Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, eds.

SESSIONSIM: ACTIVITY-BASED SESSION GENERATION FOR NETWORK SIMULATION

Lukas Kroc
Cornell University
4106 Upson Hall, Ithaca, NY 14853, USA

Stephan Eidenbenz
Los Alamos National Laboratory
PO Box 1663, Los Alamos, NM 87545, USA

James P. Smith
Los Alamos National Laboratory
PO Box 1663, Los Alamos, NM 87545, USA

ABSTRACT

We present SessionSim, a tool for generating realistic communication sessions such as phone calls, http and email data traffic. Realistic data traffic is a crucial requirement to gauge the realism of any larger communication network simulation study. SessionSim is part of a large-scale communication network simulation environment (MIITS: Multi-scale Integrated Information and Telecommunications System), where detailed information about the individuals in a synthetic population is available, including activities (e.g., sleep, work, lunch) and locations. The key aspect of the SessionSim modeling philosophy is the insight that communication behavior heavily depends on the type of activity people are engaged in; key model parameters in addition to the nature of this dependence are inter-session times, source-destination pairs, and the actual data content that determines session size or duration. We present a mix of empirical data, earlier models and intuition for determining session parameters for phone calls, http and email and briefly discuss validation studies showing that our generated communication sessions adequately mimic real-world data. We also discuss our implementation of SessionSim in the scalable OO-simulation framework called SimCore.

1 INTRODUCTION

Simulation has become a powerful and useful tool for investigating certain problems, especially problems concerning people and their behavior. In some cases, simulations are used because analytical modeling cannot describe a phenomenon under study to the required level of detail (the results are too coarse). In other cases, simulation is simply the cheapest and fastest known way to assess a given problem. The use of simulation in behavior modeling allows to construct many *possible* realistic situations (one that could occur in reality) rather than producing one *average* situation (a usual result of analytical models).

In times of rapidly growing popularity of wireless communication technology, emerging applications for ad-hoc networks, and technology changes in wire-line networks, performance simulations of these systems help to understand their behavior. For example, they allow to find a correct topology of an underlying communication network, to compare different routing schemes used in the network, or to estimate the system's performance in emergency situations. This is true for both data and voice/video applications, which are becoming more and more intertwined. The focus is on simulating the performance of the *underlying network*, because this helps to design and build networks that suit specific needs. But simulation results of a network are more valuable when performed in as realistic settings as possible. This includes imposing proper load on the network—simulating realistic traffic load with appropriate synthetic sessions. We call the process of producing synthetic communication sessions a *session generation*.

The problem of session generation is to create traffic sessions such that they match real usage behavior of people. Therefore, we want to compare the generated data with information from real-world situations, using statistical measures such as the number of calls or data sessions originated in a certain time of the day, call lengths, or the number of sessions per person per day. This paper describes SessionSim, a simulation tool for generating such communication traffic sessions based on detailed (simulated) knowledge of activities of individual people.

We first present the modeling philosophy of SessionSim in Section 2 before developing detailed models for phone calls, http traffic, and email traffic in Section 3. We present the theoretical underpinnings regarding the Poisson processes we use in Section 4. Finally, we describe the implementation details of SessionSim within the SimCore framework in Section 5.

2 MODEL OVERVIEW

We focus on modeling the communication behavior of a *single* user (person), as opposed to modeling combined traffic as it appears on communication network elements (switches, routers, etc.) To do that, we need to find answers to the following questions:

- when does communication occur,
- who communicates with whom,
- and what is being communicated.

We measure the time when communication occurs as time between two communication sessions (inter-session time), finding who communicates with whom is selecting source and destination of a session, and we refer to what is being communicated as the content of the session.

2.1 Model Components

Each of the three components is obtained using a stochastic process. Parameters to such process in general depend on *activity type* (such as work, staying home etc.) of a person whose session is being generated. General description of the processes, including their dependence on activity types, is given bellow and details for each session type are discussed in Section 3.

Inter-session Time The inter-session time can be measured either as time between end of one session and beginning of the next (I_i), or as time between two consecutive session starts (A_i). Which of the two definitions of an inter-session time is used depends on the type of sessions. Sessions with considerable length and that cannot be made in parallel (e.g. phone conversations) are modeled using I_i, while instantaneous session and sessions that can be performed in parallel (e.g. emails or Web browsing) are modeled using A_i .

Let s_i, e_i be start and end times of the ith session, respectively. Then then the inter-sessions times are defined as follows: $A_i = s_i - s_{i-1}$ and $I_i = s_i - e_{i-1}$.

We use Weibull distribution to model inter-session times (both A_i and I_i) of all session types we consider. It is defined with a cumulative distribution function of:

$$F(x) = \begin{cases} 1 - e^{-(x/\beta)^\alpha} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

with *shape parameter* $\alpha > 0$ and *scale parameter* $\beta > 0$ (Law et al. 2000).

Even though the distributions of inter-session times look qualitatively very different for different types of sessions (see Section 3), they can be modeled with Weibull distribution using suitable shape parameter α. Moreover, Weibull distribution arises as an inter-arrival time of Power-law Non-homogeneous Poisson Process, which relevance and properties are discussed in Section 4

For data sessions (emails or Web browsing), probability of having short inter-session times is high. This results in natural clustering of sessions in time, with relatively long periods between the clusters. This corresponds to Weibull distribution with $\alpha < 1$. On the other hand, phone call sessions have low probability of very short inter-session times. Thus, calls are spread in time without a tendency to form clusters. This situation arises for Weibull distribution with $\alpha > 1$. In other words, using the shape parameter α allows us to model the fact that people have a tendency to send emails in bulks, while they wait longer between phone calls.

Source and Destination A source of a session is chosen implicitly as the person who received a session-begin event. Destination is chosen explicitly at the source's side. Since the nature of a destination is different for each session type (another person for phone calls, a web or email server for data sessions), the way of choosing a destination will depend on the session type.

There is, however, a common underlying structure for destination choosing. Each source (person) has a *destination list* for each session type. The destination list consists of pairs of destination identifiers and weights for choosing that identifier. A destination identifier is then interpreted either as the destination itself (e.g. person or server, this will depend on session type), or it may be a special value signifying that extra operations need to be done to find the destination (e.g. choose randomly among all people at work etc).

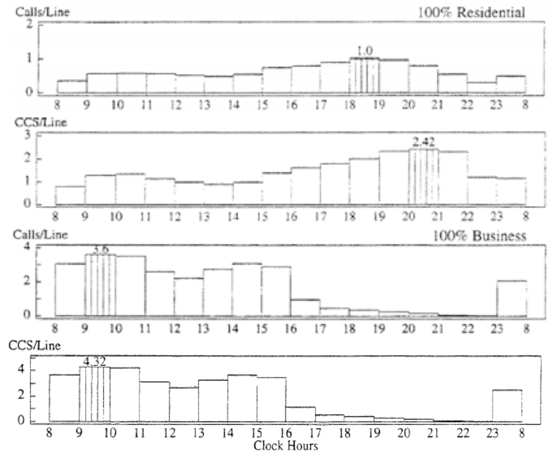

Figure 1: Call intensity for residential (top panels) and business (bottom panels) wire-line phones with respect to the time of day, taken from (Bolotin 1997)

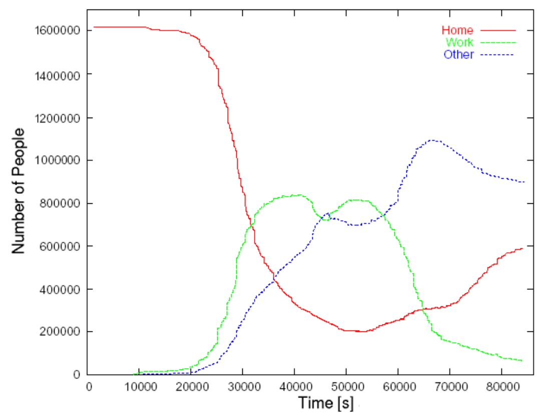

Figure 2: Activities of the generated synthetic population in the Portland study (in 15 minute bins)

Session Content The "what" is being communicated will again depend heavily on the particular session type. It ranges from finding a duration of a phone call, to determining number and sizes of various objects and requests for an http session.

2.2 Activity Dependence

All of the above processes that determine properties of communication sessions may vary with time. We identify one important variation using which we are able to reproduce realistic session intensity curve during a 24 hour period, and that is *activity*. We use three basic categories of activities (activity types): work, sleep, and default (all other activities). The activities are generated with an ActivitySim tool (Galli et al. 2009). Realistic session-intensity curves, such as the one in Figure 1, can be reproduced using different parameters in the inter-session time process for each activity type.

The call intensity curves in Figure 1 are strongly related to work and non-work activities in a simulated population of the city of Portland, Oregon, in Figure 2. This qualitatively justifies the approach of varying inter-session times with activity types. While data often shows that the intensity curve for a *single* user over *many* days itself resembles the curves shown, we reproduce the shapes by generating sessions for *many* users during a *single* day.

Other parts of the session-generation process can be varied likewise. So the destination list could vary, resulting in office workers calling other office workers more frequently than people at home.

3 INDIVIDUAL SESSION MODELS AND EMPIRICAL VALIDATION

In this section, models that were introduced before are made concrete for each of the three session types we model: phone calls, http traffic and emails. Results from using the models are also compared to real-world data that is available to us, or found in the literature. Many of the validation steps are performed by observing the *emergent* behavior of the whole population, because that is what empirical data is available for.

3.1 Phone Calls

The phone call model captures user behavior in making both wire-line and wireless calls. The distinction between the two is not described in this paper, since it is part of device usage modeling, which is not dealt with here. If data was available that would justify making the distinction at the session level, the wireline and wireless calls would use the same model, but with different parameters. As it stands now, we combine bits and pieces of information we find from both worlds to obtain parameter values for the unified model.

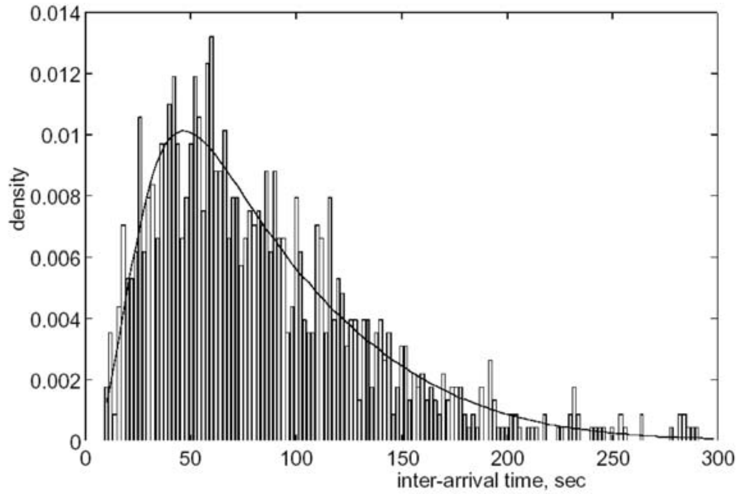

Figure 3: Inter-arrival time for cellular base station with a fit of Erlang-3,8 distributionfor $\rho = 0.6$, taken from (Barcelo 1999)

Inter-session Time The inter-session time in case of phone calls is modeled as time between end of one call and beginning of the next (using the I_i notation). The shape parameter α for the Weibull distribution is chosen to be greater than 1, according to empirically observed distributions. In Figure 3, a distribution of inter-arrival times of calls at a cellular basestation is plotted.

Note that data in Figure 3 differs from our modeling approach in two important ways: it provides a *combined* inter-arrival time (for all users of the basestation), and it plots time between *two consecutive session starts* (so the A_i times, using our inter-session time notation). But we can still use it to draw conclusions about how a distribution of inter-session time should look like in our model. In particular, we see that the distribution is *not* exponential. The fact of having combined data for the empirical distribution does not matter, because superposition of exponential distributions would again yield an exponential distribution. The error obtained by going from A_i to I_i is small, because of the very short length of a phone call compared to a time between calls for a single person. Fitting a Weibull distribution to the Erlang-3,8 data yields the shape parameter $\alpha = 1.8$ for our model.

The scale parameter β determines how frequently calls will be made. We do not have a good source for this type of information. We use 2002 Yankee Group Survey (Yankee Group 2002) to estimate the number of wireless calls per person per day using the number of monthly minutes (165) and average reported wireless call length (3.7 min). This means cca 1.5 wireless calls/person/day. Moreover, from the same survey, we learn that the average percentage of wireline calls replaced by wireless is 28%, so there are about twice as many wireline calls as wireless. This brings us to about 4.5 calls/person/day. From this, we can compute a mean inter-session time of about 320 minutes, which corresponds to scale parameter $\beta = 360$ mins.

The β parameter varies with activity type, so that the resulting number of calls per person will be different than 4.5, and the default β value must be tuned accordingly. We assign (possibly different) value of the $\beta_{default}$ parameter to each person in our model (scale parameter for the default activity). To do that, we use a *social network* of our synthetic population (Galli et al. 2009), which is a graph in which nodes correspond to people and an edge is present between two nodes if the corresponding people somehow know each other. Now following an intuition (we have no data to show it) that people with more social contacts tend to call more often, we compute the individual parameters as

$$\beta_{default} = \frac{\text{average number of social contacts}}{\text{particular number of social contacts}} \cdot \beta$$

(smaller $\beta_{default}$ value means more frequent calls).

From Figure 1, we learn that a peak intensity of business calls is approximately 3.6 times larger than home calls. So the β parameter for work activity is $\beta_{work} = \frac{1}{3.6}\beta_{default}$. (We can do this because the mean of Weibull distribution is linear

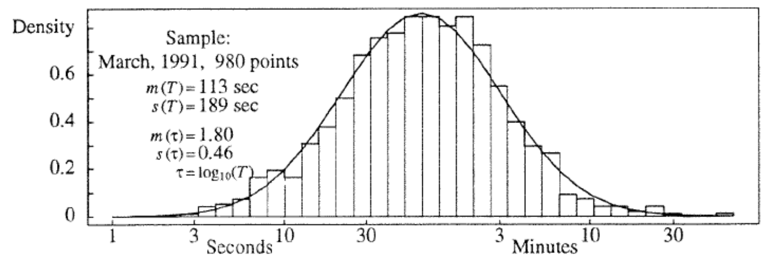

Figure 4: Call-length distribution with a normal fit on logarithmic scale, taken from (Bolotin 1994)

in its β parameter.) We have no data to set the night intensity, so we estimate $\beta_{sleep} = 10 \cdot \beta_{default}$. The shape parameter α does not change with activity type (we have no data to support the change).

Even though we concentrate on modeling exactly one day of communication sessions, it is worthwhile to mention that (Bolotin 1997) suggests that day-to-day call rate distribution can be very well modeled using normal distribution.

Source and Destination Destination lists for phone calls contain individual people that a particular person might call, plus special entries for calling random destination and random people that are currently at work.

The entries of individual contacts are obtained from the social network mentioned above. The weights are proportional to the strengths of the social contacts (a weight on the edges in the network, e.g. time duration of the contact). We have no data to set weights of the special entries, so we used the following values: 10% of random calls, and 20% of calls to random workers during work activity (no calls to random workers during other activities).

The difference in calling random workers mentioned above is the only place where destination lists differ with activity.

Session Content Content of a phone call session is defined only by its length. Figure 4 from (Bolotin 1994) suggests that a log-normal distribution represents a very nice fit for wireline call lengths. The log-normal fit (or combination of log-normals) for session lengths is also suggested for other scenarios (Barcelo et al. 1997, Jordan et al. 1997). The mean and standard deviation for the call lengths is taken from Figure 4, $m = 113$s and $s = 189$s, respectively.

The activity dependence can be derived from Figure 1 by dividing CCS (hundred Call Seconds, a measure of call intensity). by number of Calls in corresponding time bin. We find that the mean of 113 seconds corresponds best with work activity, while default activity (e.g staying home) has calls of approximately twice the length, 226 seconds. This nicely agrees with the information about wireless calls obtained from the Yankee Group Survey (Yankee Group 2002) discussed above. Parameters for to the log-normal distribution are then $meanlog = 4.07$, $sdlog = 1.15$ for work activity (values given in Figure 4 are for logarithm-10 base log-normal distribution)., and $meanlog = 4.76$, $sdlog = 1.15$ for default activities. In lack of any data to validate it, we do not vary the *sdlog* parameter with activity, so the standard deviation of the resulting call lengths will increase with increased mean.

Length of home phone calls also depends on time, as noted in (Bolotin 1997). This is also apparent from Figure 1 by observing that the ratio between CCS and Calls is not constant (it is more or less constant in case of business calls). This time dependence is not captured by our model.

3.2 Http Traffic

The http sessions that our model describes are sessions generated by people using the World Wide Web service of the Internet. Other applications that may operate using the http protocol (such as streaming (Merwe et al. 2002), web crawlers or other automated http usage) are not captured. The model is based on, and is very similar to, the one presented in (Tran-Gia et al. 2001, Choi et al. 1999).

Inter-session Time In WWW sessions, inter-session time is measured as time between two consecutive session starts (A_i). This approach differs from the one used in (Tran-Gia et al. 2001, Choi et al. 1999), where individual sessions alternate between on/off states. Scheduling a next session right after one starts (as oppose to when it ends) allows for having multiple simultaneous sessions for the same person, which we believe corresponds better to reality.

Fitting Weibull distribution to data from (Mah 1997) and to data we collected ourselves is shown in Figure 5. We use 900 seconds as threshold to distinguish http requests that belong to different sessions (shorter inter-request times are considered to belong to the same session). This technique was suggested in (Tran-Gia et al. 2001). The fitted parameters are $\alpha = 0.76$ and $\beta = 12000$ for (Mah 1997) data, and $\alpha = 0.95$ and $\beta = 5600$ for our dataset, in seconds (K-S and χ^2 goodness-of-fit tests

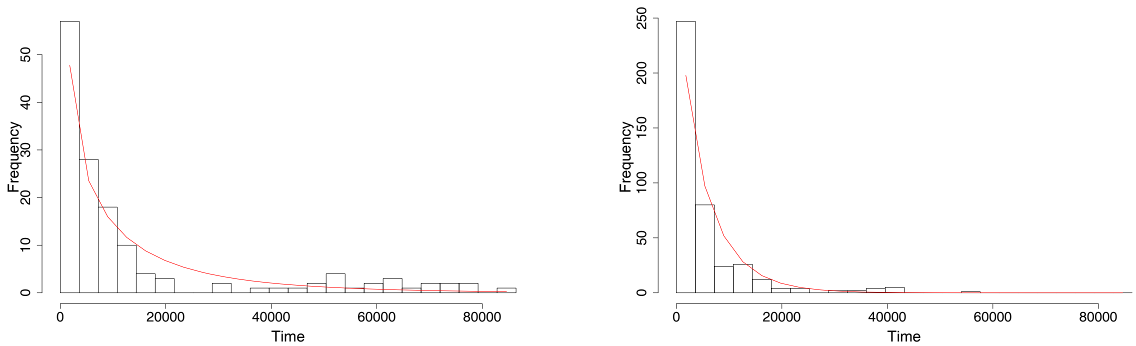

Figure 5: Distribution of time between two consecutive WWW sessions. Histogram with Weibull distribution fit, left panel data taken from (Mah 1997), right panel our data.

both reject it utterly, though). The shape parameters α are similar, while β values differ more significantly. This suggests that the model for WWW inter-session time is similar to the phone calls model. Shape parameter α is help constant for all users, while scale parameter β varies from user to user. Due to lack of data to explore the user dependence, we keep both parameters constant for all users at $\alpha = 0.81$ and $\beta = 6900$ (fit for combined data).

We obtained Internet usage data, see e.g. (Zonalatina 2004), and assuming that the usage is dominated by WWW browsing, the pattern looks remarkably similar to that of phone calls: combination of default/work/sleep activity types with different session intensities for each type will recreate the pattern. The work intensity is about twice the default one, so we set $\beta_{work} = \frac{2}{3} \cdot \beta$ and $\beta_{default} = \frac{4}{3} \cdot \beta$. We set again $\beta_{sleep} = 10 \cdot \beta_{default}$ as in the case of phone calls.

Source and Destination The destination list in case of http traffic contains http servers, along with weights corresponding to their usage. An http server can either be specified in terms of URL, or as an id of a particular server. Note only that while most surveys show ranking of URLs, many URLs may be mapped to go to the same server (e.g. Akamai servers).

The weights for individual web servers should follow the Zip's Law (Mah 1997) (weight of the ith most popular site is proportional to $1/i$). The destination list may also be different for work and default activity, but we have no data to show that.

Session Content Each WWW session consists of several *requests*, and each requests has a primary request and reply, plus several secondary request-reply pairs. The primary request-reply correspond to the main HTML file, and the secondary request-replies pairs correspond to inlined object within that web page. The process of constructing a WWW session content, along with the probability distributions that are used, is outlined bellow:

1. Number of requests in the session: X=lognormal(meanlog=1.8, sdlog=1.68) (from (Tran-Gia et al. 2001))
2. For each request:
   (a) Choose another destination server with prob 0.3 (conforms to consecutive document retrievals measure from (Mah 1997))
   (b) Primary request-reply pair:
      i. Request size [kB]: S=lognormal(meanlog=0, sdlog=0.29) (combining information from (Tran-Gia et al. 2001, Mah 1997))
      ii. Reply size [kB]: M=lognormal(meanlog=1.31, sdmean=1.41) (from (Tran-Gia et al. 2001))
   (c) Number of inlined objects in the request: N=gamma(shape=0.24, scale=23.42) (from (Tran-Gia et al. 2001))
   (d) For each inlined object (secondary request-replies)
      i. Request size [kB]: R=lognormal(meanlog=-1.4, sdlog=0.29) (combining information from (Tran-Gia et al. 2001, Mah 1997))
      ii. Reply size: O=lognormal(meanlog=-0.75, sdlog=2.36) (from (Tran-Gia et al. 2001))
3. Time between requests [s]: V=weibull(shape=0.51, scale=21) (data from (Mah 1997) fitted to median and stddev mentioned in (Choi et al. 1999))

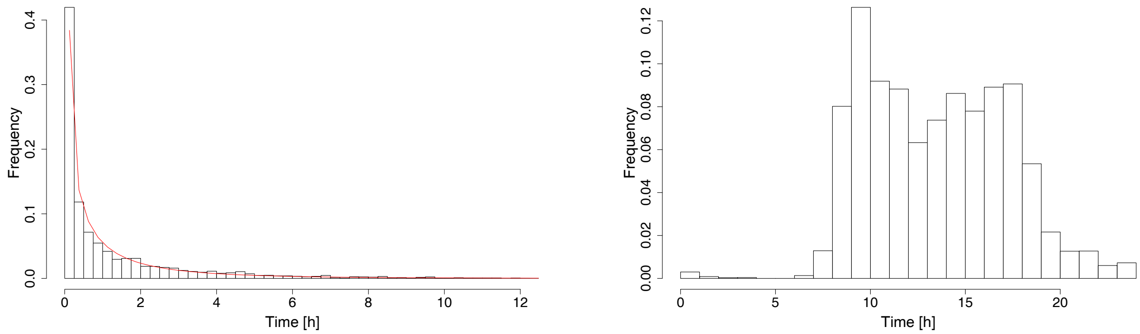

Figure 6: Distribution of inter-session time for emails and its Weibull fit (left panel) and email send intensity during a day (right panel). Obtained from our data (work-related emails).

The number of request that actually will need to be fetched from the original server will depend on the browser's and local network's caching mechanism. Data from (Choi et al. 1999) suggest that there is only a relatively small fraction (less than 10%) of locally cached data. Local network caches (such as corporate proxies) might possibly have a larger amount of prestored data from popular sites. Due to unavailability of data, we do not include any caching mechanism in our model.

3.3 Emails

There is not much email traffic analysis in the literature. A nice overview of what is available is in (Tran-Gia et al. 2001). We also obtained some data ourselves, which we use to estimate parameter values for our model.

Unlike other session types, sending emails is a three-stage process: its sending to a local email server (e.g. SMTP server), transmission to the recipient's email server (e.g. POP3 server), and its download to the recipient. Both first and last stage include modeling of user behavior, but we focus on the first stage. Another popular way of working with emails is via a WWW interface, which we assume to be captured in our WWW session model.

Inter-session Time The inter-session time is again measured by time between two consecutive email sends (A_i). We use our dataset to estimate the parameters for the Weibull distribution, as shown in Figure 6 (left panel). The Weibull fit yields parameters $\alpha = 0.6$ and $\beta = 3000$ (in units of seconds). These values correspond to an average of three users we analyzed, and to work-related emails only. Moreover, we expect that the values obtained will be highly above average due to the nature of their work. We therefore set the average work-related scale parameter to $\beta_{work} = 15000$. We keep the shape parameter $\alpha_{work} = 0.6$ the same, which is also supported by our data (the relative difference in the α values is small in our sample compared to βs). Even though we observe large variance in the scale parameter among different users (1.5K, 3K and 5.2K), we keep it fixed in our model and do not vary it by user. This is due to lack of data to explore the dependence properly.

The right panel of Figure 6 shows that email sending intensity follows again the curve of work activity type. Similarly to WWW sessions, we define the scale parameters for other activity types as follows: $\beta_{default} = 2 \cdot \beta_{work}$ and $\beta_{sleep} = 10 \cdot \beta_{default}$

Source and Destination The destination model is essentially the same as in case of WWW sessions. The destination list consists of email *server* entries (i.e. final recipients full email address is not necessary) along with weights for choosing each.

Emails can possibly have multiple recipients. There is no source in the literature to statistics about this, and our limited sample analysis is inconclusive as to which distribution should be use for this purpose. Therefore, only single recipients are included in our current model.

Session Content Email session content is fully defined in our model by the email size. Following the model in (Brasche et al. 1997), we fit our data to a trimmed Chauchy distribution (because there is a minimal email size of about 0.4KB, all values bellow this are discarded). The email size distribution along with the trimmed Cauchy fit is in Figure 7. The fitted parameter values for the Cauchy distribution are location $= 0.8$ and scale $= 1.4$. As noted in (Tran-Gia et al. 2001), the Cauchy model tends to underestimate fraction of long emails. This is somewhat true also in our case, although our location parameter is larger than the one considered in (Tran-Gia et al. 2001).

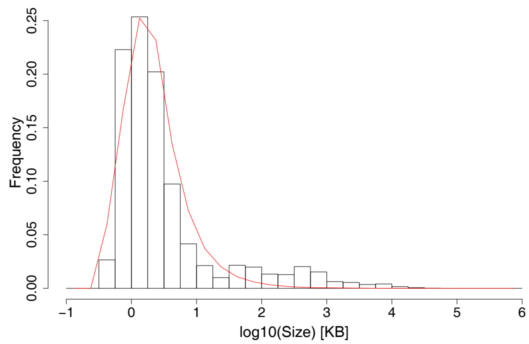

Figure 7: Email size distribution (on log scale) along with its (trimmed) Cauchy fit, from our data

It is hard to regenerate the empirical data in terms of the same mean and standard deviation. This is due to the fact that neither of the measures is defined in the case of Cauchy distribution, so when a sample is drawn from such distribution, the average and sample standard deviation vary significantly. A median (a measure that *is* defined for Cauchy distribution) of the trimmed distribution matches well with the observed data, though.

4 THEORETICAL UNDERPINNINGS

The inter-arrival times A_i are modeled as a renewal process (Frost et al. 1994). This means that A_i are independently identically distributed according to some distribution. While the distribution is different for different session types, it can be very well modeled as Weibull distribution with different shape parameters (see Section 2.1).

The Weibull distribution has some nice properties, which are discussed bellow. It is a distribution of inter-arrival times arising from a non-homohenegous Poisson Process, with an intuitive explanation of the difference in the distribution shapes. This in turn allows us to reschedule already-scheduled events without affecting the overall inter-arrival time distribution. This is important because we have to reschedule future session events whenever an activity of a person changes.

4.1 Power-law Non-homogeneous Poisson Process

A non-homogeneous Poisson process (NHPP) is a Poisson process where the rate λ is a function of time, $\lambda(t)$ (Høyland et al. 1994). The inter-arrival time cumulative distribution function (CDF) is given by:

$$P[A_i \leq t] = 1 - e^{-\int_0^t \lambda(\tau)\mathrm{d}\tau} \tag{1}$$

For $\lambda(t) = \alpha \cdot t^{-\beta}$ ($\alpha > 0$, $\beta < 1$), we obtain a Power-law Non-homogeneous Poisson Process (NIST 2005). Using this in (1), we obtain:

$$P[A_i \leq t] = 1 - e^{-\frac{\alpha}{1-\beta}t^{1-\beta}} \tag{2}$$

which is a CDF of Weibull distribution with a shape parameter $1-\beta$ and scale parameter $\left(\frac{1-\beta}{\alpha}\right)^{\frac{1}{1-\beta}}$

In session simulation, the time t always measures time since the last event. The α parameter sets the initial rate, and β governs how the rate changes in time. When an event occurs, the rate is reset to α (by setting t to 0). For $\beta = 0$, the rate is constant, and the process is therefore a homogeneous Poisson process with exponentially distributed inter-arrival times. For $0 < \beta < 1$, the rate *decreases* in time, and events are therefore more likely to happen soon after a previous event, naturally creating clusters in time (typical for data sessions). For $\beta < 0$, the rate *increases* in time, and consecutive events are therefore separated by longer intervals (typical for phone call sessions).

4.2 Rate Parameter Change with Activities

When a change in activity of a person occurs, different α_{new} and β_{new} parameters may have to be supplied to the rate function $\lambda(t)$. A new event has to be scheduled with the new rate function, replacing the old event. Let T denote the time elapsed since the last event, $T \geq 0$. Since T may now be non-zero, we have to alter (1) appropriately:

$$P_T[A_i \leq t] = 1 - e^{-\int_T^{T+t} \lambda(\tau)\mathrm{d}\tau} = 1 - e^{-\int_0^t \lambda(T+\tau)\mathrm{d}\tau} \tag{3}$$

where t measures time since the rate change.

Plugging α_{new} and β_{new} into the rate function $\lambda(t) = \alpha \cdot t^{\beta}$, we obtain a generalized version of (2):

$$P_T[A_i \leq t] = 1 - e^{-\frac{\alpha}{1-\beta}\left((T+t)^{1-\beta} - T^{1-\beta}\right)} \tag{4}$$

From (4) it follows that the time to the next event after the rate parameter change does *not* depend on the previous rate parameters, but only on the new parameters values and the time of the last event occurrence. For $T > 0$, (4) no longer corresponds to the Weibull distribution.

There is one important property that we need to preserve when rescheduling events. We need to make sure that the probability of an event occurring in a certain interval $(t, t+\Delta]$ does not change by rescheduling the event with the *same* rate parameters. This is captured by the following equality:

$$P_{T_1}[A_i \leq t+\Delta | A_i > t] = P_{T_2}[A_i \leq t+\Delta | A_i > t] \tag{5}$$

for $x \geq T_1 \geq T_2$, where $T_{1,2}$ are two times after the last event occurrence when we are rescheduling the next one. If $x < T$, then this would mean that the event had already happened at time T and we would not be rescheduling it. To show that Equality 5 holds, let $INT(T, t_1, t_2) := -\int_{t_1}^{t_2} \lambda(T+\tau)\mathrm{d}\tau$. Then using a definition of conditional probability and (3), we have:

$$\begin{aligned}
P_T[A_i \leq t+\Delta | A_i > t] &= \frac{1 - e^{INT(T,0,t+\Delta-T)} - \left(1 - e^{INT(T,0,t-T)}\right)}{1 - \left(1 - e^{INT(T,0,t-T)}\right)} \\
&= 1 - e^{INT(T,0,t+\Delta-T) - INT(T,0,t-T)} \\
&= 1 - e^{INT(T,t-T,t+\Delta-T)} \\
&= 1 - e^{INT(0,t,t+\Delta)}
\end{aligned}$$

Since $P_T[A_i \leq t+\Delta | A_i > t]$ does not depend on T, it follows that Equality 5 holds.

5 IMPLEMENTATION OVERVIEW

This section briefly describes the main steps taken when implementing the session generation module as part of a socio-technological simulation suite. At the core of the suite is *SimCore*, a library for building large-scale distributed-memory, discrete event simulations (Kroc et al. 2007) using the discrete event engine from the Parallel Real-time Immersive Modeling Environment (PRIME) for passing events, event queue maintenance, and synchronization. It has previously been used in MIITS (Waupotitsch et al. 2006) for packet-level telecommunications network simulations. The important concepts and classes within SimCore are Entity, Service, Info, and Profile. An Entity is a class that represents a simulation object such as a person, location, or facility. A Service is a class that is used to implement the behavior of an entity and operates like an event handler. Services are attached to Entities. An Info is a class that represents an event that can be scheduled and supplies additional data items and is processed by a Service. Infos are passed between Entities (more typically between the Services) to trigger an action. A Profile is a way of providing runtime specification of default parameter settings for different types of Entities, Services, and Infos.

We call the module responsible for generating realistic communication sessions *SessionSim*, and it is built on SimCore. This allows for relative easy integration with other SimCore modules, such as the network simulator MIITS (Waupotitsch et al. 2006).

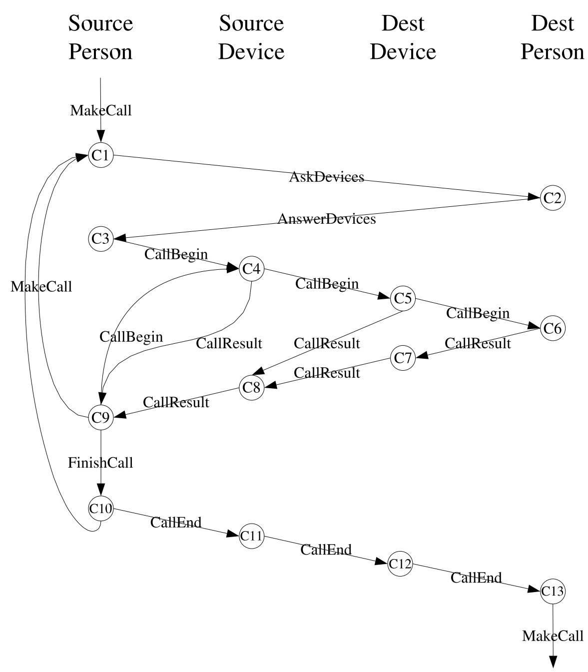

Figure 8: Info-passing map for phone call sessions

5.1 Entities and Services

There are two entity types in SessionSim: Person and Device. Person is a user whose communication behavior is being simulated. Devices are instruments that allow the communication to happen (a phone, a computer). Each communication session (a phone call, an email, ...) is started by a person, and realized by a device (or multiple devices). The relation between Persons and Devices is many-to-many (e.g. many different people can own the same device, and one person can own many devices). The problem of choosing which device(s) to use for a particular session is also part of SessionSim.

In terms of services, there is a Handler service on both Person and Device entity for each session type to be simulated. In addition, there is a ControlService for both entities. The ControlService is responsible for receiving and handling infos about changes of the entity's properties (location, activity, ...), usually send externally from Info input file. It may also provide other general functionality.

5.2 Info-passing Map for Phone Sessions

We will illustrate the detailed working of the simulation by discussing which entities and services exchange which info messages in the case of generating phone call sessions. This type of sessions is the most difficult to generate, as it involves two people. Other communication types (http and email traffic) are similar in spirit, but little simpler to implement (they only involve one person, the other party is a server).

An info-passing map is a graphical representation of which infos are sent from and to which services and entities during some part of the simulation. Columns correspond to different services on different entities involved in the process, bubbles to methods of the service on the corresponding entity that handles the info receiving, and arrows to messages being passed (labels are the info types).

Figure 8 shows the info-passing map for generating phone calls. There are four entities involved in the process: caller and callee person, and their respective devices. Each entity uses only one service for call generation: CallHandler{Person,Device}. The "MakeCall" Info going into "1" in the upper left corner is a starting Info, which is scheduled whenever some parameters of the calling behavior change.

Following is a description of what happens at each step:

C1: signifies that a new call should be made by the person (caller). Can only occur if the person is "idle". Chooses a callee, then asks the callee for his/her devices. Puts the person into "callInit" status.

C2: returns IDs of *all* devices that the person (callee) owns.

C3: (caller) picks devices that will be used, both source and destination device.

C4,C5,C6: determines whether or not the call can be made. It can be made if:

- the source device is at the same location as source person
- source device is "idle"
- destination device is "idle"
- destination person is at the same location as destination device.
- destination person is "idle".

If the call can be made, the `CallBegin` info is passed on and the person or device is put into "callInit" status. If it cannot, a `CallResult` is sent back with info on what happened. On the destination person (C6), `CallResult` is sent back either way, and the person is put into "callDestination" status is successful (and upcoming `MakeCall` info for the callee is invalidated).

C7,C8: if the call was successful, puts the device into "callBusy" status, else puts it into "idle".
This is also the place where an actual network simulator can be plugged in, to determine if the call can be made or if the network cannot handle the request.

C9: caller reacts based on received result:

- If successful, goes into "callSource" status and schedules for itself a `FinishCall` info.
- If there was a problem with the source device, choose another pair of devices with different source device and try again. A `CallBegin` info is sent to the newly chosen source device.
- If there was another problem, schedule yourself new `MakeCall` info.

C10: caller puts itself to status "idle" and sends a `CallEnd` info. It also schedules itself new `MakeCall` info.

C11,C12,C13: puts the entity into "idle" status and pass on. On callee (C13), schedules a new `MakeCall` info.

Other session types are handled in a similar manner and we omit the detailed description.

6 CONCLUSIONS

We describe an approach to session generation that is suitable for large-scale realistic communication studies. The resulting system, called SessionSim, is validated to see that the output resembles realistic traffic. We also outline its implementation in the framework of a large-scale simulation system.

REFERENCES

Barcelo, F., and J. Jordan. 1997. *Channel Holding Time Distribution in Cellular Telephony.* 9th International Conf. On Wireless Communications (Wireless '97), Calgary.

Barcelo, F., and J. I. Sanchez. 1999. *Probability Distribution of the Inter-arrival Time to Cellular Telephony Channels.* Proc. 49th Vehicular Technology Conference, Houston.

Bolotin, V. A. 1994. *Telephone Circuit Holding Time Distributions.* 14th International Teletraffic Congress ITC 14.

Bolotin, V. A. 1997. *New Subscriber Traffic Variability Patterns for Network Traffic Engineering.* 15th International Teletraffic Congress ITC 15.

Brasche, G., and B. Walke. 1997. *Concepts, services, and protocols of the new GSM phase 2+ generalpacket radio service.* IEEE Communications Magazine.

Choi, H., and J. O. Limb. 1999. *A Behavioral Model of Web Traffic.* Seventh Annual International Conference on Network Protocols, Toronto, Canada.

Frost, V. S., and B. Melamed. 1994. *Traffic Modeling For Telecommunications Networks.* IEEE Communications Magazine.

Galli, E., S. Eidenbenz, S. Mniszewski, C. Teuscher, and L. Cuellar. 2009. *ActivitySim: Large-scale Agent-Based Activity Generation for Infrastructure Simulation.* SpringSim-ADS09.

Høyland, A., and M. Rausand. 1994. *System Reliability Theory, Models and Statistical Methods.* Wiley.

Jordan, J., and F. Barcelo. 1997. *Statistical Modeling of Transmission Holding Time in PAMR Systems.* IEEE Globecom 97, Phoenix AZ.

Kroc, L., S. Eidenbenz, and V. Ramaswamy. 2007. *SimCore.* Technical Report 07-0590, Los Alamos National Laboratory. Unclassified Report.

Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling and Analysis, 3rd ed.* Mc-Graw Hill.

Mah, B. 1997. *An Empirical Model of HTTP Network Traffic.* Proceedings of the IEEE Infocom 97, Kobe.

Merwe, J., S. Sen, and C. Kalmanek. 2002. *Streaming Video Traffic: Characterization and Network Impact.* Proceedings of the Seventh International Web Content Caching and Distribution Workshop, Boulder, CO.

NIST/SEMATECH e-Handbook of Statistical Methods. 2005. http://www.itl.nist.gov/div898/handbook/.

PRIME. Parallel Real-time Immersive Modeling Environment (PRIME). http://lynx.cis.fiu.edu:8000/twiki/bin/view/Public/PRIMEProject.

Tran-Gia, P., D. Staehle, and K. Leibnitz. 2001. *Source Traffic Modeling of Wireless Applications.* International Journal of Electronics and Communications (AE), 55.

Waupotitsch, R., S. Eidenbenz, P. Smith, and L. Kroc. 2006. *Multi-scale integrated information an telecommunications system (MIITS): First results from a large-scale end-to-end network simulator.* In *Proceedings of the 2006 Winter Simulation Conference*, eds. L. R. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 2132–2139. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

The Yakee Group. 2002. *Technologically Advanced Family Survey.*

Zonalatina: *Internet Usage By Time of Day,* http://www.zonalatina.com/Zldata386.htm [accessed April 23, 2004].

AUTHOR BIOGRAPHIES

LUKAS KROC has earned his Ph.D. degree in computer science at Cornell University in 2009. His research interests lie in the field of artificial intelligence, with focus on using techniques of probabilistic inference in combinatorial reasoning. Prior to his PhD time he worked at Los Alamos National Laboratory on large scale simulation design and development. Lukas received a MSc in computer science from Charles University, Prague, in 2004. His email address is `<kroc@cs.cornell.edu>`.

STEPHAN EIDENBENZ leads the Socio-technical Modeling&Simulation team in the Information Sciences group at Los Alamos National Laboratory. He has co-authored about 75 articles in the areas of networking, large-scale discrete simulation, agent-based design, transportation modeling, game theory, computational geometry, and computational biology. His current research interests are behavioral agent-based modeling and simulation and communication networks. His email address is `<eidenben@lanl.gov>`.

JAMES P. SMITH is the NISAC (National Infrastructure Simulation and Analysis Center) project leader. He has also been a team leader and group leader in the Discrete Simulation Science Group of the Computing and Computational Sciences Division at Los Alamos National Laboratory, the technical lead for the NISAC EpiSimS project (Epidemiological Simulation System) and the NISAC population mobility project, and a technical staff member contributing to the NISAC telecommunications capability. His current research applies to modeling and simulation of socio-technical systems in national infrastructure, especially public health, telecommunication/computing, and transportation. He has scientific experience in high performance computing and parallel processing applied to large-scale microscopic simulations, including original software design and debugging of very large, evolving systems of inter-operable computational systems, and efficient analysis and synthesis of massive data produced by multiscale complex environments. He also worked for a short time in nuclear theory, for several years in experimental biophysics at the Pennsylvania Muscle Institute, and started a company working in analytic finance. His graduate work included theoretical and experimental fusion and space plasma physics research. He has publications in biophysics, analytic finance, education, space plasma physics, communications, transportation, mathematical biology and computer science, and is a co-inventor on the TRANSIMS patent. He has a Ph.D. in Theoretical Plasma Physics from the University of Texas at Austin. His email address is `<jpsmith@lanl.gov>`.