FLOW VECTOR PREDICTION

Tarem Ahmed and Mark Coates, McGill University

{tahmed, coates}@tsp.ece.mcgill.ca

ABSTRACT

This paper considers the problem of predicting the number, length and distribution of traffic flows some time into the future, based upon packets collected in the present. Three methods– the standard Expectation-Maximization algorithm, a distributed version of the Expectation-Maximization algorithm, and a Particle Filter– are used to predict the mean flow length and complete flow distributions for subsequent timesteps. We propose a model to represent the histogram of flows corresponding to any given time interval, and use the aforementioned methods to estimate the parameters of the model. The proposed algorithms are tested on a large number of commonly-available data traces. The results indicate that the three methods perform comparably well in terms of the distance between the predicted flow distributions and actual flow histograms. An important application of our work is in resource reservation for protocols that require guaranteed qualities of service.

1. INTRODUCTION

Traffic passing through a node corresponds to many different types of applications and protocols. It is important for routers to be able to predict certain characteristics about the nature of network traffic, ahead of time. Quantities of interest include the number of distinct flows within a time period, the lengths of these flows, and the distribution of flow lengths. Flow scheduling is an important factor to consider in efficiently utilizing available bandwidth. Delays from two edge nodes are typically up to 100ms in wide-area networks such as the Agile All-Photonic Network (AAPN). Assuming a frame length of 10ms, this often means that reservation of bandwidth needs to be made at least 10 frames in advance.

We begin by suggesting a suitable model for the flow distribution, and then present three methods for estimating the parameters of the model. The estimated parameters can then be used to predict the number and distribution of flow lengths in future time intervals. We test our methods by comparing the predictions based on our estimates, with the actual flow histograms.

This project was funded by AAPN.

Examples of use of information on flow distribution at core routers include the following:
Resource Reservation: Certain classes and types of traffic require guaranteed qualities of service. Knowledge of the amount of such traffic to expect with a time frame is required to reserve resources in intermediate routers, accordingly.
Sampling Rates: Keeping a record for every packet is infeasible in high-speed routers. Such routers randomly sample packets, and estimate statistics about the original packet stream from the sampled data. The efficiency of the sampling scheme depends on the flow distribution of the original stream [1].
Resource Utilization: Knowledge of the distribution of traffic flows is needed to evaluate gains in the deployment of web proxies [2] and to study efficiency of cache utilization.
Characterizing Source Traffic: Information about the distribution of flows can provide insight into the higher level protocols that the traffic corresponds to (e.g. real-time, per-to-peer, etc.), and help determine thresholds for creating new connections in flow-switched networks [3].
Traffic Engineering: One could use information on flow distribution to balance the total volume of traffic at a core node, based on a small number of identified flows [4]. Moreover, the complexity of optimizing algorithms for multipath routing is reduced if the number of flows is limited [5–7].

1.1. Definitions

An IP *flow* is usually defined to be a set of packets that share a common key, and occur within some period. We define the key to be the following 4-tuple:
key = (*source IP address*, *source port number*, *destination IP address*, *destination port number*)
Thus a flow refers to a connection between specific applications in specific end systems. The *flow length* is defined to be the number of packets that belong to a particular flow (as identified by its key).

In order to compile flow statistics, routers maintain records indexed by flow keys. A flow is said to be *active* if a record exists for its key. Once a new packet arrives at a router, the router first determines if a record is active for the flow, based on the new packet's key. If not, a new record is created with the packet's key. For a record to be active for the arriving

packet's key, the following additional conditions must also hold [1, 8]:
Inter-Packet Timeout: The interval between the arrival of the current packet and the arrival of the last packet having the same flow key, must be below some threshold. If this interval exceeds the threshold, it is assumed that the packet belongs to a new connection between the given (same) (*src IP, src port, dst IP, dst port*) 4-tuple. A flow is said to be *sparse*, if typical times between packets belonging to it exceed the threshold. Sparseness occurs in flows that are long (i.e. many packets), and where the packets are greatly interspersed in (spread over) time. Examples are streaming and multimedia applications [1].
Aging: A flow is also terminated after a given elapsed time between the arrival of the first packet belonging to it, and the current time. This is done in order to prevent data staleness.
Protocol information: A flow is terminated if specific protocol information dictate it, for example a TCP FIN packet is observed corresponding to an active TCP flow.
Equipment limitations: If the measuring equipment require memory to be released and current statistics to be exported.

Our objective for this project is to predict the mean flow length, and the number and distribution of lengths of all active flows, for a future time interval.

1.2. Experimental Data

We obtained packet traces from the Passive Measurement and Analysis (PMA) Project repository at the National Laboratory for Applied Network Research (NLANR) [9]. The traces we used correspond to data collected from the following sources:
trace COS: an OC3c link at Colorado State University;
trace ODU: an OC3c link at Old Dominion University;
trace BWY: an OC3c link at Columbia University.

Figure 1 shows the number of times that flows of different lengths are observed, during a particular 1-second interval at COS.

Certain observations about the distribution of flow lengths can be made from Fig. 1. First of all, the majority of flows are small, with flows of length 1 occurring the most often. The frequency of occurrence of flows of a particular length, initially exhibits a rapidly decaying nature (as evidenced even on the logarithmic scale) with increasing flow length. There also some very-large flows, but they form a very small fraction of the total number of flows observed in the specified time interval. In the particular example of Fig. 1, the number of flows with more than 50 packets was seen to be less than 1%. We observed this general behavior in experimental data obtained from a large number of traces from the NLANR PMA repository. For our research,

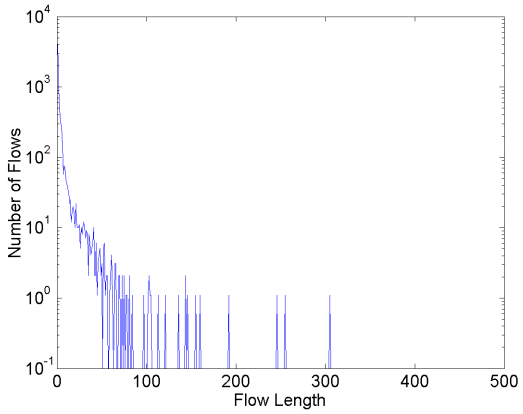

Fig. 1. Sample flow distribution in COS.

we decided to break the histogram of flows into two separate parts, and use different techniques to study the 1% of flows longer than a cutoff, from the 99% flows having lengths shorter than this cutoff. This paper deals primarily with our study of the shorter flows.

It was observed that individual large flows (as indexed by their keys) vary widely in length. Thus, it is extremely difficult to predict them. As large flows contribute a very small fraction of the total number of flows, using averages is sufficient to predict the number of packets contained in these flows.

The traces from COS, ODU and BWY was also used to simulate AAPN-like topologies, and make predictions by specific (src node, dst node) pairs. We divided the source IP space into 16 bins by looking at the number of packets with each source IP, and splitting the source IP space sequentially, so that the number of packets assigned to each source node is evenly spread. Dividing the source IPs randomly did not lead to an even distribution of the packets. Each of these (16) bins was then assigned to a core router, as the IP addresses that it exclusively services.

1.3. Methodology and Outline of Paper

We present the mathematical model (a weighted sum of geometric distributions) used to represent the distribution of the shorter flows, in Section 2. We use the Expectation-Maximization (EM) algorithm to obtain the Maximum Likelihood estimates for the parameters of the model in Section 3. We then modify the centralized EM algorithm from Section 3 into a distributed version, and present the results obtained using the distributed EM algorithm in Section 4. In Section 5, we first motivate the applicability of the Particle Filter, as another method that may be used to estimate the parameters of the model over timesteps. We then develop

and apply the Particle Filter to this problem. We conclude in Section 6 with some suggestions for further research.

1.4. Related Work and Our Contribution

Flow classification by histograms was recently suggested in [10], which argues that simple features such as the mean and variance of flows, provide very little information. Flows need to be aggregated by class, a task which is made difficult by the dynamic and diverse range of behavior exhibited by network traffic. The authors postulate that each flow yields a histogram, which is a realization coming from a stochastic source generating random histograms. Then they propose a mixture of Dirichlet distributions, as the model to represent this stochastic source. They use a stochastic approximation to the Expectation-Maximization (EM) algorithm to estimate the parameters of the model, and finally use the Maximum *A Posteriori* (MAP) principle to designate which class a particular flow belongs to. We use a mixture of geometric distributions as our model to represent a flow histogram. Further, we provide different methods– the standard EM algorithm, a distributed version of the EM algorithm and a Particle Filter– to estimate the parameters of the model, which are subsequently used to predict mean flow lengths and flow histograms in future time intervals.

Flow histograms in wide-area networks are typically such that a small subset of all flows contribute towards a large volume of total network traffic [11]. Such flows are sometimes referred to as "heavy hitters" or "elephants", while the rarely occurring flows are analogously termed "mice" [10]. Our observations agree with these results. Large flows may be identified by two algorithms known as "sample and hold" and "multistage filters" [12]. A classification scheme based on the separation criteria that the elephants must exceed by definition, is given in [13]. Flows are characterized as elephants based on their volume as well as their persistence in time. Thus flows to be termed elephants must contribute significantly to the overall load, and also exhibit sufficient perseverance over time. Our primary objective in this paper is to predict the distribution of the 99% of flows, that are of length less than a cutoff flow length. For the remaining 1% large flows of length greater than the cutoff, we use averages to predict the mean flow length only.

Scaling-based estimators are used to estimate the original flow distribution from characteristics observed in sampled streams, in [1, 14]. They also present a method based on the EM algorithm to infer the flows that missed sampling altogether. Various sampling strategies are described in [15]. We use complete data for the current time interval to predict the flow distribution in subsequent timesteps.

A time-series analysis of network traffic based on Origin-Destination pairs, along with it's use in traffic engineering, is provided in [16]. A comprehensive description of a distributed Expectation-Maximization (EM) algorithm for Gaussian mixtures, and its application to a sensor network, is provided in [17]. We have modified the algorithm to a mixture of geometric distributions, and demonstrated it's application to predicting mean flow lengths and flow distributions in a wide-area network such as the AAPN.

2. THE MODEL

Our initial objective is to propose a model to represent the histogram of all flows below length 50, observed during any specified time interval. Recall from Section 1.2 that 99% of all flows were seen to be of below length 50, for the sample 1-second interval in COS. This behavior was typical in traces obtained from a large number of sources from the NLANR PMA repository. Figure 2 shows the distribution of all flows below length 50 for the same sample 1-second interval in COS. Figure 3 shows the same distribution in linear scale, with flow lengths shown from 1 to 10 only, for the sake of clarity of presentation.

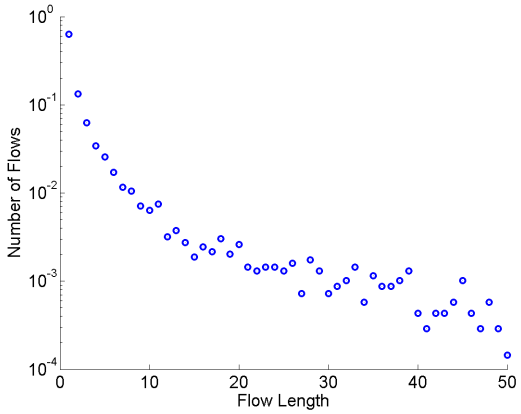

Fig. 2. Flow Distribution in logarithmic scale.

The rapidly decreasing nature of the flow distribution (as evidenced even on the logarithmic representation), suggests that the Number of Flows may be represented by a function that decays with increasing Flow Length. Given the discrete nature of this problem, a geometric probability mass function (pmf) is a possibility. However, it is obvious from Fig. 3 that no particular single geometric parameter (probability) can well explain the occurrence of the entire flow distribution: there is a very large number of flows that contain only 1 packet, the frequency rapidly decreases to about Flow Length = 5, and the rate of decrease is more gradual from then onwards. It thus appears that parts of the distribution require different geometric parameters to describe it well.

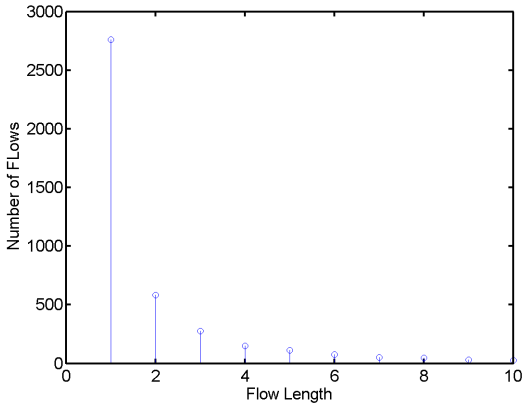

Fig. 3. Flow Distribution in linear scale.

We suggest using a model of the following form:

$$y = \sum_{m=0}^{M-1} \alpha_m (1-p_m) p_m^x \tag{1}$$

such that

$$\sum_{m=0}^{M-1} \alpha_m = 1$$

where y represents the Number of Flows and x represents the Flow Length. Our proposed model is thus a mixture of geometric distributions, with the number of components given by M. To the best of our knowledge, such a formulation has not been adopted elsewhere to predict flow distributions.

It is clear from Figures 2 and 3 that the first spike (corresponding to Flow Length = 1) will need a component in the geometric mixture with a very heavy tail to describe it, compared to the components describing all other flow lengths. It is therefore best to modify the model to include a Dirac delta function centered at $x = 1$ to explain the frequent occurrence of flows of length 1 (i.e. flows with unique keys, containing a solitary packet):

$$y = \alpha_0 \cdot \delta(x-1) + \sum_{m=1}^{M-1} \alpha_m (1-p_m) p_m^x \tag{2}$$

such that

$$\sum_{m=0}^{M-1} \alpha_m = 1$$

The fundamental underlying assumption behind the model is as follows: a particular new flow of length x will be a realization of a geometric probability mass function with parameter (probability) p_m, or a Dirac delta function centered at $x = 1$. The probability that the said flow is a realization of a particular probability mass function (the delta or any component exponential), is given by the mixing coefficients. Thus which component of the mixture a new flow is governed by (according to (2)), is first subject to the mixing probabilities α_m. The probability of the new flow being of length x, is then provided by the already-determined pmf with parameter p_m (or by $\delta(x-1)$ if the observation corresponds to the $m = 0$ case).

3. THE CENTRALIZED EM METHOD

Our objective is to find the Maximum Likelihood (ML) estimates for the parameters of the model proposed in (2), from the histogram of Number of Flows versus Flow Length in the present time interval. ML estimators have the property of being the unbiased estimators with the lowest variance [18, 19]. Once the parameters are obtained, the model may be used to predict flow histograms and mean flow lengths for future time intervals.

The probabilistic model given the set of parameters is as follows:

$$\begin{aligned} &\Pr(x|\alpha_0, ..., \alpha_{M-1}, p_1, ..., p_{M-1}) \\ &= \Pr(x|\Theta) \\ &= \alpha_0 \cdot \delta(x-1) + \sum_{m=1}^{M-1} \alpha_m (1-p_m) p_m^x \end{aligned} \tag{3}$$

where

$$\Theta = \{\alpha_0, ..., \alpha_{M-1}, p_1, ..., p_{M-1}\}$$

represents the set of parameters.

With a sample of $X = \{x_i \cdots x_N\}$ of N independent, identically distributed (i.i.d.) observations of Flow Lengths, and each observation assumed to have been generated by a component density, the Log-Likelihood function [19] is:

$$\begin{aligned} &\log(\mathrm{L}(\Theta|X)) \\ &= \log|\Pi_{i=1}^N \Pr(x_i|\Theta)| \\ &= \log|\Pi_{i=1}^N [\alpha_0 \cdot \delta(x_i-1) + \sum_{m=1}^{M-1} \alpha_m (1-p_m) p_m^x]| \\ &= \sum_{i=1}^N \log|\alpha_0 \cdot \delta(x_i-1) + \sum_{m=1}^{M-1} \alpha_m (1-p_m) p_m^x| \end{aligned} \tag{4}$$

As the log-likelihood function above is not analytically tractable, the Expectation Maximization (EM) method is used to iteratively estimate the parameters. Using standard

procedures for analyzing a mixture model using the EM algorithm [20–23], the Q-function can be derived to be:

$$\begin{aligned} & Q(\Theta^{t+1}, \Theta^t) \\ & = \sum_{m=1}^{M} \sum_{i=1}^{N} \log |\alpha_m| \cdot \Pr(c = m|x_i, \Theta^t) + \\ & \sum_{m=1}^{M} \sum_{i=1}^{N} \log |\Pr(x_i|\Theta_m^t)| \cdot \Pr(c = m|x_i, \Theta^t) \end{aligned} \quad (5)$$

where Θ^{t+1} refers to the updated values for set of parameters to be evaluated from the current values Θ^t, while the additional indicator variable $c \in \{0 \ldots M-1\}$ is thought to designate which particular component density, the *i*th observation of x (out of the $i \in \{1 \ldots N\}$ observations) came from [20–22].

The M-Step equations for the parameters are subsequently derived to be:

$$\alpha_m^{t+1} = \frac{1}{N} \sum_{i=1}^{N} \Pr(c = m|x_i, \Theta^t) \quad (6)$$

for $m \in \{0 \ldots M-1\}$, and

$$p_m^{t+1} = \frac{\sum_{i=1}^{N} \Pr(c = m|\Theta^t) \cdot x_i}{\sum_{i=1}^{N} [\Pr(c = m|x_i, \Theta^t) \cdot x_i + \Pr(c = m|x_i, \Theta^t)]} \quad (7)$$

for $m \in \{1 \ldots M-1\}$. Here,

$$\Theta^t = \{\alpha_0^t, ..., \alpha_{M-1}^t, p_1^t, ..., p_{M-1}^t\}$$

represents the value of the parameters after the *t*th iteration of the algorithm.

3.1. Results with the Centralized EM Method

Figure 4 overlays the flow distribution predicted for the next 1-second time interval based on the distribution from the current interval, and the actual distribution for the next interval. Figure 5 presents the predicted and actual flow distributions on a logarithmic scale. Although we used flow histograms with flow lengths of 1 through 50, Fig. 4 presents only flow lengths 1 to 10 for clarity of presentation. The packet trace is that of the same 1-second interval in COS as used for Figures 2 and 3, and we used $M = 4$. Figure 6 demonstrates the behavior in the tail-region of the corresponding cumulative density function (cdf) of the predicted and actual flow distributions. Figures 4 and 5 show the closeness of fit between the predicted distributions and the actual ones, while Fig. 6 shows that a high percentage of flows are accurately predicted.

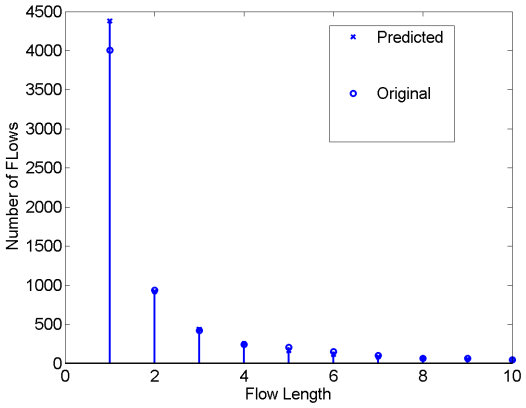

Fig. 4. Predicted flow distribution in linear scale.

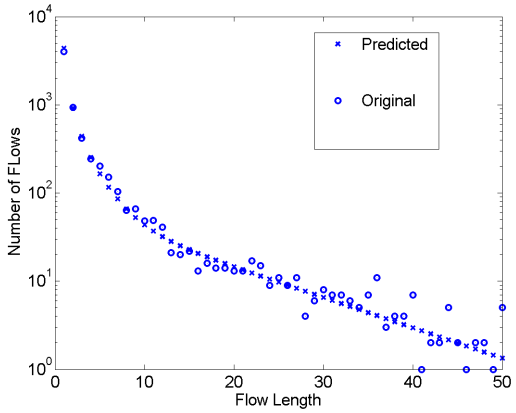

Fig. 5. Predicted flow distribution in logarithmic scale.

We obtained predictions for next timestep's flow length distribution using different values of M, using 90 consecutive 1-second intervals from a large number of packet traces. Table 1 below summarizes the mean L1 differences (over the 90 consecutive 1-second intervals) between the predicted distribution and the empirical one. The packet traces correspond to COS, ODU and BWY.

M	3	4	5	6
COS	0.0685	0.0594	0.0593	0.0591
ODU	0.0900	0.0890	0.0890	0.0889
BWY	0.0823	0.0806	0.0797	0.0795

The mean L1 differences are seen to decrease for each trace (although only marginally so), as more components (M) are used. Using $M = 4$, the mean of the original average flow lengths for the 90 1-second intervals for COS is 3.2081, while the mean of the predicted average flow

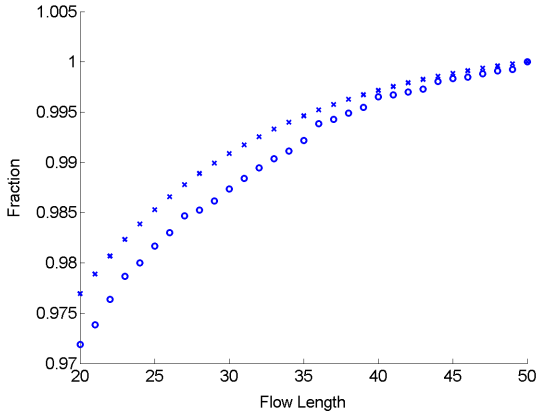

Fig. 6. Cumulative density function of predicted flow distribution.

lengths is 3.1178. The average relative error between the predicted and actual mean flow lengths over the 90 1-second intervals, is thus only 2.82%. Figure 7 presents the progression in the predicted mean flow length, and the actual (empirical) one, over 20 such 1-second intervals.

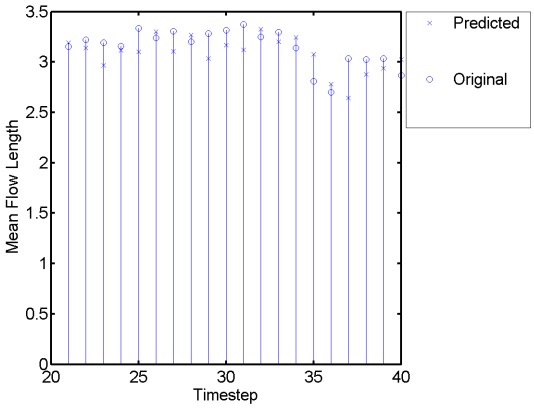

Fig. 7. Progression in the predicted and empirical mean Flow Lengths, using the EM method.

We used a tolerance of 10^{-5} as our convergence [24, 25] criteria for the parameters of the model. We observed that around 3 iterations are required to reach this tolerance level. However, the results were seen to be dependent on the initial conditions. We thus decided to run the algorithm M^2 times for each period with random initial conditions, and choose the final parameter values that gave the lowest L1 difference.

The numbers in Fig. 1 suggest that 16 bits $\times$ 50 = 800 bits can represent the flow distribution, using a cutoff of $FlowLength = 50$ and 16 bits to represent each y-axis value. This means that if a network contains 16 nodes and runs the EM algorithm at another central node over 1-second intervals of data, the network must transmit 12.8 kbits of data every second.

4. THE DISTRIBUTED EM METHOD

The EM method presented in Section 3 requires that the node running it, be in possession of the entire flow distribution information. When applied to a particular wide-area network, this means that each node must transmit all it's data to a designated central node, if predictions are desired for the overall network. Only when the central node receives the data from all the nodes in the network, can it run the EM algorithm to estimate the parameters of the model, and subsequently predict distributions into the future. Transmitting entire data distributions can be expensive if the network is large or spread over a wide geographical area.

Another approach would be to run a distributed version of the standard EM algorithm separately at each node, and transmit only updates regarding the parameter set, as opposed to raw data. The underlying assumption here is that communication costs exceed computation costs in today's networks. Thus it is better to perform most of the computation locally at each node, and only transmit information on the parameters of the model, as opposed to raw data.

To obtain a distributed version of the EM algorithm, let us assume that the network contains H nodes. Further, assume that the geometric probability parameters of our model (2) p_m for $m \in \{1 \dots M\}$ are universal to the network, while the mixing probabilities $\alpha_{h,m}$ for $m \in \{0 \dots M\}$ and $h \in \{1 \dots H\}$ are specific to each node. Assuming that we have N_h i.i.d. observations $X_h = \{x_{h,1} \cdots x_{h,N_h}\}$ of the data at node h, we get the following log-likelihood function for the total data in the network:

$$\log(\mathrm{L}(\Theta|X)) = \sum_{h=1}^{H} \sum_{i=1}^{N_h} \log(\sum_{m=1}^{M} \alpha_{h,m} \Pr(x_{h,i}|\Theta)) \quad (8)$$

where $\Pr(x_{h,i}|\Theta)$ is as defined in (3).

The resulting Q-function can be re-written in the following form [17]:

$$\begin{aligned} &\mathrm{Q}(\Theta^{t+1}, \Theta^t) \\ &= \sum_{h=1}^{H} \sum_{i=1}^{N_h} \sum_{m=1}^{M} w_{h,i,m}^{t+1} \{\log|\alpha_{h,m}| + \\ &\qquad \log|\Pr(x_{h,i}|\Theta_m^t)|\} \end{aligned} \quad (9)$$

where

$$w_{h,i,m}^{t+1} = \frac{\alpha_{h,m}^t \Pr(x_{h,i}|\Theta_m^t)}{\sum_{m=1}^{M} \alpha_{h,m}^t \Pr(x_{h,i}|\Theta_m^t)} \quad (10)$$

Further, define the following "summary" quantities

$$w_m^t = \sum_{h=1}^{H} \sum_{i=1}^{N_h} w_{h,i,m}^t \tag{11}$$

and

$$a_m^t = \sum_{h=1}^{H} \sum_{i=1}^{N_h} w_{h,i,m}^t x_{h,i,m}. \tag{12}$$

With the summary quantities, the M-Step equations for the parameters become:

$$\alpha_{h,m}^{t+1} = \frac{1}{N_h} \sum_{i=1}^{N_h} w_{h,m,i}^{t+1} \tag{13}$$

and

$$p_m^{t+1} = \frac{a_m^t}{a_m^t + w_m^t} \tag{14}$$

where the value of $w_{h,i,m}^{t+1}$ required in (13) is defined in terms of the current values of the parameters (Θ^t) by (10). In addition, we have

$$w_{h,m}^{t+1} = \sum_{i}^{N_h} w_{h,i,m}^{t+1}. \tag{15}$$

A distributed implementation of the algorithm may then be obtained as follows. Assume that all nodes have the current parameter estimates Θ^t. The updated estimates after the next iteration of the EM algorithm, Θ^{t+1}, can be computed by performing two message passing cycles through the nodes. Each message passing operation involves sending the sufficient statistic

$$s^t = \{w_m^t, a_m^t\} \tag{16}$$

for $m \in \{0 \cdots M-1\}$ to the next node. Note that the sufficient statistic does not include the probability parameter p_m that is universal to all the nodes in the network, or the mixing probabilities $\alpha_{h,m}$ which are unique to each node. Once a particular node (let us say node h) receives s^t from the previous node (node $h-1$), it will locally run the standard EM algorithm using the modified equations for the M-Step presented in (13) and (14), and the equation for $w_{h,i,m}$ given in (10). In addition, it will calculate the updated value of $w_{h,m}^{t+1}$ using (15). Note that node h uses it's locally available data $\{x_{h,1} \ldots x_{h,N_h}\}$, and the present values of it's own mixing probabilities $\{\alpha_1 \ldots \alpha_{h,m}\}$.

Finally, node h will update [17] the summary quantities according to:

$$w_m^{t+1} = w_m^t + w_{h,m}^{t+1} - w_{h,m}^t \tag{17}$$

and

$$a_m^{t+1} = a_m^t + a_{h,m}^{t+1} - a_{h,m}^t. \tag{18}$$

It will then form the updates sufficient statistic $s^{t+1} = \{w_m^{t+1}, a_m^{t+1}\}$, and transmit s^{t+1} to the next node (node $h+1$) where the process is repeated.

4.1. Results with the Distributed EM (DEM) Method

We simulated with the traces from COS, ODU and BWY, with 16 nodes. Recalling that our flow key = (src IP, src port, dst IP, dst port), we split the total data into 16 bins by source address. We could then assume that the 16 bins correspond to locally available data in a distributed network with 16 nodes. The total data is then what would be present at the central node, after each node sends it's individual readings.

The simulation proceeds as follows. Each node receives the sufficient statistic $s^t = \{w_m^t, a_m^t\}$ from the previous node, then locally maximizes the M-Step equations until a tolerance of 10^{-5} is reached for it's estimates of the parameters $p_m, m \in \{1 \ldots M-1\}$ and $\alpha_{h,m}, h \in \{1 \ldots H\}, m \in \{0 \ldots M-1\}$. It then forwards the updated sufficient statistic to the next node. In this way, updated statistics are passed from node 1 through node H (where $H = 16$ here), then again to node 1 for the next cycle. The cycles continue until the updates in the sufficient statistic itself converge to within a tolerance of 10^{-5}. We observed that about 3 local EM iterations were needed at each node for the parameter values to converge, before the sufficient statistic is forwarded to the next node. However, as many as 20 cycles through the 16 nodes were needed for the w_m^t and a_m^t in s^t to converge to the same 10^{-5} tolerance level. Another possible approach to the distributed EM algorithm is to perform a single iteration (instead of maximizing) locally at each node, before forwarding the updates. Both approaches were observed to require comparable number of cycles to converge, in this application. Recall that 3 iterations were also required for convergence of the centralized EM algorithm for each 1-second interval. A major difference between the centralized EM case in Section 3 and the distributed version here, is the following. Here we assume that while the total flow distribution at any time interval is explained by the same set of geometric probability parameters p_m, the set of mixing probabilities $\alpha_{h,m}$ is unique to each node. We again assumed random initial conditions.

Figure 8 overlays the flow distribution predicted for the next 1-second time interval based on the distribution from the current interval, and the actual distribution for the next interval. Figure 9 presents the predicted and actual flow distributions on a logarithmic scale. Although we used flow histograms with flow lengths of 1 through 50, Fig. 8 presents only flow lengths 1 to 10 for clarity of presentation. The packet trace is that of the same 1-second interval in COS

as used for Figures 2 and 3, and we used $M = 4$. Figure 10 demonstrates the behavior in the tail-region of the corresponding cumulative density function (cdf) of the predicted and actual flow distributions.

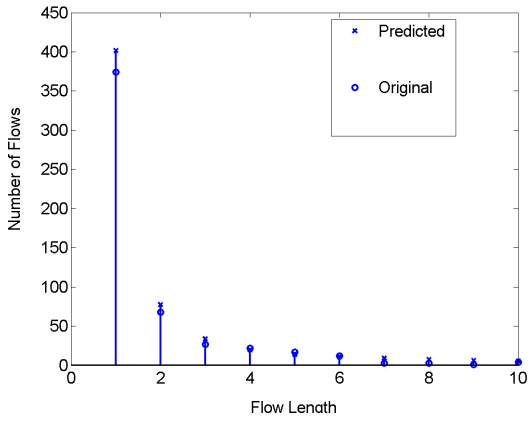

Fig. 8. Predicted flow distribution at a node using DEM, in linear scale.

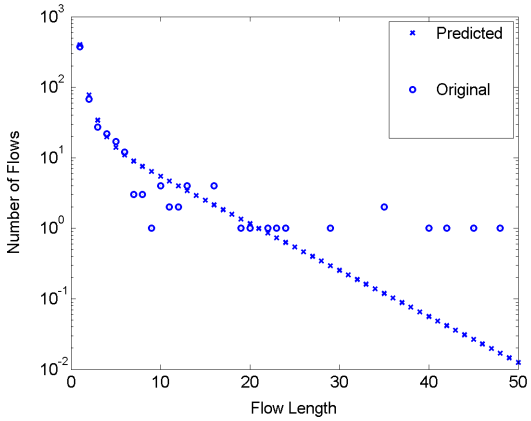

Fig. 9. Predicted flow distribution at a node using DEM, in logarithmic scale.

Figures 8 and 9 show the closeness of fit between the predicted distributions and the actual ones, while Fig. 6 shows that a high percentage of flows are accurately predicted. The prediction in the tail is observed to be worse than in the centralized EM case. This is because the total data, specially the large flows in the tail region, is not uniformly distributed across the 16 nodes.

We obtained estimates for the flow length distribution using different values of M. Table 2 below summarizes the mean L1 differences between the empirical distribution and the predicted one for the 16 nodes, for a particular 1-second interval. The packet traces correspond to COS, ODU and BWY.

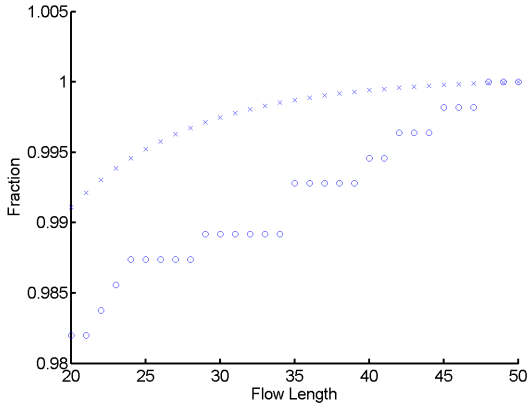

Fig. 10. Cumulative density function of predicted flow distribution.

M	3	4	5	6
COS	0.06	0.06	0.06	0.06
ODU	0.25	0.25	0.25	0.25
BWY	0.18	0.18	0.18	0.18

The most striking feature in the results is that the mean L1 differences here do not vary much with M. Also, the values of L1 differences are higher than in the centralized case. Once again, this may be attributed to the fact that the total data is not uniformly distributed across the 16 nodes.

In the distributed implementation each node runs the EM algorithm on locally available data, and only transmits the w_m and a_m components. Our results suggest that 32 bits are sufficient to represent each w_m or a_m component of s, thereby requiring a total of $64M$ bits. For our suggested value of $M = 4$, this evaluates to 256 bits. With a ring topology having H nodes and once again assuming 1-second intervals of data, this means that only 256 bits of data are flowing in the network every H-second period. It was mentioned earlier that 20 cycles are required for convergence with 16 nodes. With $H = 16$ nodes, this indicates an average throughput required of $20 \times 256 \div 16 = 320$ bits/s. Compared to 12,800 bits/s estimated for the centralized version, the distributed approach indicates a lower cost in terms of the throughput needed. We have implicitly assumed that the computation time at each node is insignificant, compared to the 1-second data collection period.

5. THE PARTICLE FILTER METHOD

5.1. Overview of the Particle Filter

Bayesian methods provide a framework for taking real-world noisy data and estimating some phenomenon based on observations. The objective is to create a model which depicts the typical behavior of the quantities being investigated. Bayesian methods enable us to relate prior distributions of the unknown data - that is previous knowledge of the phenomenon - with the likelihood function associated with the observations. Some prior knowledge regarding the phenomenon being modelled is available in many applications, thereby allowing the formulation of Bayesian models.

A method of analytically estimating the unknown is by using the popular Kalman Filter [26]. The Kalman filter allows for an exact expression of the sequence of posterior distribution to be computed. However, a serious drawback of the Kalman Filter method is that the data has to be modelled as a linear Gaussian state-space. Unfortunately, most phenomena in the real world are non-linear and non-Gaussian. The Extended Kalman Filter (EKF) [27] approximates by linearizing the predicted states.

Sequential Monte Carlo (SMC) methods, such as the Particle Filter, were devised to counter the aforementioned problem. This is a numerical method based on simulation, and is convenient to use in the modern day as computation power is easily available. Algorithms related to SMC methods come under many names. We apply one of the SMC methods, the Particle Filter, to this problem.

In the ideal situation, one is able to represent the posterior distribution using a set of samples or particles. This is achieved by taking N independent and identically distributed (i.i.d.) random samples $\mathbf{x}_{0:t}^{(i)}$. These samples (or particles) are drawn from the probability density function $p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$. As a result, we can obtain an empirical estimate of this distribution by

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{0:t}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x}_{0:t} - \mathbf{x}_{0:t}^{(i)}). \tag{19}$$

Drawing samples directly from the posterior distribution is, however, often difficult to do. In such a situation, a technique called Importance Sampling is used, whereby samples can be drawn from a known *proposal distribution*: $\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$.

We then can define the normalized importance *weight* as

$$w(\mathbf{x}_{0:t}) \propto \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} \tag{20}$$

If we then sample $\mathbf{x}^{(i)}$ from $\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ and use the appropriate weights, the density can be rewritten as

$$\widehat{p}(\mathbf{x}_{0:t}|\mathbf{y}_{0:t}) = \sum_{i=1}^{N} w(\mathbf{x}_{0:t}^{(i)})\delta(\mathbf{x}_{0:t} - \mathbf{x}_{0:t}^{(i)}). \tag{21}$$

The technique above works very well, except for one issue: the degeneracy problem. The variance of importance weights increase stochastically over time. To regulate this problem, samples with very low importance weights are eliminated and replaced with ones of higher ratio. This step is known as resampling and essentially means that particles with low weights are virtually eliminated and replaced with particles of higher weights, which best represent the posterior distribution.

A complete description of the particle filter is provided in [28].

5.2. Motivation behind Using the Particle Filter

We applied the EM method with $M = 4$, to 90 consecutive 1-second intervals in COS, ODU and BWY to estimate the parameters for each 1-second interval, and study the variation of the parameters over timesteps. Figure 11 shows how the relative differences in α_0 changes over time for COS. The behaviors of α_1, α_2 and α_3 are similar. Relative difference for α_m is defined as:

$$\mathrm{RD}(\alpha_m) = [\alpha_m(t) - \alpha_m(t-1)]/\alpha_m(t-1) \tag{22}$$

The relative difference for p_m is defined analogously.

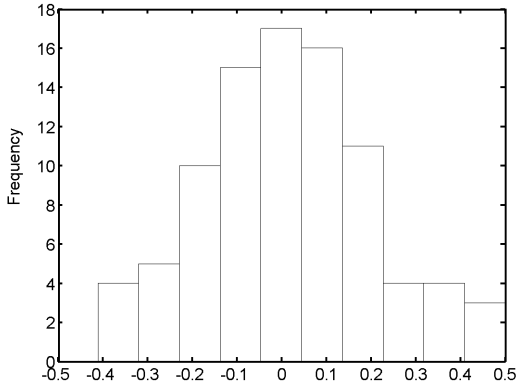

Fig. 11. Relative difference in α_0.

Figure 12 shows how the relative difference in p_1 changes over time. The behaviors of p_2 and p_3 are again similar.

Figure 13 shows how the relative difference in n, the total number of flows, changes over time.

Figures 11-13 show that the means of the relative differences are very small. Table 3 shows the mean relative

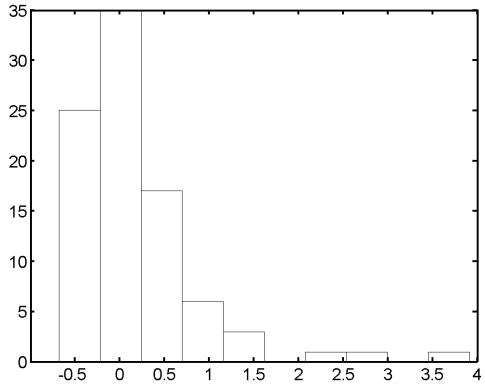

Fig. 12. Relative difference in p_1.

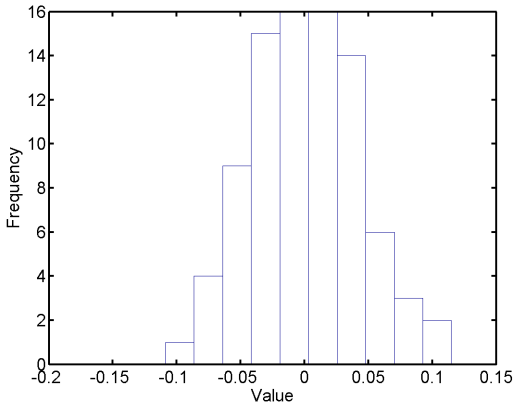

Fig. 13. Relative difference in n.

difference (relative difference, RD, was defined in (22)) and the mean relative difference as a fraction of the mean value of the parameter, for all the parameters with $M = 4$. The source is the same set of 90 consecutive 1-second intervals in COS as before.

	$\text{mean}[\text{RD}(\alpha_m)]$	$\text{mean}[\text{RD}(\alpha_m)]/\text{mean}[\alpha_m]$
$m = 0$	0.0246	0.0811
$m = 1$	0.0436	0.1531
$m = 2$	0.1844	0.8056
$m = 3$	0.2828	1.5470
	$\text{mean}[\text{RD}(p_m)]$	$\text{mean}[\text{RD}(p_m)]/\text{mean}[p_m]$
$m = 1$	0.1651	0.4982
$m = 2$	0.0492	0.0835
$m = 3$	0.0001	0.0002
	$\text{mean}[\text{RD}(n)]$	$\text{mean}[\text{RD}(n)]/\text{mean}[n]$
	7.35×10^{-4}	1.08×10^{-8}

It is apparent from Table 3 that the relative changes in the parameters over consecutive 1-second intervals, are small. This observation suggests the use of a particle filter with step size corresponding to 1-second, as a method of predicting the parameters for the next interval.

Figures 11-13 also suggest that changes in the relative differences of the parameters exhibit Gaussian behavior. In order to test the goodness of fit of a Gaussian, we performed the Shapiro-Wilk [29,30] Normality Test on the relative differences for the parameters. The resulting Shapiro-Wilk W-statistics and associated significance levels, are given in Table 4 below:

parameter	W-statitic	significance level
$\text{RD}(\alpha_0)$	1.05	0.15
$\text{RD}(\alpha_1)$	3.08	1.02×10^{-3}
$\text{RD}(\alpha_2)$	6.53	3.27×10^{-11}
$\text{RD}(\alpha 3)$	8.23	1.11×10^{-16}
$\text{RD}(p_1)$	-0.65	0.26
$\text{RD}(p_2)$	4.70	1.29×10^{-6}
$\text{RD}(p_3)$	5.43	2.79×10^{-8}
$\text{RD}(n)$	-1.5406	0.0617

The results of the Shapiro-Wilk test suggest that, one can think of the relative differences in the parameters as realizations from Normal random vectors at time t, at the 0.16 significance level (with the exception of p_1)). In addition, we know that the α_m mixing parameters must sum to 1, while the p_m parameters are independent. This led us to choose the Dirichlet distribution as the prior distribution for the α_m mixing parameters, and the Normal distribution as the prior distribution for the probability parameters p_m and the number of flows n.

5.3. Development of the Particle Filter

The state at timestep *t* for the Particle Filter has been defined as:

$$s(t) = [a(t), p(t), n(t)]^{\text{T}} \tag{23}$$

where $n(t)$ = total number of flows at timestep *t*, while $\alpha(t) = \alpha_0(t), ..., \alpha_{M-1}(t)$ and $p(t) = p_1(t), ..., p_{M-1}(t)$ denote the parameters required to explain the flow distribution as a mixture of geometric distributions at timestep *t*, in accordance with (2)

For the prior distribution, we assume that $\alpha_m(t)$ is subject to a Dirichlet probability density function with parameter $\alpha_m(t-1)$ [31, 32], while $p_m(t)$ and $n(t)$ are subject to Normal distributions with means $p_m(t-1)$ and $n(t-1)$ (the standard deviations are set from previous data). We choose the Dirichlet distribution as the prior distribution for $\alpha_m(t)$ as this distribution is very flexible and general, and capable of representing a wide variety of random processes

[10]. We choose Normal distributions as priors for $p_m(t)$ and $n(t)$ in accordance with the results presented in Table 4, for the goodness of fit of a Gaussian for the relative differences of these parameters. Particles are thus formed by sampling from Dirichlet, Normal and Normal distributions to obtain the α, p and n values respectively, at each timestep.

The estimated flow distribution is obtained at timestep t by using the following equation:

$$z(t) = \alpha_0(t) \cdot \delta(x-1) + \sum_{m=1}^{M-1} \alpha_m(t) \cdot (1 - p_m(t)) \cdot p_m(t)^x \tag{24}$$

Given the actual flow distribution $y(t)$, the Likelihood probability is a Multinomial [33] with the individual component probabilities given by $z(t)$ and the number of occurrences of each type given by $y(t)$:

$$\mathrm{L}(t) = \mathrm{Multinomial}(y(t)|z(t), n(t))$$

Each particular Flow Length (value of x) is thought of as a type or bin.

The multinomial probability of randomly distributing k_1, k_2, ..., k_m out of a total of $K = k_1 + k_2 + \ldots + k_m$ objects into each of m bins respectively, is given by:

$$M = \frac{K!}{k_1! k_2! \ldots k_m!} \cdot p_1^{k_1} \cdot p_2^{k_2} \cdot \ldots \cdot p_m^{k_m} \tag{25}$$

The Importance Sampling weights are then given by:

$$\begin{aligned} & w_i(t) \\ &= w_i(t-1) \times \mathrm{L}(t) \\ &= w_i(t-1) \times \mathrm{Multinomial}(y(t)|z(t), n(t)) \end{aligned} \tag{26}$$

for each of the $i \in \{1 \ldots P\}$ particles.

5.4. Results with the Particle Filter

We used the Particle Filter with $P = 600$ particles to predict the flow length distributions from a large number of packet traces.

Figures 14 and 15 overlay the flow distribution predicted for the next 1-second time interval based on the distribution from the current interval, and the actual distribution for the next interval. Figure 14 is on a linear scale while Fig. 15 is on a logarithmic scale. Although we used flow histograms with Flow Lengths of 1 through 50, Fig. 14 presents only Flow Lengths 1 to 10 for clarity of presentation. We used $M = 4$, and the packet trace is that of the same 1-second interval in COS as used for Figures 2 and 3. Figure 16 demonstrates the behavior in the tail-region of the corresponding cumulative density function (cdf) of the predicted and actual flow distributions. Figures 14 and 15 show the closeness of

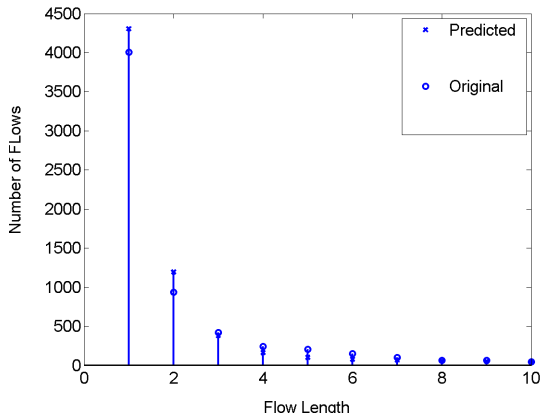

Fig. 14. Predicted flow distribution using the particle filter, in linear scale.

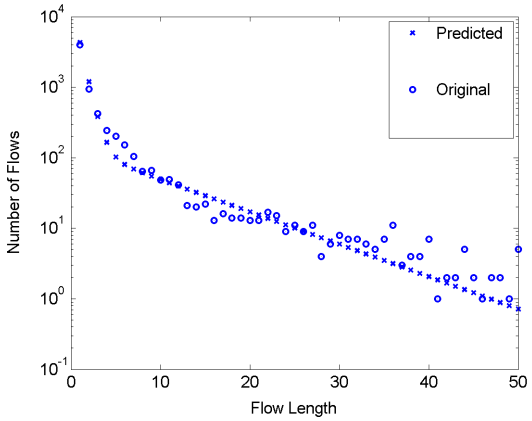

Fig. 15. Predicted flow distribution using the particle filter, in logarithmic scale.

fit between the predicted distributions and the actual ones, while Fig. 16 shows that a high percentage of flows are accurately predicted.

Table 5 below summarizes the mean L1 differences (over the 90 consecutive 1-second intervals) between the predicted distribution and the empirical one, using different values of M. The packet traces correspond to COS, ODU and BWY, and we used $P = 600$ particles.

M	3	4	5	6
COS	0.14	0.13	0.13	0.13
ODU	0.14	0.13	0.13	0.13
BWY	0.10	0.10	0.10	0.10

The particle filter is a stochastic method, and different runs produce different results. One must therefore decide on how many particles to use in a particular application. To study

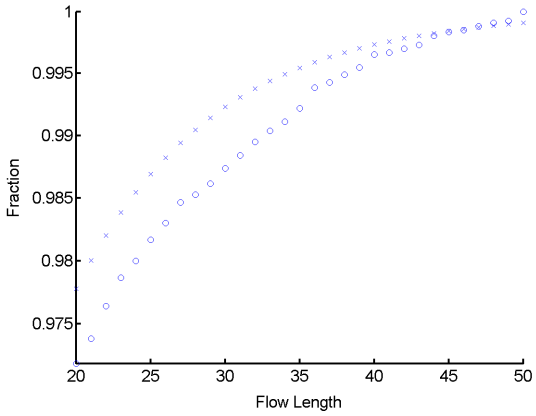

Fig. 16. Cumulative density function of predicted flow distribution using the particle filter, in logarithmic scale.

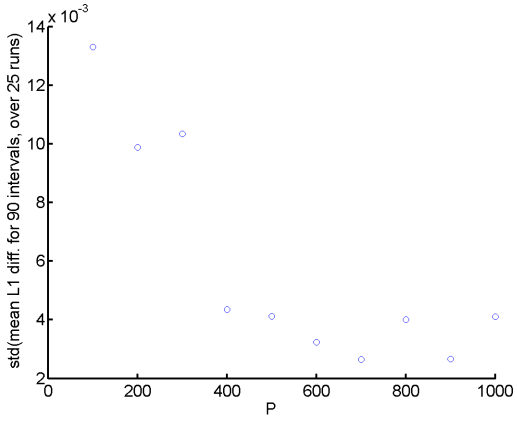

Fig. 18. Precision of results with different number of particles.

the accuracy and precision (stability) of results using different number of particles, we ran the particle filter 25 times on the COS trace (with $M = 4$) for various values of P. Figure 17 shows how the *average* over 25 runs of the mean L1 difference (over the 90 intervals), varies with P. Figure 18 shows how the *standard deviation* over 25 runs, of the mean L1 difference varies with P. Recall that Table 5 presented the mean L1 difference over the 90 intervals, obtained over 1 run with $P = 600$.

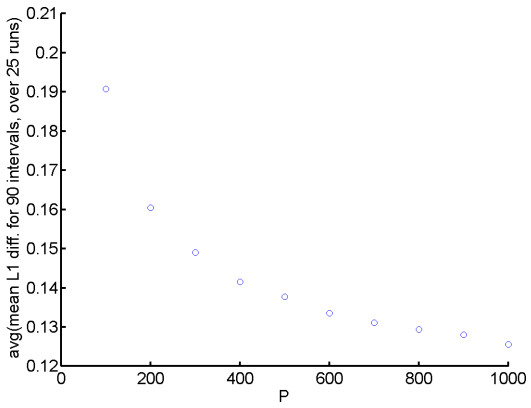

Fig. 17. Accuracy of results with different number of particles.

It is observed from Figures 17 and 18 that using a value of P greater than 600 does not significantly improve the prediction (i.e. decrease the average L1 difference) or reduce the variation in the results (i.e. decrease the standard in the L1 differences). As increasing the number of particles increases the runtime, we decided to use $P = 600$ particles.

The mean of the original average flow lengths for the 90-second duration for COS is 3.2081, as previously stated in Section 3. The mean of the predicted average flow lengths with the particle filter is 3.2172. The histogram in Figure 19 shows the corresponding difference between the predicted and original average flow lengths, for each 1-second interval in COS.

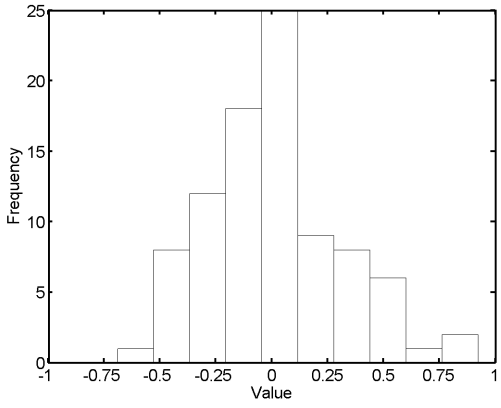

Fig. 19. Error in prediction of average flow length by the Particle Filter.

In this case, we see an average overestimate of 0.0091, with a standard deviation of 0.3071. This overestimate is only 0.28% of the original mean flow length.

Figure 20 presents the progression in the predicted mean flow length, and the actual (empirical) one, over 20 intervals.

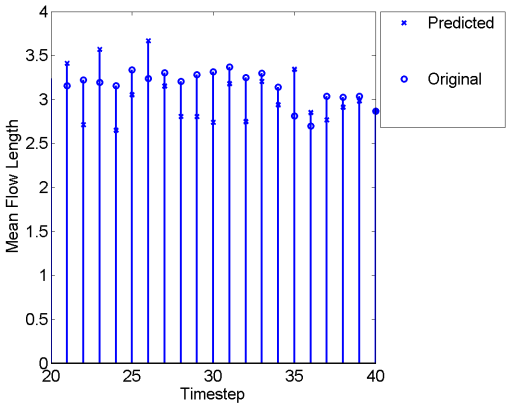

Fig. 20. Progression in the predicted and empirical mean Flow Lengths, using the particle filter.

6. CONCLUSIONS AND FURTHER WORK

Nodes such as core routers in wide-area networks need to predict flow lengths and flow distributions ahead of time for resource reservation, as well as for a number of other purposes. We have postulated that the flow histogram observed at a node is a realization of a random vector. We have proposed a mixture of geometric probability mass functions as a model to describe the flow distribution. We have then presented three methods (a centralized EM, a distributed EM and a Particle Filter) to estimate the parameters of the model, and thereby predict the entire flow distribution at a node. We have tested our methods on a large number of packet traces from the NLANR PMA repository. The results indicate that all three methods perform equally well, as demonstrated by similar L1 differences between the empirical flow distribution and the predicted one.

We observed that the standard centralized version of the EM algorithm required about 3 iterations to converge, while the distributed version required about 3 iterations at each node and 20 cycles with 16 nodes. In terms of runtime, these two algorithms are significantly faster than the Particle Filter. The reason is that about 600 particles are required with the Particle Filter. The number of particles is thus much greater than the number of iterations involved with the two EM methods. Moreover, the different stages of the Particle Filter method (sampling from initial distributions, propagation, resampling) make this method relatively slower.

The distributed version forwards only the parameter updates and requires far less transmission capacity, compared to the centralized version that sends raw data, when overall statistics for a wide-area network such as the AAPN are desired.

Further research possibilities regarding this topic include designing a distributed version of the Particle Filter, experimentally testing the performance of the distributed EM algorithm on a real wide-area network, and relaxing the restriction in the distributed model that the geometric (probability) parameter set is universal to the network.

7. REFERENCES

[1] N. Duffield, C. Lund, and M. Thorup, "Estimating flow distributions from sampled flow statistics," in *ACM SIGCOMM, Karlsruhe, Germany*, August 2003.

[2] A. Feldmann, R. Caceres, F. Douglis, G. Glass, and M. Rabinovich, "Performance of web proxy caching in heterogeneous bandwidth environments," in *Proc. IEEE INFOCOMM, New York, NY*, March 1999, pp. 107–116.

[3] A. Feldmann, J. Rexford, and R. Caceres, "Efficient policies for carrying web traffic over flow-switched networks," *IEEE/ACM Trans. on Networking*, vol. 6, no. 6, pp. 673–685, December 1998.

[4] A. Shaikh, J. Rexford, and K. Shin, "Load-sensitive routing of long-lived ip flows," in *Proc. of SIGCOMM, Cambridge, MA*, September 1999, pp. 215–226.

[5] H. Abrahamsson, B. Ahlgren, J. Alonso, A. Andersson, and P. Kreuger, "A multi path routing algorithm for ip networks based on flow optimisation," in *Int'l Workshop on Quality of future Internet Services (QofIS), Zurich, Switzerland*, October 2002.

[6] A. Sridharan, R. Guerin, and C. Diot, "Achieving near-optimal traffic engineering solutions for current ospf/is-is networks," in *Proc. of IEEE INFOCOM, San Francisco, CA*, March 2003, pp. 1167–1177.

[7] X. Su and G. de Veciana, "Dynamic multi-path routing: asymptotic approximation and simulations," in *ACM SIGMETRICS Performance Evaluation Review*, June 2001, vol. 29, pp. 25–36.

[8] IETF Working Group, "Packet sampling charter," IETF charter, http://www.ietf.org/html.charters/psamp-charter.html.

[9] National Laboratory for Applied Network Research, "Passive measurement and analysis project," repository of data traces, http://pma.nlanr.net/.

[10] A. Soule, K. Salamatian, N. Taft, R. Emilion, and K. Papagiannaki, "Flow classification by histograms: or how to go on safari in the internet," in *Proc. of Joint International Conference on Measurement and Modeling of Computer Systems, New York, NY*, March 2004, pp. 49–60.

[11] K. Papagiannaki, N. Taft, and C. Diot, "Impact of flow dynamics on traffic engineering design principles," in *IEEE INFOCOM, Hong Kong*, March 2004.

[12] C. Estan and G. Varghese, "New directions in traffic measurement and accounting," in *Proc. of 1st ACM SIGCOMM Workshop on Internet Measurement, San Francisco, CA*, November 2001, pp. 75–80.

[13] D. Papagiannaki, N. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian, and C. Diot, "A pragmatic definition of elephants in internet backbone traffic," in *Proc. of 2nd ACM SIGCOMM Workshop on Internet Measurment, Marseilles, France*, November 2004, pp. 175–176.

[14] N. Duffield, C. Lund, and M. Thorup, "Properties and prediction of flow statistics from sampled packet streams," in *Proc. of 2nd ACM SIGCOMM Workshop on Internet Measurment, Marseille, France*, November 2002, pp. 159–171.

[15] K. Claffy, G. Polyzos, and H. Braun, "Application of sampling methodologies to network traffic characterization," in *Proc. ACM SIGMCOMM, San Francisco, CA*, September 1993, pp. 194–203.

[16] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *ACM SIGMETRICS, New York, New York*, June 2004.

[17] R. Nowak, "Distributed em algorithms for density estimation and clustering in sensor networks," *IEEE Trans. of Signal Processing*, vol. 51, no. 8, pp. 2245–2253, August 2003.

[18] National Institute of Science and Technology (NIST), "Engineering statistics handbook," Section 8.4.1.2: Maximum likelihood estimation, http://www.itl.nist.gov/div898/handbook/apr/section4/apr412.htm.

[19] A. Papoulis and S. Pillai, "Probability, random varibles and stotastic processes, fourth edition," 2002.

[20] J. Bilmes, "A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," Technical report tr-97-021, International Computer Science Institute, Berkeley, CA, 1997.

[21] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions, 1st Edition*, Wiley-Interscience, New York, 1996.

[22] M. Desco, J. Gispert, S. Reig, A. Santos, J. Pascau, N. Malpica, and P. Garcia-Barreno, "Statistical segmentation of multidimensional brain datasets," in *Proc. SPIE*, July 2001, vol. 4322, pp. 184–193.

[23] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Royal Stat. Society, Series B.*, vol. 39, pp. 1–38, 1997.

[24] C. F. J. Wu, "On the convergence properties of the em algorithm," *Annals of Statistics*, vol. 11, no. 1, pp. 95–103, 1983.

[25] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for gaussian mixtures," *Neural Computation*, vol. 8, no. 1, pp. 129–151, 1996.

[26] G. Welch and G. Bishop, "An introduction to the kalman filter," in *SIGGRAPH 2001*, 2001.

[27] A. H. Jazwinski, *Stochastic processes and filtering theory*, Academic Press, New York, 1970.

[28] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods*, Springer-Verlag, New York, 2001.

[29] S. Shapiro and M. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, pp. 591–611, 1965.

[30] National Institute of Science and Technology (NIST), "Engineering statistics handbook," 7.2.1.3.: Anderson-Darling and Shapiro-Wilk tests, http://www.itl.nist.gov/div898/handbook/prc/section2/prc213.htm.

[31] A. Ranganathan, "The dirichlet process mixture model," unpublished document, 1997, http://www.cc.gatech.edu/people/home/ananth/dirichlet.pdf.

[32] A. Honkela, "Nonlinear switching state-space models," Master's thesis, Helsinki University of Technology, Helsinki, Finland, 2001, Dirichlet distribution, http://www.cis.hut.fi/ahonkela/dippa/node95.html.

[33] A. Honkela, "Nonlinear switching state-space models," Master's thesis, Helsinki University of Technology, Helsinki, Finland, 2001, Multinomial distribution, http://rkb.home.cern.ch/rkb/AN16pp/node179.html.