

Real-Time Multi-path Tracking of Probabilistic Available Bandwidth

Frederic Thouin, Mark Coates, Michael Rabbat

Department of Electrical and Computer Engineering

McGill University, Montreal, Canada

frederic.thouin@mail.mcgill.ca, mark.coates@mcgill.ca, michael.rabbat@mcgill.ca

Abstract—Applications such as traffic engineering and network provisioning can greatly benefit from knowing, in real time, what is the largest input rate at which it is possible to transmit on a given path without causing congestion. We consider a probabilistic formulation for available bandwidth where the user specifies the probability of achieving an output rate almost as large as the input rate. We are interested in estimating and tracking the network-wide probabilistic available bandwidth (PAB) on multiple paths simultaneously with minimal overhead on the network. We propose a novel framework based on chirps, Bayesian inference, belief propagation and active sampling to estimate the PAB. We also consider the time evolution of the PAB by forming a dynamic model and designing a tracking algorithm based on particle filters. We implement our method in a lightweight and practical tool that has been deployed on the PlanetLab network to do online experiments. We show through these experiments and simulations that our approach outperforms block-based algorithms in terms of input rate cost and probability of successful transmission.

I. INTRODUCTION

The latest research in overlay network routing [1], [2] and anomaly detection [3] has shown that knowing the amount of available bandwidth (AB) of paths across the network can lead to better performance. This knowledge could be helpful to many other applications, such as SLA compliance, network management, transport protocols, traffic engineering or admission control, but current available bandwidth estimation techniques and tools generally do not meet application-specific requirements in terms of accuracy, overhead, latency and reliability [4]. Furthermore, the commonly-used definition of available bandwidth, in terms of utilization, and several tools and techniques suffer from three major deficiencies: 1) the models that associate the AB metric to the observed data are inaccurate in many practical scenarios; 2) the majority of existing tools produce a point estimate of average AB and do not provide a confidence interval; 3) most current tools are not well-adapted to multi-path or real-time estimation for applications such as traffic engineering and network provisioning [5].

In this paper, we focus on the problem of tracking available bandwidth in real-time for multiple paths simultaneously. Existing available bandwidth tracking tools, most of which are based on the Kalman filter [6]–[10], do not address the simultaneous tracking of the available bandwidths of multiple intersecting paths. The available bandwidth estimation techniques which do address multiple paths [11]–[13] construct a

single estimate for each path based on a block of data; they do not perform tracking.

In [5], we proposed a tool for block-based estimation of a metric (that we assumed to be static) called probabilistic available bandwidth (PAB). The PAB metric is defined to address the weaknesses of the utilization-based definition and related estimation tools and techniques¹. In this paper we allow PAB to vary with time and formulate inference as a filtering task. Our main contribution is to propose a network-wide multi-path available bandwidth tracking procedure. We demonstrate a lightweight, practical implementation of the proposed approach that dramatically reduces the measurement overhead for PAB estimation and renders it feasible for multiple paths. We extend the Bayesian formulation of our estimation problem and use Dynamic Bayesian Networks (DBNs) [14], [15] to track the PAB. We use particle filtering (a method that combines sequential importance sampling [16], [17] and resampling) to approximate the PAB of the links in the network with a mixture of weighted Gaussians. Our Bayesian approach allows us to use the marginal posteriors of the paths directly to produce confidence intervals.

The rest of the paper is organized as follows. In Sect. II, we summarize work related to available bandwidth tracking and network-wide estimation. In Sect. III, we provide the exact definition of probabilistic available bandwidth and formulate the tracking problem. In Sect. IV, we present our measurement model constructed using chirps that provide considerable savings in terms of number of probes required compared to constant-rate packet trains. In Sect. V, we outline the tracking methodology based on factor graphs, belief propagation and particle filtering. In Sect. VI, we show the results of our simulation and online experiments on the PlanetLab network. In Sect. VII, we summarize our work and discuss future research possibilities.

II. RELATED WORK

For real-time estimation, the proposed techniques [6]–[10] use Kalman filtering by taking advantage of the piecewise linear relation between utilization and available bandwidth. The main drawback of the Kalman filter is that conditional probability distributions have to be Gaussian-linear. Goldoni et al. use Vertical Horizontal Filtering that ignores sharp and

¹A precise definition of PAB is given in Sect. III.

non-persistent changes but quickly converges to the new value if they do persist [18]. However, since their tool was only tested in an environment with constant bit-rate cross-traffic, its performance for tracking is unknown.

These tools cannot be applied directly to scenarios where the available bandwidths of multiple paths have to be simultaneously estimated. The probes will generate interference on links shared by multiple paths, which can lead to significant underestimation, and also introduce an unacceptable overhead and overload on the network and the hosts [19]. Alternatively, each path can be probed independently in a sequence rather than simultaneously. This approach is not only time-consuming but also very inefficient since it does not take advantage of the notable correlations in the AB when links are shared among paths. The techniques that have been proposed for large scale scenarios do rely on the correlations between links or even between the various metric (route, number of hops, capacity) to reduce the number of probes required to produce accurate estimates [11]–[13]. However, all of them are limited to estimating and not tracking the available bandwidth. Multipath tracking has been proposed for other metrics; Coates and Nowak use sequential Monte Carlo inference to estimate and track internal delay characteristics [20].

III. PROBLEM STATEMENT

The probabilistic available bandwidth (PAB) of a path p is defined in [5] as the maximum input rate r_p at which we can send traffic such that the output rate r'_p is almost as large as the input rate with specified probability². The PAB of each path p is modelled as a discrete³ random variable y_p . The PABs of the constituent links are denoted by x_ℓ . The probability that the PAB of path p is r is then $\Pr(y_p = r)$. For a $\epsilon > 0$ and $\gamma > 0$, we have:

$$y_p(t, \lambda) = \max\{r_p : \Pr(r'_p > r_p - \epsilon) \geq \gamma\} \quad (1)$$

where $y_p(t, \lambda)$ denotes the PAB at time t , for a measurement period $(t - \lambda, t]$, where λ is the number of observations made in one measurement period⁴.

The problem of interest in this paper is that of tracking PAB over time: for a user-specified (ϵ, γ) and known topology that consists of a set of N links, denoted \mathcal{L} , and M paths, denoted \mathcal{P} , we want to produce, at regular time intervals, estimates of the probabilistic available bandwidths for all paths in the network. Our measurement strategy is such that, within each measurement interval $(t - \lambda, t]$, we make λ measurements; each one on a single path. Each measurement may probe one or several rates. For each rate, we evaluate a binary outcome z_t that indicates whether or not the output rate is within ϵ Mbps of the probing rate. We use a binary outcome, rather

than the output rate, because it is less sensitive to noisy measurements and it is also easier to construct an accurate likelihood function empirically. We want to identify the most informative measurement at each iteration t such that, for successive windows of λ measurements, we can compute $\Pr(y_p(t, \lambda) | z_{t-\lambda:t})$ for every path. This marginal posterior can be used to produce credible intervals for the estimates, which can be employed to construct predictions of available bandwidth for subsequent time windows.

A. Assumptions

Our measurement and tracking methodology is based on three assumptions.

- 1) Each link is modelled as a store-and-forward first-come first-serve router/switch. If the network employs other forms of queueing or router-level Quality-of-Service provisioning, then we infer the PAB as seen by the class of packets transmitted as probes.
- 2) The routing topology of this network is known and it remains fixed during the tracking⁵. The IP addresses of the routers and the mapping to a physical topology is done using `traceroute`, which provides sufficiently accurate topology estimates for us to assess the performance of our algorithm. If there is per-packet load balancing, only one of the paths will be identified. Destination-based load balancing does not affect our method.
- 3) There is a single tight link on each path that determines the PAB of that path. The PAB of a path is the minimum of the PABs of its constituent links. The presence of more than one tight link results in noise propagated in the factor graph during the estimation procedure.

IV. MEASUREMENT MODEL

One measurement methodology to estimate PAB consists of sending constant-rate trains of packets at probing rate r_p and measuring the receiving rate r'_p [5], [21]–[23]. The result of each train is a binary decision as to whether the probing rate was greater or smaller than the receiving rate. Estimating the receiving rate with sufficient accuracy requires a minimal number of packets. This means that for each rate probed, there is a significant overhead (load on the network) involved. To reduce this overhead, an alternative solution to constant-rate packet trains is chirps [18], [24], [25]. A chirp train consists of sending a train of packets with exponentially increasing packet spacing. By varying the spacing between the packets rather than keeping it constant, it is possible to probe multiple rates with a single chirp.

A chirp is defined by its length K , window size K_{min} , spacing factor θ , minimum spacing T and packet size P (Fig. 1). The length K represents the total number of packets in the chirp. Increasing the value of K allows for a larger number of rates to be probed with a single chirp, but it also

²The probability is defined over all possible multi-packet flows of average rate equal to the input rate that can be transmitted during the measurement period.

³We chose discrete, rather than continuous, random variables because it is not meaningful to have an infinite precision on the transmission rates.

⁴The average duration of one measurement using our tool is under 2 seconds.

⁵Simulations in [5] have shown that effect of changes in the topology in terms of accuracy and convergence time is negligible.

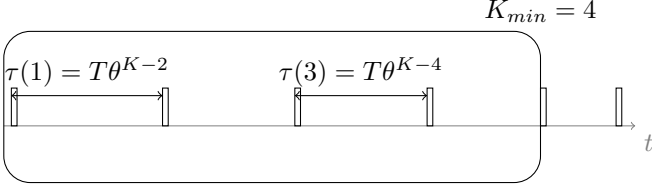


Fig. 1. Chirp of $K = 6$ packets. Probing rate with sliding window of $K_{min} = 4$ packets of $P = 1\text{KB}$ each. This chirp probes $K - K_{min} = 4$ rates.

increases the number of bytes injected into the network. We do not want our chirp to be too intrusive, but we are interested in having the ability to cover a wide enough range of rates with a single chirp (potentially the entire range of possible values).

To probe multiple rates with a single chirp, we apply a sliding window of size K_{min} across the chirp. The size of the sliding window represents a trade-off between accuracy and range of rates. The maximum number of rates probed by one chirp is $K' = K - K_{min}$, which means that reducing the value of K_{min} will allow for a larger range of rates to be probed. However, it will also result in noisier measurements. In pathChirp [25], a single packet pair is used to estimate one probing rate ($K_{min} = 1$). In the constant-rate approach, all the packets in the train are used ($K_{min} = K - 1$). We interpret each window of packets as a single probe with an input rate equal to the average rate over that window. The average input rate of the k th window in the chirp is calculated as follows:

$$r_p(k) = \frac{K_{min} \cdot P}{\sum_{i=k}^{k+K_{min}-1} \tau(i)} \quad (2)$$

where $\tau(i)$ is the spacing between the i th and $i + 1$ th packet, i.e., the time elapsed between the departure of the two packets.

The spacing factor θ and the minimum spacing T determine the packet spacing:

$$\tau(i) = T\theta^{K-(i+1)} \quad \forall i = 1..K - 1 \quad (3)$$

For a fixed packet size P , these values are adjusted to obtain the desired range of probing rates $[r_p(1), r_p(K')]$ covered by the chirp. In the constant-rate method, the spacing is constant throughout the entire train of packets, which results in a single probing rate. The main feature of chirps is that the spacing varies in order to test multiple probing rates with a single train. We fix the value of T and θ , by satisfying the following two equalities:

$$\frac{K_{min} \cdot P}{\sum_{i=1}^{K_{min}} \tau(i)} = r_p(1) \quad \frac{K_{min} \cdot P}{\sum_{i=K-K_{min}}^{K-1} \tau(i)} = r_p(K') \quad (4)$$

where $r_p(1)$ is the smallest rate we wish to probe and $r_p(K')$ is the largest. We note that a chirp does not necessarily probe K' different rates. As the range of rates gets smaller, it is possible that more than one window of packets will probe the same rate. Although we could reduce the chirp size to avoid this situation, we keep K constant and use the extra measurements to reduce the amount of noise and make the outcome more reliable.

Each chirp provides up to $K' = K - K_{min}$ measurements if no packets are dropped or discarded⁶. Each window of K_{min} packets is interpreted as a single probe rate at the average input rate over that window. We then measure the output rate r'_p for the same set of packets:

$$r'_p(k) = \frac{K_{min} \cdot P}{\sum_{i=k}^{k+K_{min}-1} \tau'(i)} \quad (5)$$

where $\tau'(i)$ is the delay between the arrival of the i th and $i + 1$ th packet at the receiver. Using the input and output rate vectors, we can compute a vector of binary outcomes:

$$z(k) = \mathbf{1}(r'_p \geq r_p - \epsilon) \quad \forall k = 1..K' \quad (6)$$

where $\mathbf{1}(\cdot)$ is the indicator function.

According to our simulations and online experiments, the chirp-based approach provides significant savings in terms of time required to form reliable estimates (at least 70%) and number of measurements required for accurate block-based estimation (at least 88%). Whereas the number of measurements increases drastically when using constant-rate packet trains for values of γ close to one, there is almost no impact when using chirps.

A. Likelihood function

To use the outcome of a chirp measurement in our Bayesian framework, we need to construct a likelihood function $L(\mathbf{z}|y_p)$ where $\mathbf{z} = [z(1) \dots z(K')]$ and y_p is the PAB of the probed path. To simplify the learning of such a function, we first assume that the outcomes from the different probed rates within a single chirp are independent:

$$L(\mathbf{z}|y_p) = \prod_{i=1}^{K'} L(z(i)|y_p) \quad (7)$$

As an example, we learn $L(z|y_p)$ empirically from 71050 data points collected over a 24 hour period from 42 different paths for $K = 75$, $K_{min} = 15$ and $\epsilon = 5$. As a comparison, for a constant-rate approach with three trains of 25 packets for each rate, the chirp equivalent (in terms of bytes per measurement) is $K = 75$ and $K_{min} = 25$. For the same number of packets, a chirp can probe 50 rates instead of 3. In this case, we choose $K_{min} = 15$ because we did not notice a significant difference in the amount of noise and we are able to probe up to 10 extra rates.

For our likelihood function, we choose the form $L(z = 1|r_p, y_p) = \text{logsig}(\alpha \cdot (r_p - y_p))$. Here α is a constant learned empirically co-jointly with the estimate of y_p through a single regression procedure. The best fit in this regression is determined by minimizing the mean-squared error. The regression identifies $\alpha = -0.27$ with a MSE of 0.08 over the range from $[1, 100]$ Mbps⁷. In theory, the learning procedure

⁶A packet is discarded if the time elapsed between the departure of the previous packet and this one is greater than $\tau(i)$, which means that there was an unusual delay at the sender. A discarded packet is simply excluded from the vector of delays τ' .

⁷We omit a graphical depiction of the functional regression results due to lack of space.

should be repeated every time we estimate a new set of paths. However, from the data collected during multiple experiments, the sigmoid function is consistently a good fit and the value of α rarely changes significantly. It is also interesting to note that the value of α is almost the same as the one obtained for the learning procedure in [5] for a completely different set of paths and measurement methodology (chirps instead of constant-rate packet trains). This suggests that the behaviour of a single window in a chirp is similar to a constant-rate packet train (which validates in part our independence assumption) and that this function is not specific to a single topology.

B. Active Sampling

Before making each measurement, we must choose which path to probe and the range of rates of the chirp. Adaptive selection algorithms, which consist of using information collected with previous measurements to make decisions about the future, can provide important reductions in the number of probes [26].

We employ a probabilistic greedy active learning procedure to select the path to probe at each iteration. The procedure assigns a selection probability to each path that is proportional to the width of the current confidence interval of the path's PAB; it then chooses a path at random according to the assigned probabilities. This means that paths are more likely to be probed if there is more uncertainty about their PABs, and the range of rates probed is mostly likely to contain the PAB of the particular path, given all previous measurements.

From the marginal posterior of each path, we can calculate a confidence range where a specified fraction of the probability mass lies. We use the bounds of the confidence range as the range of rates for the next chirp. For a pre-specified sliding window size, we obtain the desired range of input rates by adjusting the packet spacing and the length of the chirp.

V. TRACKING MODEL AND ALGORITHM

We track the PAB by forming a dynamic model for how it evolves over time. This allows us to implement a forgetting factor; the impact of the first measurement is gradually eliminated because the dynamic model incorporates diffusion. The model we adopt is a dynamic Bayesian network (DBN), which consists of two temporal “slices” related by a transition function [14], [15]. Each slice contains variables interconnected based on their dependencies, including a set of variables used for the distributed representation of the (unknown) belief state; in our case, the PABs. The transition function determines the evolution of the variables between the slices; it is a model for the dynamics of the PABs. In this section, we describe this model in detail and describe the algorithm used to track and estimate the path PAB.

A. Bayesian Inference

Each slice of the DBN is a factor graph; a graphical representation of the factorization of the joint distribution from which we calculate the marginal posteriors [27]. The factor graph is composed of nodes for each variable and factor (in

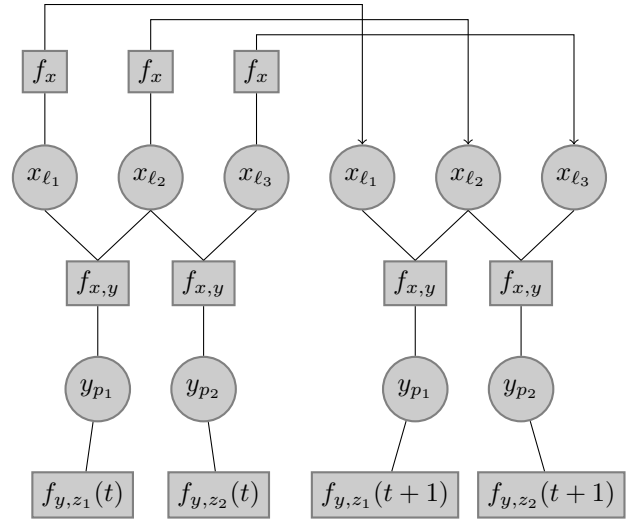


Fig. 2. Two slices of the DBN. Circles represent random variables; PAB of links and path. Rectangles are factors in the joint distribution. f_x is the transition function between the two slices of the DBN.

the distribution of interest) connected through edges based on their dependencies. In our case, the variables consist of the PAB of every path (y_p) and every link (x_ℓ). The first factor, $f_{x,y}$, represents the relation between the PAB of links and paths; the PAB of a path is equal to the minimum of the PABs of all of its constituent links. For example, in Fig. 2, path p_1 is constituted of links ℓ_1 and ℓ_2 . In the Bayesian inference framework, the posterior is proportional to the product of the likelihood and the prior distribution. For each path, there is a likelihood function $f_{y,z}$ equal to the product of all the measurement outcomes on that path during the measurement period. After each measurement, we update the likelihood of the measured path (the existing likelihood is multiplied with the likelihood of the last measurement) and we run the belief propagation algorithm on the factor graph. This allows us to compute the marginal posterior of each path efficiently in order to determine the next most informative measurement (according to the algorithm presented in Sect. IV-B). The only variables in the graph that depend on their value in the previous iteration are the variables that describe the belief state; i.e., the x_ℓ s. The prior distribution of each link is used as a transition function.

B. Belief State

The belief state is represented by the set of random variables x_ℓ , i.e., the link PABs (from which we can calculate all the path PABs). We approximate the belief state with a weighted mixture of Gaussians for each link [28] as follows:

$$\widehat{\Pr}(x_\ell(t)|\mu_\ell^v(t)) = \mathcal{N}(\mu_\ell^v(t), \sigma_\mu) \quad (8)$$

$$\widehat{\Pr}(x_\ell(t)|\mu_\ell(t)) = \sum_{v=1}^{N_v} w_\ell^v(t) \widehat{\Pr}(x_\ell(t)|\mu_\ell^v(t)) \quad (9)$$

where $\mu_\ell(t) = [\mu_\ell^1(t), \dots, \mu_\ell^{N_v}(t)]$ is the vector of means of the Gaussians of link ℓ at time t , $w_\ell^v(t)$ is the weight of each Gaussian in the mixture, $\mathcal{N}(\mu, \sigma)$ is a normal distribution with mean μ and variance σ . Here, σ_μ is taken to be a constant and is equal to 1. Initially, the means of the Gaussians are drawn from a uniform distribution that covers the range of possible PAB, $[B_{min}, B_{max}]$; $\mu_\ell^v \sim \mathcal{U}(B_{min}, B_{max})$ and weights are all equal; $w_\ell^v = 1/N_v$. Sequential importance sampling [16], [17] is employed to update the means and weights of the Gaussians each time the belief is propagated to a new measurement period. Ideally, we would use a N -dimensional weighted Gaussian mixture to approximate the joint distribution of the PAB of each link; $\Pr(x_{\ell_1}, \dots, x_{\ell_N})$, rather than individual mixtures for each link. However, such a large space becomes intractable and the particle filter is known to often fail for high-dimensional problems. Therefore, we consider each link independently (each link has its own weighted Gaussian mixture) and approximate the joint distribution as follows:

$$\Pr(x_1(t), \dots, x_N(t) | z_{1:t}) \approx \prod_{\ell=1}^N \Pr(x_\ell(t) | z_{1:t}). \quad (10)$$

The result of this approximation is that we are propagating the marginals of the links instead of the joint prior in the factor graph. However, the belief propagation algorithm still operates on the joint prior even if it is producing marginals. The approximation significantly decreases the dimensionality and complexity of the problem. It introduces an approximation error, but our simulations and experiments indicate that the approximation is reasonable and leads to acceptable tracking performance.

C. Transition Function

Every time we gather λ observations, we perform the transition between the two slices of the DBN. The means of the Gaussians are propagated to the next slice of the DBN as follows:

$$\mu_\ell^v(t+1) = \mu_\ell^v(t) + \epsilon_h \quad (11)$$

where ϵ_h is sampled from a Gaussian distribution $\mathcal{N}(0, \sigma_h)$.

A product of marginals will produce a more diffuse distribution than the original joint distribution. So by adopting the approximation, we already introduce diffusion in the joint posterior (the new prior). For that reason, we downplay the true variance in the temporal dynamics when we choose the value of σ_h . For all our experiments, we use $\sigma_h = 4$ based on data collected during our online experimentation.

D. Sequential Importance Sampling and Resampling

Once the transition between the two slices is complete, before taking new measurements, we update the weights of the mixtures based on the observations gathered in the previous slice. In the context of a DBN, this procedure is called likelihood weighting [29], [30] and consists of taking the

product of all the observed nodes (for a given link, all the observed paths that include this link).

Let $z_p^\lambda = z_p(t-\lambda+1, t)$ be the observations on path p during the last interval of λ measurements, P_ℓ be the set of paths that include link ℓ and P'_ℓ the subset of P_ℓ for which z_p^λ is not empty (i.e., the set of paths that include link ℓ observed during the last λ measurements). Then, the weights are updated as follows:

$$\tilde{w}_\ell^v(t+1) = w_\ell^v(t) \prod_{p \in P'_\ell} \tilde{w}_{\ell,p}^v(t) \quad (12)$$

$$\begin{aligned} \tilde{w}_{\ell,p}^v(t) &= \Pr(z_p^\tau | \mu(t)) \\ &= \sum_{x_\ell \in L_p} \left[\Pr(z_p^\tau | x_\ell(t)) \widehat{\Pr}(x_\ell(t) | \mu_\ell^v(t)) \right. \\ &\quad \cdot \left. \prod_{l \in L_p \setminus \ell} \widehat{\Pr}(x_l(t) | \mu_l(t)) \right] \end{aligned} \quad (13)$$

where L_p is the set of links composing path p . The factorization in the joint distribution reduces the complexity of the weight computation. Even if the expression still looks computationally expensive, it reduces, for each link in L_p to the multiplication of two $B \times B$ matrices, where $B = B_{max} - B_{min} + 1$ is number of possible discrete values for the PAB. The weights are then normalized as follows:

$$w_\ell^v(t) = \tilde{w}_\ell^v(t) / \sum_{v=1}^{N_v} \tilde{w}_\ell^v(t) \quad (14)$$

Because of the multiplicative update applied to each weight in (12), the importance of some weights may converge towards zero, which effectively makes the associated Gaussian useless and also increases the variance of the estimator. To address this issue, if the number of effective Gaussians, N_{eff}^ℓ , drops below a specified threshold, the means are resampled. The effective number of Gaussians is calculated as follows:

$$N_{eff}^\ell = \frac{1}{\sum_{v=1}^{N_v} w_\ell^{v2}} \quad (15)$$

We use a form of resampling called importance resampling [17], which consists of drawing a new set of N_v samples from a multinomial distribution with parameters N_p and the set of current weights w_ℓ^v . The new weights after this procedure are $1/N_v$.

E. Algorithm

The procedure we described in this section is summarized in Alg. 1. Lines 5-12 represent the measurement methodology described in Sect. IV and the belief propagation. The transition between the two-slices of the DBN occurs at line 13. Lines 14-28 include the particle filtering, the sequential importance sampling and resampling.

```

1 initialize particles  $\mu \sim \mathcal{U}(B_{min}, B_{max})$ ;
2 create DBN;
3  $t = 1$ ;
4 while tracking do
5   repeat
6     select path for next measurement;
7     select rates for next measurement;
8     make measurement;
9     run belief propagation;
10    compute confidence intervals;
11     $t = t + 1$ 
12  until  $t \bmod \lambda = 0$  ;
13  propagate particles using (11);
14  foreach link  $\ell \in \mathcal{L}$  do
15    foreach particle  $v$  do
16      foreach observed path  $p \in P_\ell$  do
17        compute partial weights  $w_{\ell,p}^v$  using (13)
18      end
19    end
20    if link  $\ell$  is observed then
21      update weights using (12) and (14);
22      resample particles (if necessary);
23      update prior using (8) and (9);
24    else
25       $w_\ell^v(t+1) = w_\ell^v(t)$ 
26    end
27  end
28 end

```

Algorithm 1: Belief Propagation Particle Filtering (BP-PF) tracking algorithm

VI. EXPERIMENTS AND RESULTS

A. Matlab Simulations

To assess the performance of our PF-based algorithm, we simulate a network environment where the PAB varies at every time instant. We use a topology extracted from the PlanetLab network that consists of 72 paths (all possible paths between 9 end nodes) and 134 links. The PAB of each link is initially drawn from a uniform distribution in the range $[B_{min} = 1Mbps, B_{max} = 100Mbps]$. The PAB of each path is obtained from the minimum of the PAB of all its constituent links and is defined as $y_p = \max_{r_p} \Pr\{r'_p \geq r_p - 5\} \geq 0.8$ ($\epsilon = 5$ and $\gamma = 0.8$). At every instant t , a measurement is taken. The outcome z is generated based on the likelihood model presented in Sect. IV-A. The PAB of each link also changes at every instant t according the probabilities⁸ shown in Tab. I. The PAB are generated for $T = 1000$ measurements; $1 \leq t \leq 1000$.

Our algorithm produces confidence intervals for each path at every λ measurements, which are used to produce estimates of the PAB, $\hat{y}_p(t)$. For these simulations, the prediction is

⁸The probabilities correspond to a binomial distribution shifted by 2; $\mathcal{B}(x+2; N=4, p=0.5)$.

TABLE I
PROBABILITIES OF LINK PAB VARIATION $\delta_\ell: x_\ell(t+1) = x_\ell(t) + \delta_\ell$.

$p(\delta_\ell = -2)$	$p(\delta_\ell = -1)$	$p(\delta_\ell = 0)$	$p(\delta_\ell = 1)$	$p(\delta_\ell = 2)$
0.0625	0.2500	0.3750	0.2500	0.0625

valid for the upcoming interval of $\lambda = 10$ measurements⁹; $[t, t+10]$. The confidence interval is obtained directly from the marginal posterior of the path in two possible ways: i) smallest confidence interval that includes at least η of the probability mass or ii) confidence interval of size β that has the most probability mass (confidence level). We use the former approach with $\eta = 0.95$. For these simulations, we compare three possible ways to select $\hat{y}_p(t)$: i) lower bound, ii) 25th percentile and iii) median of the confidence interval¹⁰.

From the definition of the PAB and our problem statement, we are interested in determining the largest input rate such that the output rate is within ϵ of the input rate with probability γ . From our likelihood model, any rate smaller or equal to the PAB satisfies that constraint. Such a rate is not necessarily optimal, but it represents a feasible solution; underestimates are preferable to overestimates. To measure the performance of our approach we define three different metrics. First, we evaluate the probability of identifying a feasible solution for path p , $PS_p(t)$, as a running average up to time t :

$$PS_p(t) = \frac{1}{t} \sum_{k=1}^t \mathbf{1}(\hat{y}_p(k) \leq y_p(k)) \quad (16)$$

For any feasible solution, the PAB was underestimated by a certain margin that represents the cost we are paying in terms of input rate. In other words, by how much the input rate could have been increased without reducing the probability of successful transmission below γ . At any instant t where the estimate for path p is smaller or equal to the PAB, we can calculate the cost in rate, $CR_p(t)$, as follows:

$$CR_p(t) = \hat{y}_p(t) - y_p(t) \quad (17)$$

For estimates greater than the PAB, there is no penalty in terms of input rate, but there is one in terms of probability of success. From the likelihood model, any rate greater than the PAB has $L(z=1) < \gamma$. For a path p where the estimate at time t is greater than the PAB, we measure the cost paid in probability of success, $CP_p(t)$, as follows:

$$CP_p(t) = L(z=1|\hat{y}_p, y_p) - \gamma \quad (18)$$

⁹In practice $\lambda = 10$ is the equivalent of a 20 seconds interval. The duration of this time interval must be chosen carefully such that it is long enough to gather sufficient information from measurements to update the PF accurately and short enough to make sure that the predicted values do not become outdated too quickly. From our simulations and online experiments, $\lambda = 10$ is a good compromise, but it is important to mention that this value can be adjusted by the operator and is totally dependent on the network dynamics and the application requirements.

¹⁰To determine the 25th percentile and median values, we first normalize the probabilities so that the sum of probabilities in the confidence interval is equal to 1.

For these three metrics, we compare our PF-based algorithm with two other block-based algorithms. The first one, BB, is similar to our previous approach [5] and assumes that the PAB is constant during the entire estimation procedure. In that context, every measurement has the same weight and the estimate produced at time t uses every observation from $z_{1:t}$. This algorithm is not designed for tracking, but rather to produce confidence intervals for each path that satisfy tightness criteria as fast as possible. The second algorithm, BB-R, is a variation of BB that only uses the last λ measurements to produce the estimates. It basically consists of re-running BB at every λ measurements. This approach discards all the information obtained from previous observations and therefore produces much wider confidence intervals, but reacts more quickly to changes in the system. For these simulations, we used mixtures of $N_v = 100$ Gaussians for each link and a threshold of $N_{eff} = 10$ for resampling. We also simulated for $N_v = [50, 250, 500, 1000]$ and $N_{eff} = [5, 15]$, but did not observe any significant variation on the results.

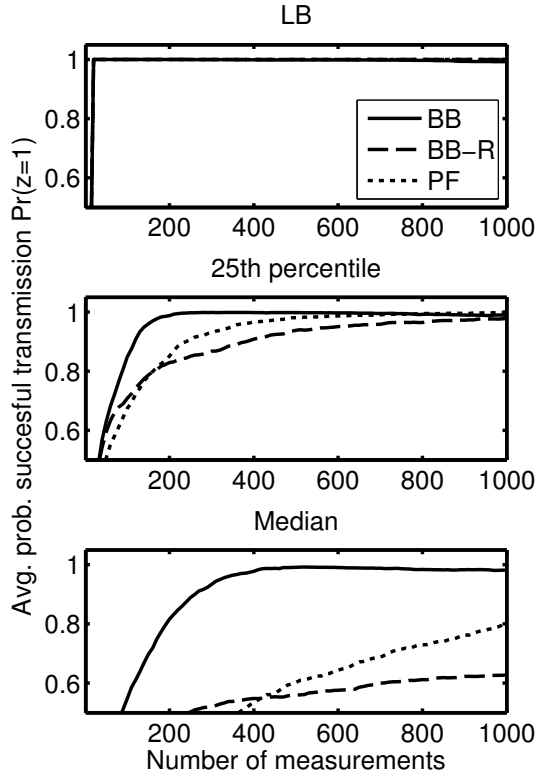


Fig. 3. Running average probability of successful transmission as a function of the number of measurements. Probability is averaged over 72 paths and 30 simulations. Three selection modes for estimate of PAB $\hat{y}_p(t)$: the lower bound of the confidence interval (TOP), 25th percentile of the confidence interval (MIDDLE) and median of the confidence interval (BOTTOM).

In Fig. 3, we show the evolution as a function of the number of measurements of the probability of successful transmission for the three proposed methods. The value at t is first averaged over all time instants and then over all 72 paths. The experiment was repeated 30 times with different link PAB initial values. For the first two selection methods

(lower bound, 25th percentile, median), all techniques are able to avoid congestion almost all the time. However, when probing at the median, only the BB approach is able to avoid congestion; our PF-based approach drops to 80% congestion avoidance whereas the BB-R approach drops to almost 60%. This metric is not really significant by itself because it is easy to avoid congestion if the estimate is conservative and much lower than the PAB. This is the reason why we also look into the cost in terms of input rate and percentage of success.

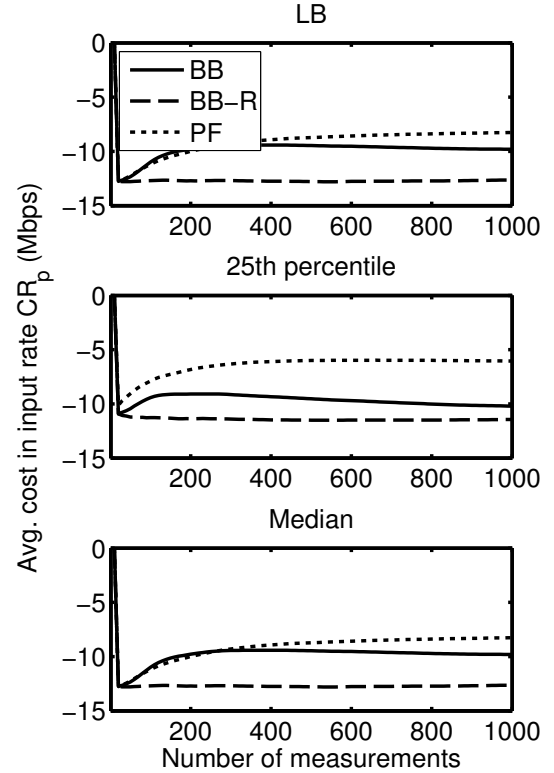


Fig. 4. Cost in terms of input rate when estimate is smaller than the PAB. Running average of the difference between the input rate and PAB. The difference is averaged over 72 paths and 30 simulations. Three selection modes for estimate of PAB $\hat{y}_p(t)$: the lower bound of the confidence interval (TOP), 25th percentile of the confidence interval (MIDDLE) and median of the confidence interval (BOTTOM).

In Fig. 4, we show the cost in terms of input rate for estimates smaller than the PAB. The values represent running averages over all 72 paths and time instants, i.e., over $M \times t$ values at time t . The experiment was repeated 30 times with different link PAB initial values and the running averages are averaged over all these experiments. The PF outperforms the block-based approaches for all three estimation choices. The cost decreases when the input rate is chosen to be around the 25th percentile to around 5 Mbps, which is 5Mbps closer than other algorithms. The BB always underestimates the PAB by 10 Mbps. This can be explained by the fact that the confidence intervals become very tight after a few hundred measurements and there is not much difference between the lower bound and the median. On the other hand, the BB-R algorithm always underestimates the PAB significantly.

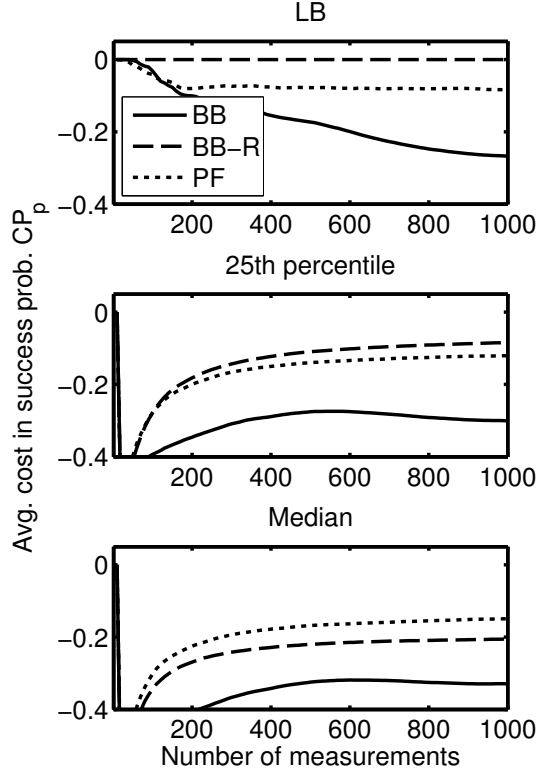


Fig. 5. Cost in terms of percentage of successful transmission when estimate is greater than the PAB. Running average of the difference between the likelihood of successful transmission $L(z = 1)$ and $\gamma = 0.9$. The difference is averaged over 72 paths and 30 simulations. Three selection modes for estimate of PAB $\hat{y}_p(t)$: the lower bound of the confidence interval (TOP), 25th percentile of the confidence interval (MIDDLE) and median of the confidence interval (BOTTOM).

In Fig. 5, we look at the cost in percentage of successful transmission for estimates greater than the PAB. Whereas BB performed better in terms of cost in input rate, BB-R was the better approach in terms of cost in percentage. PF outperformed BB in all three cases and was slightly under or over BB-R. Combining the results from all three metrics, the PF approach provides the best balance and manages to reduce the cost input rate difference without affecting the probability of success unlike the other block-based algorithms. In that case, increasing the input rate to the 25th percentile of the confidence interval would incur minimal cost in terms of input rate (-5) or percentage (-10%).

In Fig. 6, we show an example of tracking for a single path when the estimate is at the 25th percentile of the confidence interval. The PF approach provides a much closer estimate than the two block-based approaches. We also note that the PF estimate becomes more accurate after approximately 400 measurements while the other approaches are unaffected by time. In the future, we intend to adjust our approach such that fewer measurements are required before the tracking reaches an acceptable level of accuracy.

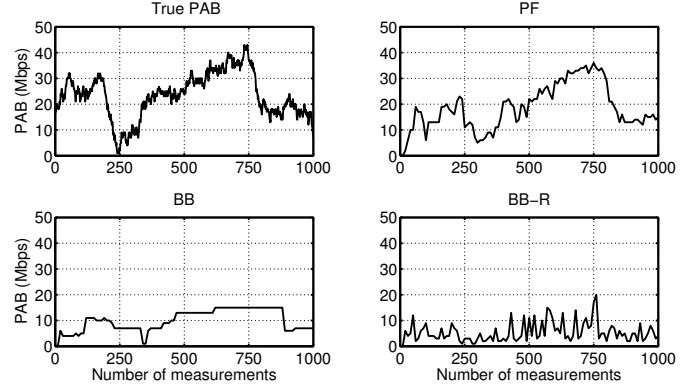


Fig. 6. Tracking the PAB of a single path for 1000 iterations when the estimate is equal to the 25th percentile value of the confidence interval.

B. Online Experiments

We implemented our tracking algorithm into a tool that we deployed on the PlanetLab network. For our experiments, we test our approach on a topology that consists of 8 nodes¹¹, 56 paths and 119 logical links. We use the exact same parameter settings as in the simulations except that we set $\gamma = 0.8$. Our testing procedure consists of running the tracking algorithm for 300 measurements and probing some of the paths at regular intervals based on our estimated PAB (each experiment lasts approximately 20 minutes). More specifically, we construct the smallest subset of paths that includes every link at least once (in this case, the test set includes 36 of the 56 paths). After every sequence of 10 measurements, we send a constant-rate train of 500 packets on 3 of the paths from the test set at an input rate equal to our estimate of the PAB for that path. We then calculate the average output rate and the binary outcome $z = \mathbf{1}(r'_p \geq r_p - \epsilon)$. We observe the probability of success $\Pr(z = 1)$ when the estimated PAB is chosen to be 1) the lower bound, 2) the 25th percentile value and 3) the median of the confidence interval. We repeated the experiment five times and present the results in Tab. II.

TABLE II
PLANETLAB RESULTS. PROBABILITY OF SUCCESS AND MEDIAN OVERESTIMATE IN CASE OF FAILURE FOR THREE ESTIMATED PABS.

	LB	25th perc.	Median
$\Pr(z = 1)$	0.97 ± 0.02	0.84 ± 0.08	0.7 ± 0.1
OE (Mbps)	10 ± 7	3 ± 2	5 ± 2
$\Pr(OE > 10)$	0.013 ± 0.009	0.03 ± 0.03	0.05 ± 0.02

The probability of successful transmission, $\Pr(z = 1)$ is averaged over the 90 tests in each experiment. As we observed in the simulations, there is a deterioration in accuracy when we choose the PAB estimate at the median of our confidence interval. However, for the lower bound and 25th percentile, we satisfy our constraint that $\Pr(z = 1)$ should be larger than $\gamma = 0.8$. We also looked at the running average of

¹¹planetlab-01.cs.princeton.edu, planetlab3.csail.mit.edu, planetlab2.rutgers.edu, pl2.csl.utoronto.ca, planetlabone.ccs.neu.edu, planetlab2.csg.uzh.ch, planetlab1.millennium.berkeley.edu, planet1.zib.de

the probability of success in time, but there was very little variation. We noticed that, in all experiments, a small subset of paths was responsible for a large fraction of the failed tests. As future work, we intend to investigate this problem and identify the characteristics of these paths that make them harder to estimate accurately using our method. In case of failure, we can calculate the overestimation margin (OE); the difference between the estimate (probed rate) and the largest input rate that would satisfy our constraint: $\hat{y}_p - (r'_p + \epsilon)$. The median OE for lower bound estimates is much larger than for the other selection methods due to the fact that there were very few overestimates. In the other case, the over estimation is generally pretty low; under 5 Mbps. We also look at the probability of an overestimation larger than 10 Mbps. In all cases, that probability was under 5%.

VII. CONCLUSION

In this paper, we attacked the problem of tracking network-wide probabilistic available bandwidth for multiple paths. We proposed a methodology based on chirps, belief propagation and particle filtering that we implemented in a practical and lightweight tool. According to our simulations, our approach can track PAB with more accuracy at the same cost than block-based approaches. We also deployed the tool in a practical environment and showed through online experimentation that it can produce estimates that satisfy our constraints in terms of probability of successful transmission (output rate almost as large as the input rate). In the future, we intend to simulate our tool in an environment where the PABs can undergo large sudden changes or where there is a mismatch between the PAB evolution and our tracking model.

REFERENCES

- [1] Y. Hiraoka, G. Hasegawa, and M. Murata, "Effectiveness of overlay routing based on delay and bandwidth information," in *Proc. Australasian Telecommunication Networks and Applications Conf.*, Christchurch, New Zealand, Dec. 2007.
- [2] S.-J. Lee, S. Banerjee, P. Sharma, P. Yalagandula, and S. Basu, "Bandwidth-aware routing in overlay networks," in *Proc. IEEE Int. Conf. Computer Communications*, Phoenix, AZ, Apr. 2008.
- [3] L. He, S. Yu, and M. Li, "Anomaly detection based on available bandwidth estimation," in *Proc. IFIP Int. Conf. Network and Parallel Computing*, Shanghai, China, Oct. 2008.
- [4] C. D. Guerrero and M. A. Labrador, "On the applicability of available bandwidth estimation techniques and tools," *Computer Communications*, vol. 33, no. 1, pp. 11–22, Jan. 2009.
- [5] F. Thouin, M. Coates, and M. Rabbat, "Multi-path Probabilistic Available Bandwidth Estimation through Bayesian Active Learning," vol. arXiv:1001.1009, Jan. 2010.
- [6] S. Ekelin, M. Nilsson, E. Hartikainen, A. Johnsson, J.-A. Mångs, B. Melander, and M. Björkman, "Real-time measurement of end-to-end available bandwidth using kalman filtering," in *Proc. IEEE/IFIP Network Operations and Management Symp.*, Vancouver, Canada, Apr. 2006.
- [7] E. Bergfeldt, S. Ekelin, and J. Karlsson, "Real-time available-bandwidth estimation using filtering and change detection," *Computer Networks*, vol. 53, no. 15, pp. 2617–2645, Oct. 2009.
- [8] A. Cabellos-Aparicio, F. Garcia, and J. Domingo-Pascual, "A novel available bandwidth estimation and tracking algorithm," in *Proc. Network Operations and Management Symp. Work.*, Salvador da Bahia, Brazil, Apr. 2008.
- [9] Y. Yang, X. Wen, X. Lin, W. Zheng, and Y. Sun, "A Novel Available Bandwidth Estimation Scheme in Probe Packets Losing Condition," in *Proc. Int. Symp. Information Engineering and Electronic Commerce*, Beijing, China, May 2009.
- [10] Z. Bozakov and M. Bredel, "Online estimation of available bandwidth and fair share using kalman filtering," in *Proc. Int. Conf. on Networking*, Aachen, Germany, May 2009.
- [11] N. Hu and P. Steenkiste, "Exploiting internet route sharing for large scale available bandwidth estimation," in *Proc. ACM SIGCOMM Internet Measurement Conf.*, Berkeley, CA, Oct. 2005.
- [12] B. Maniymaran and M. Maheswaran, "Bandwidth landmarking: A scalable bandwidth prediction mechanism for distributed systems," in *Proc. IEEE Global Telecommunications Conf.*, Washington, DC, Nov. 2007.
- [13] P. Yalagandula, S.-J. Lee, P. Sharma, and S. Banerjee, "Correlations in end-to-end network metrics: Impact on large scale network monitoring," in *Proc. IEEE Int. Conf. Computer Communications Work.*, Phoenix, AZ, Apr. 2008.
- [14] Z. Ghahramani, *Adaptive Processing of Sequences and Data Structures*. Berlin: Springer-Verlag, 1998, vol. 1387, ch. Learning Dynamic Bayesian Networks, p. 168.
- [15] K. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. dissertation, University of California Berkeley, Jul. 2002.
- [16] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *J. of the American statistical association*, vol. 93, no. 443, pp. 1032–1044, 1998.
- [17] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [18] E. Goldoni, G. Rossi, and A. Torelli, "Assolo, a new method for available bandwidth estimation," in *Proc. IEEE Int. Conf. Internet Monitoring and Protection*, Venice, Italy, May 2009.
- [19] D. Croce, M. Mellia, and E. Leonardi, "The quest for bandwidth estimation techniques the quest for bandwidth estimation techniques for large-scale distributed systems," in *Proc. ACM Work. Hot Topics Measurement and Modelling of Computer Systems*, Seattle, WA, Jun. 2009.
- [20] M. J. Coates and R. D. Nowak, "Sequential monte carlo inference of internal delays in sequential monte carlo inference of internal delays in nonstationary communication networks," *IEEE Trans. Signal Processing, Special Issue on Monte Carlo Methods for Statistical Signal Processing*, vol. 50, no. 2, pp. 366–376, Feb. 2002.
- [21] R. Carter and M. Crovella, "Measuring bottleneck link speed in packet-switched networks," *Performance evaluation*, vol. 27, pp. 297–318, 1996.
- [22] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?" in *Proc. IEEE Int. Conf. Computer Communications*, Anchorage, AK, Apr. 2001.
- [23] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput," *IEEE/ACM Trans. Networking*, vol. 11, no. 4, pp. 537–549, Aug. 2003.
- [24] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk, "Multifractal cross-traffic estimation," in *Proc. ITC Specialist Seminar on IP Trac Measurement, Modeling, and Management*, Monterey, CA, Sep. 2000.
- [25] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil, and L. Cottrell, "pathchirp: Efficient available bandwidth estimation for network paths," in *Proc. Passive and Active Measurement Conf.*, La Jolla, CA, Apr. 2003.
- [26] R. Castro and R. Nowak, "Active learning and sampling," in *Foundations and Applications of Sensor Management*, A. Hero, D. Castanon, D. Cochran, and K. Kastella, Eds. Springer-Verlag, 2007, pp. 177–200.
- [27] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco, CA: Morgan Kaufmann, 1988.
- [28] J. Kotecha and P. Djuric, "Gaussian sum particle filtering," *IEEE Trans. Signal Processing*, vol. 51, no. 10, pp. 2602–2612, 2003.
- [29] R. M. Fung and K.-C. Chang, "Weighting and integrating evidence for stochastic simulation in weighting and integrating evidence for stochastic simulation in bayesian networks," in *Proc. Conf. Uncertainty in Artificial Intelligence*, Windsor, Canada, Aug. 1989.
- [30] R. Shachter and M. Peot, "Simulation approaches to general probabilistic inference on belief networks," in *Proc. Conf. Uncertainty in Artificial Intelligence*, Windsor, Canada, Aug. 1989.