis that the sample values came from a single exponential distribution can be tested. A test statistic based on the parametric family of distributions with linear failure rate density has been shown to be applicable to a large class of nonparametric distributions as well, and has been shown to be applicable to machine behavior [3]. For n samples t_1 through t_n with mean $\bar{t}$, the test statistic is given by

$$T = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \left[1 - \frac{1}{2}(t_i/\bar{t}\,)^2 \right] .$$

With H_0 representing the hypothesis that the points come from a single exponential distribution, H_0 can be rejected with a confidence level based on the value of the equivalent formula

$$T = \frac{1}{2}\sqrt{n} \left[1 - \frac{\hat{\sigma}^2}{\bar{t}^2} \right] ,$$

where $\hat{\sigma}^2$ is the sample variance.

If a population distribution is known to have a linear failure rate density, then large values of T indicate that H_0 can be rejected with a significance probability based on the standard normal distribution. Population distributions that have a nondecreasing failure rate average can also reject H_0 with the same significance probability for large n [3].

No matter how large the sample size, no amount of testing can assure that a population distribution is exponential. By contrast, the test statistic T can quantify the prohibitively small probability that certain samples were derived from an exponential population distribution.

The analysis of an initial sampling of over $1,000$ host responses was instructive. The sample mean was 15 days, and the median was 7.5 days. The raw data failed the exponentiality test rather spectacularly with a test statistic of 6.6; if the sample is drawn from an exponential distribution, the probability of observing a test statistic value as large as 6.6 is far less than one in ten thousand. Since several hosts were known to more than one domain server, approximately three percent of the sample points were repetitive; an additional two percent advertised improbably large up-times. When these repetitions were consolidated and the large numbers purged, the test statistic based on the modified sample shrank to 0.4. Under the assumption that the extremely large numbers are invalid data, the evidence against the exponential hypothesis is much weaker. If this modified sample is controlled by an exponential distribution, the probability of observing a test statistic value as large as 0.4 is just 2 in 3. This extremely small test statistic value provides strong evidence of the exponential nature of the sample.

Testing the collection of all $12,987$ raw data points shows that V for this larger set of samples is definitely not exponentially distributed. This is certainly not surprising, function of V given a fixed sampling interval $L_0 = l_0$. This function is given by

$$f_V(v|L_0 = l_0) = \begin{cases} \frac{1}{l_0} & 0 < v \leq l_0 \\ 0 & l_0 < v \end{cases} .$$

The marginal probability density function of V is

$$g_V(v) = \int_0^\infty f_V(v|L_0 = l_0) f_{L_0}(l_0) dl_0 = \int_v^\infty \frac{1}{l_0} f_{L_0}(l_0) dl_0. \tag{1}$$

The distribution of L_0 can be related to X, which denotes the length of the interval between reinitialization and the next failure of the host. It is well known [2, 4] that the probability density function of L_0 can be written in terms of the probability density function of X,

$$f_{L_0}(x)dx = \frac{x f_X(x) dx}{E[X]}.$$

Thus, using this in equation 1,

$$g_V(v) = \frac{1}{E[X]} \int_v^\infty f_X(x) dx.$$

Now consider

$$\Pr[V > v_0] = \int_{v_0}^\infty g_V(v) dv = \frac{1}{E[X]} \int_{v_0}^\infty \left[\int_v^\infty f_X(x) dx \right] dv = \frac{1}{E[X]} \int_{v_0}^\infty \Pr[X \geq v] dv.$$

If X is assumed to have an exponential distribution with mean $\frac{1}{\lambda}$, then

$$\Pr[V > v_0] = \frac{1}{E[X]} \int_{v_0}^\infty e^{-\lambda v} dv = \frac{1}{E[X]} \left[-\frac{1}{\lambda} e^{-\lambda v_0} \right]_{v_0}^\infty = \left[\frac{\frac{1}{\lambda}}{E[X]} \right] e^{-\lambda v_0} = e^{-\lambda v_0}. \tag{2}$$

Under these assumptions, equation 2 implies that $E[X]$ matches the sample mean for V. Intuitively, the degree to which X should exceed V is exactly counterbalanced by the length-biased sampling of V. Indeed, V will also be exponentially distributed with mean $\frac{1}{\lambda}$.

The contrapositive of this implication ensures that if V is not exponentially distributed, then neither is X. Since V is readily observable, a much richer sampling can be tested for exponentiality. If it is found that it is highly improbable that V is drawn from an exponential distribution, this constitutes strong evidence that X is also not controlled by an exponential distribution.

In the case where X is exponentially distributed, the distribution of V matches X in both shape and mean. It is reasonable to assume that the shape of X approximates that of V when the distributions are approximately exponential. This implies that in typical cases, a sampling of up-time reports can be treated as a sampling of times-to-failure. This correspondence is the foundation of the estimation of the MTTF in § 5.

the Sun RPC protocol. This case is indistinguishable from a non-operational host, since UDP is a connectionless protocol and no response is returned if there is no server there to reply. For hosts that were identified by their domain server as Suns, a lack of response was interpreted as a failed host.

A fourth possibility is that a host with Sun RCP may have `rpc.statd` disabled. Such a host will decline to respond, but this is distinguishable from a failure and so could be safely discounted.

Initially, responses from a random sampling of over $1,000$ hosts were gathered to determine that the MTTF of a typical Internet host was on the order of 15 days. The plot of the sample cumulative distribution bore a striking resemblance to an exponential distribution. This hypothesis was tested in § 4, and these tests [3] showed that while some data collected did indeed fit this pattern, other data did not.

Few hosts were found to report up-times of greater than 60 days. This observation mandated a sizable delay of several months beyond the initial polling, after which all hosts were again queried using the ICMP echo protocol. This allowed a nearly independent sampling to be made in order to estimate the average host availability. While repeated polling could have been used to increase the confidence in the availability data, it was not attempted for this study due to time constraints.

The most accurate availability data was obtained from the hosts that responded to the original Sun RPC requests. These hosts were known to be reachable at some time in the past, and so were less likely to be on isolated network segments or otherwise unreachable.

An interesting complication arises because an unavailable host is not necessarily a host that has failed. A null response could instead be caused by a network failure. For many applications, current availability is indeed the proper measure, and the data supplied by ICMP is directly correlated to the expected availability of a typical host. The distinction is important when inferring absolute availability, but can be largely ignored when comparing the relative availability of different classes of hosts. The distinction is also irrelevant when local availability, rather than availability across the Internet, is analyzed.

3 Length-biased Sampling

Randomly sampling the length of time since the last system initialization is distinct from sampling the length of time between initialization and failure. Sampling system up-time reports results in a skewed set of data, as hosts which have been up the longest are more likely to be polled. Analysis of the data must accommodate this effect. Let the length of the time interval from the reinitialization of a host until its next failure be denoted by the random variable X. This quantity is not directly observable, but must be inferred from data that is available. Let V represent the interval spanning the time between the last initialization and the time the sample was obtained. This value is observable, and can be obtained using Sun RPC.

Analysis of the shape of the distribution representing X must be undertaken in light of the length-biased sampling of V. Let $f_V(v|L_0 = l_0)$ the conditional probability density

2 Data Acquisition and Reduction

To acquire information about the status of Internet hosts, a list of top-level domains was obtained from the network information center using the "who is" service [7]. The domain servers for each of these top-level domains were then queried for lists of hosts at each site. There are two significant problems with this approach. First, lists of secondary domains appear to be more difficult to obtain, and so only hosts known to the primary domain servers for each site were considered. For example, MIT's Laboratory for Computer Science alone has 584 hosts in the subdomain[3] `LCS.MIT.EDU`, but a list of similar secondary domains for other sites was unavailable. Second, the domain servers for several large sites such as Stanford University and Carnegie-Mellon University would not provide lists of hosts. This was disappointing since these sites in particular have large numbers of hosts that could have answered queries. Still, even considering only hosts in the primary domain, the lists provided by cooperative domain servers contained over $100,000$ hosts. Responses were received from almost $13,000$ of these, providing a wealth of data for statistical analysis.

Once lists of hosts were obtained, the domain servers were again queried to determine the type and operating system of each host. This information was useful in analyzing the system status information returned by each operational host. Again, there are two significant difficulties to this approach. First, there are some sites that do not provide any host-specific information. For example, the University of California at San Diego has $2,663$ hosts, including 287 that answered queries for system status. The information provided by these hosts is of little value due to the unfortunate lack of host-specific information. The second, and possibly most challenging, problem stemmed from the many ways a system administrator may describe a host. This made it difficult to precisely identify the manufacturer and model of queried hosts. Often it was impossible to determine more than the manufacturer, or the processor family.

Once lists of hosts were collected for a large number of sites, data were gathered by polling each host using Sun RPC [16] to determine the system status, including the length of time it had been up. A time-out period of 15 seconds was chosen, since a typical round trip time for an ICMP echo request is less than 1 second. Sun RPC uses UDP [13], which, like ICMP, is layered on top of IP [15]. The remaining 14 seconds was judged to be sufficient for the host to respond to the request. If the response was not received within that time window, then the host was deemed to be too heavily loaded to be considered available.

A host that does not respond to the Sun RPC request may indeed be unavailable, but there are also other reasons for not receiving the desired response. The host may have failed, or may be unavailable due to a network failure. These two failure modes are often distinguishable: the IP protocol will often notify the polling process of an unreachable network, and even if it does not, the network failure would also be manifested as a cluster of unresponsive hosts. Surprisingly, only a few network failures occurred during the seven day polling period.

A third possibility is that the host is reachable and available, but does not understand

[3] Once the domain name is known, obtaining a list of hosts is simple.

The exponential hypothesis is rigorously testable, although the process of gathering the data and the problem of interpreting it are non-trivial. Two of the important statistics that are derived are *mean-time-to-failure* (MTTF) and *mean-time-to-repair* (MTTR). MTTF is not directly available from hosts, but it can be estimated using the length of time that hosts have been up, provided that the pattern of up-times is governed by an exponential distribution. The problem of determining the length of time that a host has been down is an obvious example of indirect data acquisition, since a failed host is not in the position to immediately report its demise. Using the estimate of MTTF, along with the average host availability, MTTR can be derived.

For this study, data were collected from as many hosts as was practical and then used to derive estimates of availability, MTTF and MTTR. While it might have been possible to install monitors at a large number of sites, it was impractical to solicit the cooperation of the hundreds of system administrators[1] necessary to gather the desired data. Instead, the analysis was done using only data that could be obtained using the Internet[2] with no special privileges or added monitoring facilities. This was principally done by polling hosts using Sun RPC [16] to query `rpc.statd`, the ICMP echo protocol [14] to test availability, and by polling domain servers [9, 10] to obtain host-specific information. A surprisingly rich collection of information can be gathered in this fashion, allowing several important parameters to be estimated.

Availability is difficult to estimate accurately using the Internet. This is due to the many possible reasons for a host not responding to a request, most of which are indistinguishable to the polling process. Among these are the host being down, the host not implementing the polling protocol, and both hard and soft network failures.

Analyses of MTTF and the causes of failure have usually been confined to specific systems. Recent studies include analyses of Tandem systems [5, 6] and the IBM/XA system [11]. Research covering heterogeneous systems is less common. The difficulty in assembling sufficient data and applying the appropriate statistical tests has inhibited a thorough analysis of the shape of the failure distribution. The failure rate distributions of several common classes of architectures are analyzed, and estimates of their MTTF are derived. No attempt has been made to characterize the causes of failure, though it seems that most failures are brief and are caused by software faults.

The method of data acquisition and the problems encountered in its reduction are described in §2. The host MTTF can be derived from the *up-times* reported by each host by using the *length-biased sampling technique* described in §3. Success depends on the exponential nature of the data, a hypothesis which is examined in §4. The problem of estimating MTTF is studied in §5, followed by the average host availability in §6. These results are used to derive MTTR in §7. A summary of the results of this study follows in §8.

[1] A large number of system administrators answered the initial call for data. Unfortunately, the data they provided were too few and were often incomplete.

[2] It is interesting that these polling activities, encompassing more than $100,000$ hosts withing the period of a week, elicited inquiries from only four system administrators.

A Study of the Reliability of Internet Sites

D. D. E. Long
Computer & Information Sciences
University of California
Santa Cruz, CA 95064

(408) 459-2616

darrell@cis.ucsc.edu

J. L. Carroll, C. J. Park
Mathematical Sciences
San Diego State University
San Diego, CA 92182

(619) 594-7242, (619) 594-6171

carroll@sdsu.edu, cjpark@sdsu.edu

Abstract

It is often assumed that the failure and repair rates of components are exponentially distributed. This hypothesis is testable for failure rates, though the process of gathering the necessary data and reducing it to a usable form can be difficult. While no amount of testing can prove that a sample is drawn from an exponential distribution, the hypothesis that a population distribution is exponential can in many cases be rejected with confidence.

For this study, data were collected from as many hosts as was feasible using only data that could be obtained via the Internet with no special privileges or added monitoring facilities. The Internet was used to poll over $100,000$ hosts to determine the length of time that each had been up, and again polled after several months to determine average host availability. A surprisingly rich collection of information was gathered in this fashion, allowing estimates of availability, mean-time-to-failure (MTTF) and mean-time-to-repair (MTTR) to be derived. The measurements reported here correspond with common experience and certainly fall in the range of reasonable values.

By applying an appropriate test statistic, some of the samples were found to have a realistic chance of being drawn from an exponential distribution, while others can be confidently classed as non-exponential. With very large sample sizes, sufficient evidence could be accumulated to reject the exponential hypothesis. However, for moderately-sized samples, it was often not possible to exhibit the deviation from exponentiality, lending credence to the common practice of assuming that MTTF is exponentially distributed.

1 Introduction

Many availability and reliability models assume that the failure and repair rates of components are exponentially distributed. This assumption is often made more for analytic simplicity than out of a conviction that it is the best model of reality. Recent studies of replicated data that employ Markov models [12, 8] depend on that assumption.