Resource Management with Hoses: Point-to-Cloud Services for Virtual Private Networks

N.G. Duffield, Pawan Goyal[1], Albert Greenberg
Partho Mishra[2], K.K. Ramakrishnan and Jacobus E. van der Merwe
AT&T Labs–Research, 180 Park Avenue, Florham Park, NJ 07932, USA

Abstract—

As IP technologies providing both tremendous capacity and the ability to establish dynamic security associations between endpoints emerge, Virtual Private Networks (VPNs) are going through dramatic growth. The number of endpoints per VPN is growing and the communication pattern between endpoints is becoming increasingly hard to forecast. Consequently, users are demanding dependable, dynamic connectivity between endpoints, with the network expected to accommodate any traffic matrix, as long as the traffic to the endpoints does not overwhelm the capacity of the respective ingress and egress links. We propose a new service interface, termed a *hose*, to provide the appropriate performance abstraction. A hose is characterized by the aggregate traffic to and from one endpoint in the VPN to a set of other endpoints in the VPN, and by an associated performance guarantee.

Hoses provide important advantages to a VPN customer: (i) flexibility to send traffic to a set of endpoints without having to specify the detailed traffic matrix, and (ii) reduction in the size of access links through multiplexing gains obtained from the natural aggregation of the flows between endpoints. As compared with the conventional point-to-point (or customer-pipe) model for managing QoS, hoses provide reduction in the state information a customer must maintain. On the other hand, hoses would appear to increase the complexity of the already difficult problem of resource management to support QoS. To manage network resources in the face of this increased uncertainty, we consider both conventional statistical multiplexing techniques, and a new *resizing* technique based on online measurements.

To study these performance issues, we run trace driven simulations, using traffic derived from AT&T's voice network, and from a large corporate data network. From the customer's perspective, we find that aggregation of traffic at the hose level provides significant multiplexing gains. From the provider's perspective, we find that the statistical multiplexing and resizing techniques deal effectively with uncertainties about the traffic, providing significant gains over the conventional alternative of a mesh of statically sized customer-pipes between endpoints.

Keywords: Service Level Agreements, Quality of Service, Point-to-Multipoint, Point-to-Cloud

I. Introduction

Virtual Private Network services have been offered in various forms over an extended period of time and have recently received considerable attention within the IP, frame-relay, MPLS, and ATM networking communities [1], [2], [3]. VPNs are likely to be used by customers as a replacement for networks constructed using private lines and should therefore, at the very least, provide a comparable service. Substantial progress in the technologies for IP security [4] enable us to improve on the security and privacy provided in existing VPN service offerings based on private lines or frame-relay. Other work on IP-based VPNs has mainly dealt with group membership, routing protocols and tunneling [3]. Much less attention has been paid to resource management issues related to VPNs. However, supporting a variety of mission-critical functions requires a VPN service to provide performance assurances, backed by Service Level Agreements (SLAs). Private lines isolate the performance seen by a VPN from other flows and provide guaranteed bandwidth, loss and delay characteristics. A VPN service must offer comparable performance assurances. Our focus in this paper is on the performance issues related to VPNs.

Due to the progress in security and the overwhelming success of IP networking technologies, the number of endpoints per VPN is growing, and communication patterns between endpoints are becoming increasingly difficult to forecast. We expect that users will be unwilling to, or simply unable to predict loads between pairs of endpoints. Similarly, it will become increasingly difficult to specify QoS requirements on a point-to-point basis, the conventional approach. Our solution, which we call the *hose model* serves as both a VPN service interface (i.e., the way a customer thinks of a VPN) as well as a performance abstraction (i.e., the way a provider thinks of a VPN). A hose offers performance guarantees from a given endpoint to the set of all other endpoints in the VPN, and for the traffic to the given endpoint from the set of all other endpoints in the VPN. The hose is the customer's interface into the network, and is the equivalent of the customer having a "link" into the network. The hose service interface allows the customer to send traffic into the network without the need to predict point-to-point loads.

Though the hose model provides customers simpler, more flexible SLAs, the model appears to present the provider with a more challenging problem in resource management. Under the conventional point-to-point model for specifying QoS, there is uncertainty about temporal variation in the traffic between the two points. Under the hose model, there is also spatial uncertainty; i.e., uncertainty about traffic sinks. To cope with these uncertainties, we develop mechanisms that allow providers to use the hose model to achieve significant multiplexing gains in the network, by the use of signaling to dynamically size hose and network capacity.

A hose is a service level assurance for a point to cloud VPN. This paper considers essentially the uncapacitated design prob-

[1] Currently with IBM Corporation
[2] Currently with Woodside Networks, Inc.

lem, i.e., how much capacity is needed to support the hoses. In particular, we wish to determine the cost incurred by the provider in providing sufficient capacity to accommodate traffic whose matrix is not completely known.

We evaluate the proposed hose VPN service model by performing a number of trace driven experiments. In particular we show that significant multiplexing gains may be achieved for both the customer and the provider when the network is capable of exploiting the hose model. Two sets of traces were used for these experiments. The first was voice traffic traces from the AT&T backbone network. The second was data traffic traces from a large corporate backbone network.

The rest of the paper is organized as follows. In the next section, the hose model for VPNs is presented. Section III describes implementation scenarios, and the traffic predictors we used in our experiments to estimate required capacity. After an outline of the simulation framework in Section IV, we briefly consider the variability of the traffic matrix in Section V based on an analysis of the data traffic traces. The section then continues by examining the benefit of the hose from the perspective of a customer. In Section VI we look at the multiplexing benefits within the provider's network and examine the performance of alternative means of implementing a hose in the network. Of interest is the reduction in capacity as we dynamically resize the amount of resources used to adapt to changing traffic needs. We address issues of arriving at an effective bandwidth for admission control in Section VII, and conclude in Section VIII.

II. The Hose Service Model

A simple service model for an IP VPN is to emulate the private line or frame relay service. This would require a customer to buy a set of **customer-pipes**, i.e., allocations of specific bandwidth on paths between source-destination pairs of endpoints of the VPN (much like virtual circuits). Figure 1 illustrates an example of the use of this kind of interface. The network provider would need to provision adequate bandwidth along the path of each pipe to ensure that the Service Level Agreement (SLA) is satisfied. The primary disadvantage of this approach is that it requires the customer to have precise knowledge of the traffic matrix between all the VPN sites. Resources made available to a customer-pipe cannot be allocated to other traffic. It is important to note the network provider may not be able to take advantage of statistical multiplexing gains across the customer-pipes.

In this paper, we propose a richer and more flexible VPN service model that we refer to as a **hose**. In the hose model, a VPN customer specifies a set of endpoints to be connected with common endpoint-to-endpoint performance guarantees. The connectivity of each endpoint to the network is specified by a hose,

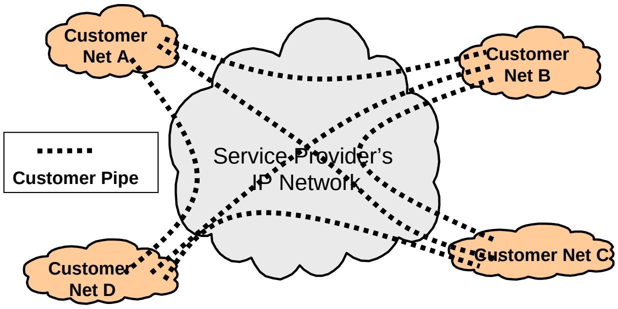

Fig. 1. A VPN Based on the Customer-Pipe Model. A mesh of customer-pipes is needed, each extending from one customer endpoint to another. A customer endpoint must maintain a logical interface for each of its customer-pipes.

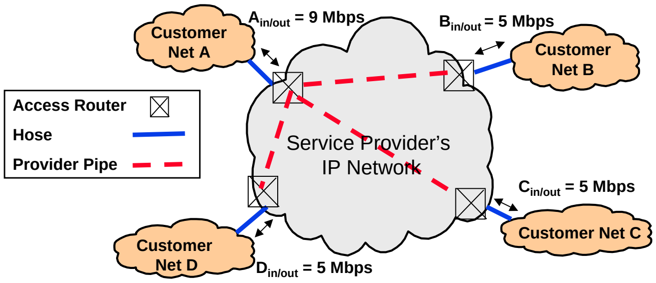

Fig. 2. A VPN Based on the Hose Model. A customer endpoint maintains just one logical interface, a hose, to the provider access router. In the Figure, we show the implementation of one hose (based at A) using provider-pipes.

comprising:

- the capacity required for aggregate outgoing traffic from the endpoint into the network (to the other endpoints of the VPN)
- the capacity required for aggregate incoming traffic out of the network to the endpoint (from the other endpoints of the VPN);
- the performance guarantee for the hose, conditioned only on the *aggregate* traffic seen at the hose interface.

The notion of virtual private links (essentially ”customer-pipes”) was introduced at the 40th IETF in December 1997, and subsequently laid out in an Internet Draft in 1998 [5]. A number of efforts have built on the concept, with the term ”point-to-cloud VPN's” being used to describe hose-based VPNs. Implementation issues for VPNs in general are addressed in [3], [6]. Additional work examining the performance of the hose model have been described in [7], [8].

Figure 2 illustrates an example of the use of hoses. Consider 4 VPN sites: A, B, C, and D. The customer buys 4 hoses at each of these sites and specifies the aggregate outgoing and incoming traffic for each of these hoses. The hose specification may be arrived at in a variety of ways. For example, if it is known that each of the sites B, C and D sends and receives at no more than 3 Mb/s to site A, and that each of these sites sends and receives no more than 2 Mb/s *in aggregate* to each other, then the hose capacity would be chosen as: $A_{in} = A_{out}$ =9 Mb/s and $B_{in} =$

$B_{out} = C_{in} = C_{out} = D_{in} = D_{out} =$5 Mb/s. Figure 2 depicts one possible realization of the connectivity from hose A by means of a set of provider-pipes.

There are several advantages of the hose model from a customer's perspective:

Ease of Specification: Only one inward and outward rate (possibly asymmetric) per hose endpoint needs to be specified, as compared with that for each customer-pipe between pairs of endpoints.

Flexibility: Data to and from a given hose endpoint can be distributed arbitrarily over other endpoints provided the aggregate conforms to the hose size.

Multiplexing Gain: Due to statistical multiplexing gain, hose rates can be less than the aggregate rate required for a set of customer-pipes.

Characterization: Hose requirements are easier to characterize because the statistical variability in the individual source-destination traffic is smoothed by aggregation into hoses.

The nature of the service level agreement between a customer and a service provider is driven by the traffic characteristics and QoS requirements of the (customer) applications that make use of the VPN. For example, an IP voice VPN service might require tight bounds on the per-packet loss rates, delay and possibly jitter. On the other hand, a data-only VPN service might have relatively less stringent delay requirements. To ensure that the appropriate requirements can be met, it is essential for the customer to provide a description of the traffic characteristics.

The complexity of the traffic specification provided by a customer is expected to vary depending on the needs and sophistication of the customer. A reasonable service offering would indeed be for customers to start off with a fairly simple specification and to then refine this specification based on operational experience and service provider feedback.

Providing an initial estimate of the VPN specification is a service that might be offered by the provider to customers as part of a **VPN characterization phase**. The customer (or the provider) can monitor the performance of the VPN in terms of loss and delay to determine whether it satisfies the needs of the customer's applications, and then negotiate a different capacity and the corresponding SLA.

During the characterization phase, the SLA for the VPN might be undefined or might be defined as some best-effort service. Alternatively, provider may act conservatively and overprovision in terms of the resources it allocates for the VPN during this phase, and provide QoS assurances.

Service level agreements following the characterization phase might be based on the current traffic load with provisions made for expected gradual growth as well as expected drastic traffic changes that the customer might foresee (or protect against).

Both the customer and the provider may play a role in testing whether the SLAs are met. The provider may police (and possibly shape) the incoming traffic to a hose from the customer's access link to ensure that it stays within the specified profile. Similarly, traffic leaving the VPN at a hose egress (i.e., traffic potentially generated from multiple sources that has traversed the network) may have to be monitored and measured at the hose egress point, to ensure that such traffic stays within the specified profile and that the provider has met the SLA. The customer might also be required to specify a policy for actions to be taken should egress traffic be more than the specified egress hose capacity.

A. Capacity Management

From a provider's perspective, it is potentially more challenging to support the hose model, due to the need to meet the SLAs with a very weak specification of the traffic matrix. To manage resources so as to deal with this increased uncertainty, we consider two basic mechanisms:

Statistical Multiplexing: In order to decrease aggregate bandwidth requirements, a provider can consider multiplexing together different traffic flows that are subject to the same QoS assurance. This applies at three different levels of aggregation. First, as a single QoS assurance applies to a hose, all the traffic of that hose can be multiplexed. Second, the hoses making up a VPN have common QoS assurance and can be multiplexed together. Third, distinct VPNs that have the same QoS assurance can be multiplexed together. These techniques can be applied on both access links and network internal links. In either case, network elements must be able to map each packet to its appropriate multiplex, either implicitly or explicitly.

Resizing: In order to provide tight QoS assurances, the provider may use (aggregate) network resource reservation mechanisms that allocate capacity on a set of links for a given hose or VPN. A provider can take the approach of allocating this capacity statically, taking into account worst case demands. Alternatively, a provider can make an initial allocation, and then resize that allocation based on online measurements. Again, such techniques can be applied on both access and network internal links. Resizing is allowed only within the envelope defined by the SLA. Resizing would occur at a substantially finer time scale than the time scale over which SLA's might be renegotiated.

These two resource management mechanisms can be used separately or in combination.

Some more remarks are in order on resizing. Provisioning decisions normally have an impact over fairly long timescales. Within the context of a VPN framework, measurements of actual usage can be used on much shorter timescales to enable efficient capacity management. Underlying this is an assumption that, within the network, boundaries will exist between resources that might be used by different classes of traffic to ensure that performance guarantees are met. For example, traffic from different VPNs might be isolated from each other, and from other classes of traffic. In the context of this paper, resources available for VPN traffic cannot be used by other traffic requiring performance guarantees. We assume that this perspective holds whether the boundaries reflect reservation of resources, such as in the case of IntServ, or whether it represents some allocation in a bandwidth broker in a DiffServ environment.

If we can use the measurements of actual usage to resize the boundary for a given VPN's traffic, more bandwidth will be made available to other traffic and we can make better use of available capacity. In reality, measurements of current usage would be used to make a prediction about near term future usage, and this prediction will be used to resize the share of resources allocated.

In the hose model, this approach can be realized by allowing customers to resize the respective hose capacities of a VPN. Presumably there will be some cost incentive for customers to resize their hose capacities. While we envisage this mechanism to be mainly used to track actual usage, by exposing this interface to the customer, it would also enable the customer to resize its hose capacities based on local policy decisions.

How frequently hoses may be resized will depend on implementation and overheads for resizing and measurement. More important, however, is whether frequent resizing is beneficial and whether it is possible to make predictions with sufficient accuracy. Finally, short timescale resizing is not a replacement for provisioning and admission control and the appropriate relationship between these resource management approaches is important.

III. Realizing the Hose Model

The flexibility offered by the hose interface presents a number of challenges in terms of its realization. In this section we examine the various alternatives that a provider has for implementing a hose. Section III-A considers the various implementation scenarios in a technologically neutral fashion.

The realizations described here all require: (1) a method for measuring the traffic and based on such measurements, predicting the required capacity, and (2) signaling protocols for dynamically reserving resources based on predicted capacity requirements. Therefore, in Section III-B, we present techniques for predicting the required capacity. The design of appropriate signaling mechanisms is the subject of future work.

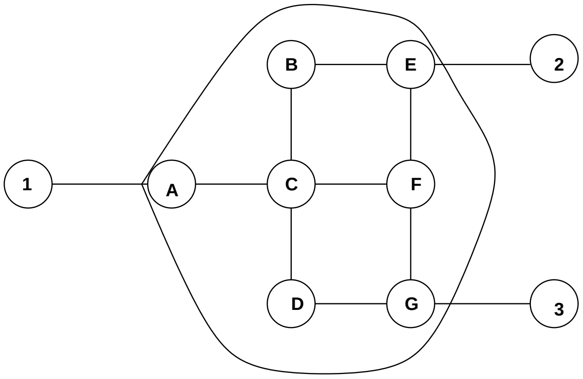

Fig. 3. An example network to illustrate various implementation possibilities. 1-3 represent customer routers, and A-G represent provider routers.

A. Implementation Scenarios

A.0.a Provisioned VPNs. Consider the example network shown in Figure 3. Let 1, 2, and 3 be the customer routers that are part of a VPN, each originating a hose of size 1 Mb/s. To simplify the discussion, let us assume that capacity is reserved for hoses based on a worst-case traffic split, i.e., the traffic from each hose can be directed entirely to just one other endpoint. To begin with, a provider may not want to employ any hose specific state in the network, relying simply on default routing and not making use of any resource sharing (in this case the hose model is only a service interface for the customer.) Then, the hose originating from, say, router 1, can be implemented by reserving two "provider-pipes" from ingress router A to egress routers E and G[1]. Since a worst-case traffic split is being assumed, the capacity of both the provider-pipes are 1 Mb/s. To determine the total resources reserved in the network, let us assume that the default shortest paths from A to E and G are A-C-B-E and A-C-D-G, respectively. Thus, the total capacity required (i.e. summed across all links) when a hose is implemented using provider-pipes is 6 Mb/s.

Observe that on link A-C, 2 Mb/s is reserved even though from the hose specification we know that at most 1 Mb/s worth of traffic may enter the network from source 1 and thus be present on that link at any time. To reduce the reservation to 1 Mb/s on link A-C, the provider can utilize the *source tree* rooted at A and constructed using the default shortest paths to destinations E and G. On each edge that belongs to this tree, 1 Mb/s is reserved. Since 5 links belong to the source tree rooted at A, the total capacity required is reduced to 5 Mb/s. However, to achieve this reduction, the provider is required to know about and make use of hose specific state in the network in the form of a source tree. Since default shortest path routing is being assumed, this hose state is limited to control of *resource sharing*.

[1] We use the term provider-pipe to clearly indicate that these pipes are not exposed in the customer interface.

A provider can achieve further reduction in the capacity by employing hose specific state not only in the sharing of resources but also by making use of non-default or *explicit routing*. For example, let the forwarding tables be changed such that traffic from A to E and G is forwarded along paths A-C-F-E and A-C-F-G, respectively. Though these are also shortest paths, these were not considered the default shortest paths in our example. In this case, by employing resource sharing as before, as well as explicit routing, the total reserved capacity reduces to 4 Mb/s.

The above implementation alternatives only exploit hose specific state. A provider can achieve further reduction by additionally exploiting *VPN specific state*. To observe this, assume that the routing occurs as in the previous scenario for hose 1 and shortest path routing is employed for hoses 2 and 3, i.e., for hose 1 the tree is A-C-F-E and A-C-F-G, for hose 2 it is E-F-C-A and E-F-G and for hose 3 it is G-F-E and G-F-C-A. If we limit ourselves to hose specific state only then the total reserved capacity in the network for the VPN is 12 Mb/s (4 Mb/s for each hose). Now, consider the link FG. On link FG, capacity of 2 Mb/s (1 Mb/s each for hoses 1 and 2) is being reserved for traffic that is destined for node 3. However, from the specification of hose 3, we know that at most 1 Mb/s may be received by 3. Hence, by recognizing that hoses 1 and 2 belong to the same VPN, a provider can reduce the capacity reserved on link FG to 1 Mb/s. A similar reduction of 1 Mb/s can be achieved on links FE, FC, and CA. Thus, by exploiting VPN specific state, the aggregate reserved capacity can be reduced from 12 Mb/s to 8 Mb/s.

The above implementation alternatives are applicable even when a network does not accommodate the capability to signal the traffic requirements, e.g., in the case where VPNs are provisioned. This case is of practical importance as initial VPN offerings might be realized in this fashion. We investigated this in our simulation experiments. In this type of realization, however, a provider has no alternative but to allocate for the peak rate and to assume the worst case traffic split from the hose, because the hose model explicitly allows changes in the traffic matrix. Without the capability to signal the traffic requirements, this can lead to significant underutilization of resources in a network. To address this inefficiency, we propose an extension to the above set of implementations, which could lead to a better utilization of network resources.

A.0.b Dynamically Resized VPNs. The essential idea is to use online measurements to determine the capacity requirements of hoses and to then dynamically adapt the amount of resources reserved based on such measurements. We assume the existence of appropriate mechanisms to achieve these two functions. Broadly, the mechanisms are as follows:

- Service Level Agreements. We focus on the rate (or capacity) as the technical specification of the SLA, although a more general technical specification could be incorporated into the SLA.
- Measurements to guide how much capacity is needed to support the hose for a VPN. Sections 4, 5 and 6 of the paper examine the capacity requirements for a variety of scenarios.
- Admission control mechanisms for hoses of a desired capacity, when the assurance is provided that the SLA would be met.
- Signaling procedures to ensure that the capacity requirements are not violated.

The implementation alternatives are as follows.

Resized Provider Pipes: In this case, a hose is implemented by a mesh of provider-pipes between the ingress and egress routers of a VPN. The resizing of these pipes is done from the ingress edge router: it measures the traffic for each provider-pipe, and based on these measurements predicts the size of the pipes required. It then signals the reservation for each of the provider-pipes. Since this implementation does not use any hose specific state in routing or resource reservation (i.e., no sharing), it can be realized in any IP network that can support dynamically resized pipes, e.g., IntServ, DiffServ or MPLS networks.

Resized Trees: A hose is realized by a source based tree. The aggregate hose traffic is measured at each link and resources are reserved for the aggregate. However, as we traverse down the tree, we measure and reserve resources for progressively smaller aggregates.

Instead of employing a source based tree, we can also employ a sink based tree. For example, in Figure 3, we can employ a sink tree rooted at A with E and G being the leaves. In this case, the reservation is for traffic flowing out of the hose into router 1. Reservation of resources occurs in a manner analogous to the source based tree.

Since this implementation does not have any hose specific state in routing, it can be realized in an IP network that has only default routing. However, compared to the first case, the signaling protocols that enable reservation of resources have to make use of hose specific information. We will use the terms hose and tree interchangeably for this implementation alternative.

Resized Trees with Explicit Routing: In this case, a hose may be realized using a Steiner tree, i.e., a minimum-weight tree connecting all the hose endpoints. Measurement and signaling for resources occur as in the previous case. But, the main advantage of this approach over the previous one is that it increases the number of links on which resources are shared. This implementation, however, can only be realized in networks that support explicit routing (for example, MPLS) and also requires the signaling protocol to be hose aware.

Resource Aggregation across a VPN : In this case, we first route each of the hoses using the default shortest path routing or Steiner trees (i.e., either of the previous two approaches). Then on the links of the graph resulting from the union of the hose trees, we measure the aggregate traffic for the VPN and reserve resources appropriately. This requires the signaling and measurement mechanisms to associate different hoses to the VPN they belong to.

B. Prediction of Traffic Rates

In this section we describe schemes for predicting future traffic rates of a traffic flow from measurements. As explained in the previous sections, such predictions are used to dynamically change or renegotiate the resources associated with a VPN. Specifically, we are interested in flows comprising traffic aggregated at either the pipe, the hose, or the VPN level. We assume that the measurements comprise samples gathered at (regularly spaced) instants during a window of duration T_{meas}. The samples are themselves some function of the traffic rates in the interval between sampling instants; in this paper we use average rate over the inter-sample interval. The measurements are used to predict an effective bandwidth for the traffic flow over some window of duration T_{ren} following the measurement window. Such predictions have the locality property that they depend only on measurements over the window of duration T_{meas} into the past. Suppose n rate samples $r_i,\ i = 1, \ldots, n$ are collected over the measurement window. Specific examples that we employ are:

B.0.c Local Maximum Predictor:. The renegotiated rate is the maximum of the rate sampled during the measurement window, i.e., $\max_{i=1}^{n} r_i$.

B.0.d Local Gaussian Predictor:. The renegotiated rate is equal to $\widehat{m} + \alpha\sqrt{\widehat{v}}$, where $\widehat{m} = n^{-1}\sum_{i=1}^{n} r_i$ and $\widehat{v} = (n - 1)^{-1}\sum_{i=1}^{n}(r_i - \widehat{m}^2)$ are respectively estimates of mean and variance formed from sampled rates, and α is a multiplier that controls the extent to which the negotiated rate accommodates variability in the samples. The interpretation of α is that in a Gaussian approximation to the rate distribution, we expect the bandwidth $\widehat{m} + \alpha\sqrt{\widehat{v}}$ to be exceeded with probability $1 - G(\alpha)$, where G is the cumulative distribution of the standard normal distribution. In the particular case of independent identically distributed rate samples, $\widehat{m}$ and $\widehat{v}$ are unbiased estimates of the true mean and variance m and v. The value of α should be chosen according to the SLA, $1 - G(\alpha)$ being the acceptable frequency with which the traffic is allowed to exceed allocated rate.

B.0.e . Both these predictors aim to take account of the statistical variability of the traffic. The local maximum does this in a relatively crude manner, being just the largest rate seen over a window. The local Gaussian predictor leverages certain expectations about the traffic, and is thus potentially more accurate than the local maximum predictor, or other predictors—e.g. auto-regressive or moving average—that utilize no model. A pipe is expected to carry aggregated traffic from many hosts; for this reason the Gaussian model should be a reasonable predictor over a short enough window, on account of the Central Limit Theorem, provided the tails of the distributions of marginal rates are not too heavy.

Below we will use fractional Brownian motion as a reference model for discussing robustness of the local Gaussian estimator. Following [9], the cumulative arrivals over a period of duration t are modelled as $A(t) = at + \sigma^2 B(t)$ where B is a fractional Brownian motion with some Hurst parameter H, and the mean rate a and variance parameters σ^2 are possibly local to the measurement period. Our own measurements on WAN traffic show that the marginal distributions of aggregate rates at time scales greater than about 1s can be well approximated by Gaussian distributions, at least up to 2 or 3 standard deviations from the mean. Moreover, the temporal behavior of rate *at these timescales* has been found to be consistent with the monofractal behavior of fBM. We assume that traffic rates are to be predicted at such timescales. In distinction, such models fail to describe the behavior observed at smaller timescales, i.e., the round trip time and shorter [10].

B.0.f Robustness to Systematic Variability.. Both predictors are robust with respect to systematic variability (i.e. non-stationarity) in the demand provided that T_{meas} is smaller than the timescale at which demand systematically varies. For example, it is well known that telephone traffic exhibits diurnal variation; the call arrival process can be accurately modeled as a time dependent Poisson process. On the other hand, the call arrival rate is relatively static over intervals of a few minutes; robustness to systematic variations requires choosing T_{meas} to be no larger than this timescale.

If the interval between renegotiations encompasses periods of systematic variation the local estimators above can become inaccurate. Remedies for this include:

(i) Use a worst case predictor over the largest time scale of variation, e.g., the maximum rate over a day for telephone traffic. This is robust but wasteful of resources.

(ii) Use historical data to predict trends, e.g., when average telephone call arrival rates are a known non-constant process $Q(t)$, then instead of using the predicted bandwidth $S(t)$ directly, we may use instead $S(t)Q(t + T_{\text{ren}})/Q(t)$ as are our predictor. Here the ratio $Q(t + T_{\text{ren}})/Q(t)$ is used to model the systematic change of the arrival rate upwards or downwards.

B.0.g Robustness to Statistical Estimation Error.. For the local Gaussian predictor, estimation of mean and variance is subject to inherent statistical error since the estimates are themselves random variables. This additional variability can lead to violation of target quality metrics if estimated parameters are assumed to be the true ones; see, e.g., [11]. In the local Gaussian model one can take account of the likely errors and correct for them in the predictor. The probability $\mathrm{Prob}[\mathrm{r} > \widehat{\mathrm{m}} + \alpha\sqrt{\widehat{\mathrm{v}}}]$ that a given independent rate r exceeds the local Gaussian predictor can be approximated as follows. Assume that $\widehat{m}$ and $\widehat{v}$ are formed from n measurements over some measurement window of independent Gaussian random variables with mean m and variance v. Then the pair $(\widehat{m}, \widehat{v})$ is approximately Gaussian with mean (m, v) and covariance matrix $n^{-1}((v, 0), (0, 2v^2))$; see Section 1.2 of [12]. Using the δ-method [13], it follows that $r - \widehat{m} - \alpha\sqrt{\widehat{v}}$ is approximately Gaussian with mean $-\alpha\sqrt{v}$ and variance $v(1 + n^{-1}(1 + \alpha^2/2))$. Thus the probability that r exceeds the predictor is actually approximately $1 - G(\alpha_0)$ where $\alpha_0^2 = \alpha^2/(1 + n^{-1}(1 + \alpha^2/2))$. Inverting this relationship, we see to meet a given target number of standard deviations α_0, one must select α such that

$$\alpha^2 = \alpha_0^2 \frac{1 + 1/n}{1 - \alpha_0^2/2n}$$

For example, with $\alpha_0 = 3$ and $n = 60$, then $\alpha = 3.15$.

B.0.h Prediction Across Timescales and Burstiness.. We assume that sampled rates r_i are formed as normalized differences $(A((i+1)t) - A(it))/t$ of counter values A across an interval of width t. We denote the generic such sample by $r(t)$, and call it a t-averaged rate. In order to satisfy a given SLA, it may be required to predict s-averaged rates from t-averaged samples for some $t > s$. In general, the variance of the sample $r(t)$ will dependent on the width t. In the simple case that A has independent increments, $\mathsf{Var}\,\mathsf{r(s)} = \mathsf{(s/t)Var}\,\mathsf{r(s)}$. Thus, for the local Gaussian predictor, the predicted s-averaged rate should be $\widehat{m}(t) + \alpha(t/s)\sqrt{\widehat{v}(t)}$, where $\widehat{m}(t)$ and $\widehat{v}(t)$ are the mean and variance estimated from t-averaged samples.

Now burstiness at multiple timescales has been observed in Internet data traffic [14], [15]. For the local Gaussian predictor, a priori knowledge of the scaling relations between rate variance at different time scales [14] can be used to accommodate short timescale variability. Suppose $A(t)$ is modelled using a fBM with Hurst parameter $H \in [1/2, 1)$. In this case $\mathsf{Var}\,\mathsf{A(t)}$ is proportional to t^{2H}. Hence to the local Gaussian predictor for s-average rates from t-averaged samples becomes $\widehat{m}(t) + \alpha\,(t/s)^{2-2H}\sqrt{\widehat{v}(t)}$.

B.0.i Robustness to Correlations. For the local Gaussian predictor, dependence between rate samples can bias the variance estimator $\widehat{v}$, although not the mean $\widehat{m}$. In particular, positive correlation between samples will, on average, lead to underestimation of the rate variance: the estimate $\widehat{v}$ has expected value $\mathsf{E}[\widehat{\mathsf{v}}] = \mathsf{v} - \kappa\mathsf{m}^2$ where $\kappa = \sum_{k=1}^{n-1}(n-k)c(k)$, and $c(k) = \mathsf{E}[\mathsf{r_{i+k}r_i}]/\mathsf{E}[\mathsf{r_i}]^2$ is the lag-k autocorrelation. Suppose for a given class of traffic and sampling method, κ can be calculated independently of the rate samples. Then robustness against sample correlations be achieved by using the modified variance estimator $\widehat{v}' = \widehat{v} - \kappa\widehat{m}^2$ in place of $\widehat{v}$. In the previous example of fBM, $\kappa = 2^{-1}\sum_{k=1}^{n-1}\left\{(k+1)^{2H} + (k-1)^{2H} - 2k^{2H}\right\}$.

B.0.j Composite Robustness. Any or all of the above approaches may be combined.

IV. Simulation Experiments

We use trace-driven simulation to examine the effectiveness of using hoses. For all the simulated experiments reported in this section, we used an approximation of the AT&T IP backbone topology in the continental U.S. (as of December 1998), comprising 12 core router centers. We use two sets of traces, one for voice traffic and one for data traffic.

.0.k Voice Traffic. We use the call detail records for telephone calls offered to the AT&T switched network in the domestic United States for the August-December 1998 time period. Each call detail record enumerates the originating, dialed and terminating number, along with the origination time and duration of a call. The origination time and duration are captured at the granularity of seconds. One may view the records as giving the potential load to be carried over an IP telephony VPN.

We use the following rules to determine the routing of each voice call. A given ten-digit telephone number is associated with an area code (based on the first three digits of the phone number). Each of the 168 area codes is assumed to funnel all its traffic through an access link to one of the 12 backbone nodes that is geographically closest to the centroid of the geographical region corresponding to an area code. Traffic between the backbone nodes follows the shortest path route in the network topology, following normal IP intra-domain routing.

To simulate the network traffic, we convert the call detail records into call counts on a minute by minute interval between every pair of area codes. These call counts are then used to compute the aggregate call statistics on every access and backbone link. Most of the results are derived from simulations corresponding to call counts over a 24 hour interval on Monday, November 9 1998. However, we also use call statistics gathered over the longer period (from August to December 1998) to motivate how admission control decisions could be made for this class of VPN traffic.

.0.l Data Traffic. For our data experiments we used NetFlow traffic records [16] gathered from a set of Cisco routers in a corporate network during a 12 hour period. NetFlow provides records at the level of traffic flows. A flow is a logical grouping of packets that share common properties and which are localized in time. The properties used for grouping may include source and destination IP addresses or subnets, and port numbers. Localization may be achieved by terminating flows through criteria such as timeouts (a packet is deemed to be the last in the flow if no additional packets with the same property are received within a subsequent timeout period) or the presence of protocol-specific information (e.g., TCP FIN flags). The range of IP addresses present in the NetFlow trace data was divided into 12 groups, each of which was assigned to one of the nodes in the physical topology. Each flow record contained the start and end time of the flow, together with the number of packets and bytes in the flow. Each such flow was mapped into a flow of constant rate transferring the same total number of bytes between its start and end times. Start and end times were given at the granularity of 1 second.

.0.m Experiments. For all the simulations using the data traffic, we assume that there are 12 VPN access points, one each for the 12 nodes in the example topology. For the voice simulations, we allow additional levels of traffic aggregation. At one extreme, each area code is considered as a separate VPN access port. In this case, the traffic generated by a hose corresponds to all of the calls originating from a particular area code (independent of the destination) resulting in a total of 168 hoses. Analogously, with a customer-pipe service interface, all the calls made from one area code to another would be a distinct pipe, resulting in a total of 168x168 customer-pipes. At the other extreme, all of the area codes funneling traffic into a particular backbone node are considered to constitute a single VPN access port. Thus, there are a total of 12 hoses and 144 customer-pipes. We also present results for two intermediate levels of aggregation with 24 and 48 hoses respectively. The larger scale aggregation is achieved with multiple hoses per VPN access point. For example, with 24 hoses, we assume that there are 2 hoses per VPN access point and with 48 hoses, there are 4 hoses per VPN access point, etc.

Using these two sets of data traces we conducted a number of experiments to evaluate the effectiveness of the hose model.

- We investigate the stability of VPN traffic matrices.
- We evaluate the usefulness of the hose model from the point of view of a customer, compared to the customer using a set of customer-pipes, and the benefit of resizing hoses.
- We compare two different mechanisms a service provider may use to realize a hose: a mesh of provider-pipes in the network vs. a source based tree. The provider can manage capacity in a couple of ways:

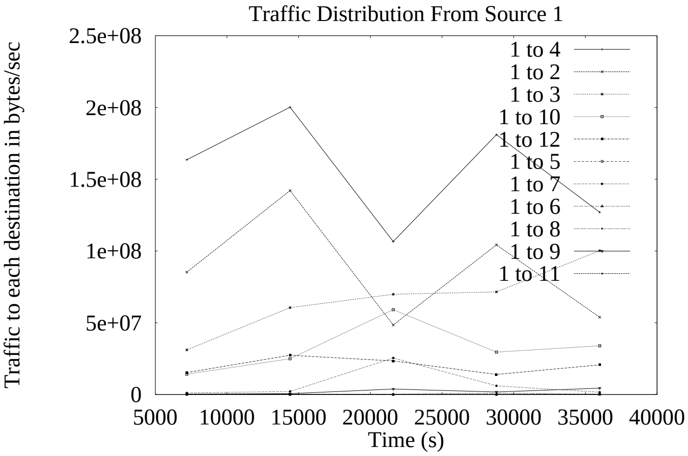

Fig. 4. VARIABILITY IN THE DATA TRAFFIC MATRIX: Capacity for each destination of a selected hose

– aggregating the required capacity across a hose or across an entire VPN.

– renegotiate the required capacity, either for the mesh of provider-pipes or for the source based tree.

We quantify the bandwidth savings from either one as well as using a combination of both techniques.

- Finally, we examine the relationship between short term capacity management by resizing and the longer term admission control algorithms within the context of the hose model.

V. HOSES FROM A CUSTOMER'S VIEWPOINT

One of the primary benefits of the hose model for the customer is the ability of the hose to accommodate changes in the traffic matrix. The use of hoses can also lead to more efficient use of access-link capacity relative to customer-pipes. We explore these issues experimentally in this section.

A. *Traffic Matrix Variability*

We examine the variability of the traffic matrix for data traffic in Figure 4, for a given source hose to all the hose destinations. Out of the 12 hoses in the configuration, we chose one that was representative of the high variability in the traffic matrix, observing the traffic matrix changing across five 2-hour intervals. We observe from the figure that there is considerable variability in the rate of traffic from the source to the different destinations. The proportion of traffic going from the source to the different destinations changes across the 2-hour intervals. The variability in the traffic matrix indicates the benefit to the customer from the hose interface. However, from the perspective of the network provider, this variability poses a challenge. Indeed, observe in Figure 4 that the most significant variability is for the source-destination flows that dominate the hose.

B. Performance Benefit of Hoses for the Customer

In this section we evaluate and compare the benefits to the customer of provisioning with hoses (as opposed to customer-pipes) on the access link, and the benefits of dynamic reprovisioning for either the hose or the customer-pipe model. To define our experiments more precisely, let $E_\ell(i)$ denote the set of destination endpoints for traffic sent from source endpoint i and which is routed through link ℓ. Let $r_{ij}(t)$ denote the time series of traffic from endpoint i to endpoint j. Let $S(r)$ denote the capacity required for traffic with a given time series r. With static provisioning, we calculate $S(r)$ as the maximal rate attained over the time series. With dynamic resizing, we calculate $S(r)$ as in Section III-B, using the Gaussian predictor. In this case, the required capacity is the time series of the predictor. On an access link carrying traffic sourced on hose i, we define the customer-pipe and hose requirements as follows:

- **Customer-Pipe Requirement** $= \sum_{j \in E_\ell(i)} S(r_{ij})$
- **Hose Requirement** $= S\left(\sum_{j \in E_\ell(i)} r_{ij}\right)$

We calculate the access hose-gain as the ratio of the customer-pipe requirement to the hose requirement. We refer to this ratio as the **statically provisioned access hose-gain** when the capacity for both the customer-pipe and the hose are determined based on their maximum rates. It is called the **dynamically resized access hose-gain** when the customer-pipes and hoses are dynamically resized, calculated as the ratio of their time averages.

C. Provisioning the Access Link

The capacity required by a VPN customer on each access link depends on the service model being offered to the customer. If the customer is presented the abstraction of a frame-relay style independent customer-pipes, adequate capacity would need to be provisioned independently on the access link for each such customer-pipe. A conservative approach would provision capacity per pipe to equal the maximum traffic demand for that pipe, over all time. The aggregate capacity needed on the access link would then be the sum of the maximum capacities needed for each of the customer-pipes on that link.

On the other hand, when the customer's service interface into the network is a hose, then the capacity needed on the access link is the maximum traffic demand for that hose, i.e., the maximum of the aggregate traffic transmitted from the source to all destinations.

Using the voice and data traffic sets described above, we performed an evaluation of the access link capacity required to be statically provisioned for each of these two service models and use this to compute the statically provisioned access hose-gain.

The results of this comparison for the data traffic is shown in Table I. The statically provisioned access hose-gain varies between 1.07 and 3.57 for this experiment. Examining the actual trace data showed that in the cases where the hose-gain was close to 1, there was a very large burst of traffic between the source and a particular destination which completely overshadowed the traffic contribution to other destinations. Such a burst would naturally dominate both the sum of customer-pipe capacities and the hose capacity, leading to a small gain.

Hose	Static Requirement (kB/sec)		Static prov.
Source	Customer-Pipe	Hose	hose gain
1	2229	1164	1.92
2	2873	1379	2.08
3	13379	12538	1.07
4	4925	2031	2.43
5	619	255	2.42
6	86	79	1.08
7	112	59	1.88
8	3104	2538	1.22
9	1483	416	3.57
10	752	251	2.99
11	778	303	2.56
12	1606	771	2.08

TABLE I
STATICALLY PROVISIONED ACCESS HOSE GAIN FOR DATA TRAFFIC: STATIC REQUIREMENTS FOR CUSTOMER-PIPES AND HOSES.

A similar comparison was done for telephony traffic. In this case the statically provisioned access hose-gain varies between 1.22 and 16.39, with a mean of 1.78.

D. Resizing the Access Link

When there are significant fluctuations over time in the offered traffic, it is useful to provide customers with the capability to renegotiate hose capacities. This renegotiation may be based on demand predictions derived from measurements that track the fluctuations in the offered traffic. Erroneous predictions would result in either wastage of network capacity or the inability to accept all of the offered traffic (resulting in blocked calls for the telephony service and packet losses for the data service). The next set of experiments evaluate the utility of renegotiating hose capacity and quantify how well the predictors described earlier deal with the deterministic and statistical variations in the traffic.

To quantify the utility of renegotiation, we compute the **hose resizing gain**: this is the ratio of the statically provisioned hose requirement (i.e. the maximum offered traffic over the length of the experiment) to the time-average of the renegotiated hose requirement.

	number of hoses			
	12	24	48	168
Max. resizing gain	2.15	2.17	2.22	3.08
Mean resizing gain	1.94	1.95	1.98	2.06
Min. resizing gain	1.80	1.79	1.77	1.82

TABLE II
HOSE RESIZING GAIN FOR VOICE TRAFFIC: MAXIMUM, MEAN AND MINIMUM ACROSS HOSES AT DIFFERENT AGGREGATION LEVELS. RENEGOTIATION AT 1 MINUTE INTERVALS .

Resize Freq.	12 hoses	24 hoses	48 hoses	168 hoses
1 minute	2.34, 0.40	2.37, 0.39	2.53, 0.39	3.55, 0.47
5 minute	2.72,0.88	2.73, 0.86	2.88, 0.86	3.96, 0.98
10 minute	3.11,1.42	3.11, 1.38	3.25, 1.37	4.36, 1.49
30 minutes	4.72,3.54	4.64, 3.40	4.72, 3.36	5.76,3.48

TABLE III
BLOCKING ABOVE HOSE REQUIREMENT FOR VOICE TRAFFIC: IMPACT OF RESIZING INTERVAL ON PERFORMANCE. EACH TABLE ENTRY REPRESENTS THE PERCENTAGE OVERALLOCATION AND PERCENTAGE OF CALLS BLOCKED.

D.1 Benefit of Resizing the Access Link for Voice Traffic

The next set of experiments explores the utility of hose resizing for the telephony traffic. In these experiments, we use the variance based predictors to compute the capacity required for hoses at each of the 4 different levels of aggregation (12, 24, 48, 168). We use a moving window of 10 minutes for the measurement and traffic predictions and a renegotiation interval of 1 minute.

Table II shows the minimum, maximum and average values for the hose resizing gain, computed across all hoses, over a 24 hour interval, at each of the 4 different levels of aggregation. The mean resizing gain of nearly 2 indicates that there is a significant benefit to be derived by resizing for telephony traffic, even when there is considerable aggregation.

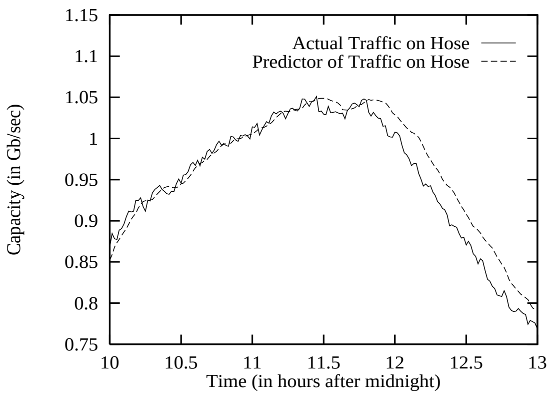

Fig. 5. TIME SERIES OF VOICE TRAFFIC: actual traffic and hose prediction for a single hose.

The dynamic behavior of the variance based traffic predictor

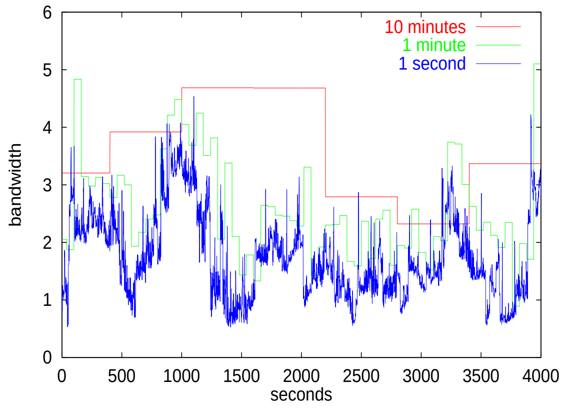

Fig. 6. HOSE PREDICTORS FOR DATA TRAFFIC: variance-based predictors for data over 1 second, 1 minute and 10 minute windows. Bandwidth is shown in nominal units.

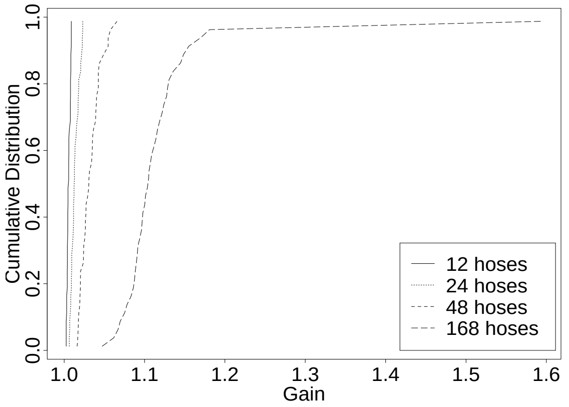

Fig. 7. DYNAMICALLY-RESIZED ACCESS HOSE GAIN FOR VOICE TRAFFIC. CDF (across access links) of the gain in capacity in going from dynamically resized customer-pipes to dynamically resized hoses.

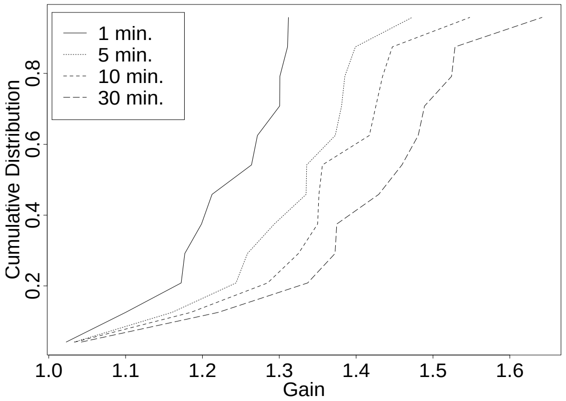

Fig. 8. DYNAMICALLY-RESIZED ACCESS HOSE GAIN FOR DATA TRAFFIC. CDF (across access links) of the gain in capacity in going from dynamically resized customer-pipes to dynamically resized hoses.

is illustrated in Figure 5. This is a plot of the time series over the interval [10 am - 1 pm] (busy hour) for the aggregated traffic and the corresponding capacity predictor for a hose. We assume an aggregation level of 168, i.e. each area code constitutes a VPN hose endpoint; this particular hose trace corresponded to telephone traffic originating from an area code in the New York area.

The figure illustrates that the predictor tracks the actual capacity requirements quite closely. A measure of effectiveness of the predictor is the overallocation of capacity, which is 1.78% for this trace. Another measure, which is blocking of offered calls, is nearly 0.3% over a 24 hour interval for this trace.

The results in Table II were for a resizing interval of 1 minute. Table III illustrates the effect of reducing the resizing frequency on the performance of the predictor as measured by the values of the tuple {%overallocation, %blocking}. The results show that the blocking probabilities become unacceptably high when the resizing interval increases beyond about 5 minutes. However, the amount of overallocation remains reasonable for such resizing intervals.

D.2 Benefit of Resizing the Access Link for Data Traffic

In Table IV we display the hose resizing gains for data traffic on access links. The more frequently the hose is resized, the higher the gain. The mean gain improves from around 9 to 15 as the resizing interval changes from 30 minutes down to 1 minute.

The behavior of the variance based predictor used in these experiments is illustrated in Figure 6. We plot a sample path of the aggregate traffic over a hose (at one second granularity) together with variance-based predictors with 1 and 10 minutes windows. Observe how the 1 minute predictor follows the trace more closely that the 10 minute predictor. The variance-based predictor is responsive to both upward and downward trends in the trace. This is an advantage for prediction during upward trends; on the other hand it can lead to periods of low utilization after downward trends; see e.g., the central portion of the 10 minute predictor in Figure 6.

The data traffic exhibits far greater short-term variability than the voice traffic.

E. Comparison of Benefits of Resized Hoses and Customer-Pipes

In the next set of experiments, we compare the access link capacity required with resized hoses with that required by resized customer pipes. As before, we express the ratio of these capacities as a "dynamically resized" access hose-gain. Figure 7 provides results for the voice traffic comparing the access link capacity required using resized hoses instead of resized customer-pipes. The average access-hose gain varies between 10% and 20% depending on the degree of aggregation, with the greater benefits at lower levels of aggregation (observe the case of 168 hose endpoints). The comparison is based on commensurate renegotiation intervals for both alternatives. Thus, on the access link, there is a benefit from the natural aggregation that the hose provides.

Resizing Interval	1 min.	5 min.	10 min.	30 min
Max. resizing gain	62.9	49.8	44.4	37.8
Mean resizing gain	15.5	11.9	10.7	9.23
Min. resizing gain	1.95	1.84	1.80	1.75

TABLE IV

Hose Resizing Gain for Data Traffic: maximum, mean and minimum across hoses for different resizing intervals T_{ren}.

Next, consider the data traffic. The CDF for the dynamically resized access-hose gain is shown in Figure 8 for renegotiation windows $T_{\text{ren}} = T_{\text{meas}}$ of 1,5,10 and 30 minutes. The gain is less pronounced for smaller time windows, ranging on average from about 1.4 for the 30 minute window down to about 1.2 for the 1 minute window. The predictor follows the trace more closely for smaller windows; hence the pipe predictor becomes closer to the hose predictor.

VI. Performance Benefit of Hose Realizations for a Provider

For the topologies and voice and data traffic described in Section IV, we evaluate the benefits for a provider of the three implementations of the hose model described in Section III: a set of provider-pipes, trees, and aggregated VPNs. For all of these alternatives, we assume default (shortest path) routing.

Using the same notation as in Section V we define:

- **Provider-pipe Requirement** $= \sum_i \sum_{j \in E_\ell(i)} S(r_{ij})$
- **Tree Requirement** $= \sum_i S\left(\sum_{j \in E_\ell(i)} r_{ij}\right)$
- **VPN Requirement** $= S\left(\sum_i \sum_{j \in E_\ell(i)} r_{ij}\right)$

We evaluate the **tree gain** for a link ℓ as the ratio of the provider-pipe requirement to the Tree requirement. For voice traffic we perform this evaluation at different levels of hose aggregation; see Section IV. For data traffic we evaluate the requirements for two levels of aggregation. First, we calculate the tree requirement for trees based on hoses from 12 VPN endpoints, one accessing the network at each of the 12 nodes of the network topology. Second, we calculate the VPN requirement: that of the aggregate of all provider-pipes routed over link ℓ; the **VPN gain** is the ratio of the provider-pipe requirement to the VPN requirement. We calculate these gains under both static provisioning and dynamic resizing.

We summarize our results for this section below. More details

Resizing Freq.	1 min.	5 min.	10 min.	30 min.
max	1291	592	422	272
mean	48.9	35.2	30.6	25.0
min	2.11	1.91	1.80	1.69

TABLE V
PROVIDER-PIPE RESIZING GAIN FOR DATA TRAFFIC: MAXIMUM, MEAN, AND MINIMUM OF STATIC TO DYNAMICALLY RESIZED PROVIDER-PIPE. THE MEAN IS CALCULATED BY WEIGHTING EACH PROVIDER-PIPE GAIN BY ITS STATIC PROVIDER-PIPE CAPACITY REQUIREMENT.

can be found in [17].

Consider the requirement for a tree-based implementation of a single hose. Moving from the root of a tree corresponding to a given hose towards a leaf, progressively fewer flows are aggregated together and hence we expect the benefit of sharing reservations in the tree to decrease. For both data and voice traffic the tree gain varies between 1 and approximately 2.5.

A. Benefits of Dynamical Resizing

A.0.a Voice Traffic.

Dynamically resized provider-pipes: We examine the resizing gain for voice traffic. By resizing gain, we mean the ratio of the peak capacity of the static provider-pipe to the peak capacity of the dynamically resized provide pipe. When there is a medium level of aggregation (where the 168 area codes are aggregated to only 48 distinct hose endpoints), there is a reasonable benefit from resizing. The average gain for an internal link capacity is about 1.27, with a minimum of 1.08 and a maximum of 1.71, using a resizing interval of 1 minute. However, as we increase the level of aggregation and only have 12 distinct hose endpoints, the gain due to resizing becomes much smaller. The mean gain is only about 7%.

With dynamically resized trees, the additional gain provided by the tree implementation is between 1% and 17%.

A.0.b Data Traffic.

Dynamically Resized Provider-Pipes: In Table V we summarize the statistics of provider-pipe gains when resizing; i.e. the ratio of maximum data rate on the provider-pipe to the time averaged resized rate, for renegotiation windows $T_{\text{ren}} = T_{\text{meas}}$ of 1,5,10 and 30 minutes. Some extreme gains are evident due to localized peaks in the data rate. Mainly these extremes occurred for a provider-pipe with low maximum rate and so did not dominate the weighted mean in Table V.

Dynamically Resized Trees:

Similar to the results for voice traffic the gains with resizing trees are nominal. This is because the average number of distinct endpoints for a tree on a link was 1.85. Aggregation has little impact especially as we near the destination.

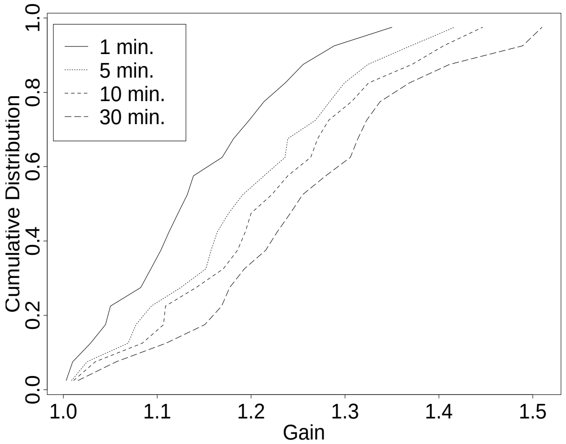

Fig. 9. DYNAMICALLY RESIZED VPN GAIN FOR DATA TRAFFIC: CDF (across all network links) of ratio of resized VPN requirement to resized provider-pipe requirement, according to resizing interval.

Dynamically Resized VPNs: Aggregating *all* flows present from a VPN on a given link potentially gives more multiplexing gain. We believe that aggregating per VPN is feasible in that hoses within a VPN are likely to be "aggregatable" in that they will be statistically similar and to have the same QoS requirements. This motivates sharing reservations by VPN's. The CDF (over network links) of the VPN gain with dynamic resizing is shown in Figure 9.

VII. EFFECTIVE BANDWIDTHS FOR ADMISSION CONTROL

For admission control for hoses of a fixed capacity it suffices to perform a test at each network link over which a hose's traffic is routed. The test verifies that the total capacity allocated for active hoses on that link plus the capacity required for the new hose does not exceed the available bandwidth.

However, when hoses can be resized, the computation of the total capacity allocated to active hoses at a link, cannot be done based on the currently requested capacity for each of these hoses. This is because the decision to admit a hose impact resource usage for the lifetime of the hoses, not just in the short term. Instead, it is desirable to use a more stringent measure of the bandwidth requirement of each active hose - we will refer to this as an effective bandwidth. Here we propose that the effective bandwidth for an admitted hose be the maximum over its lifetime of its shorter term bandwidth requirements. Candidates for algorithms to characterize the latter include algorithms proposed for bandwidth estimation for statistical QoS; see e.g. [18] for a review. A natural choice available in the present framework is the predicted bandwidth used for renegotiation. For the local maximum predictor, the effective bandwidth reduces to the

	Access Links		Internal Links		
Resize Freq.	Pipe	Hose	Pipe	Hose	VPN
1 minute	1.4e-5	5.9e-6	1.6e-5	1.3e-5	1.0e-5
5 minute	1.1e-4	3.9e-5	1.0e-4	9.1e-5	4.1e-5
10 minute	2.1e-4	6.3e-5	1.9e-4	1.6e-4	9.9e-5
30 minutes	5.8e-3	1.7e-4	5.4e-4	4.1e-4	3.5e-4

TABLE VI

EXCURSIONS ABOVE EFFECTIVE BANDWIDTH FOR DATA TRAFFIC: IMPACT OF RESIZING INTERVAL ON THE PROPORTION OF BYTE UNDERALLOCATION OF REQUIREMENT, ACCORDING TO LEVEL OF AGGREGATION

maximum traffic rate seen at the sampling timescale.

The experiments reported in this section used as the effective bandwidth the maximum over the hose lifetime of the local Gaussian predictor. We compare the properties of the effective bandwidths for the different levels of aggregation: pipe, hose and VPN. The CDF of the ratio of effective bandwidths for customer-pipes vs. hoses on access links in Figure 10, for provider-pipes versus hoses on internal links in Figure 11 and for provider-pipes vs. VPN in Figure 13. Observe greater gain for link than for hose aggregation; however, the gain is relatively insensitive to the size of the prediction window. Table VI shows that the proportion of bytes above the effective bandwidth is exceedingly small, and decreases with the degree of aggregation.

The Gaussian predictor admits some tuning for sensitivity through the parameter α that determines the target number of standard deviations above the mean at which the predictor should lie. Tuning this parameter downwards enables the provider to overbook resources.

A. Comparison of Dynamic Pipe Resizing with Overbooking

Current networks offering a VPN service typically manage capacity by reserving capacity for a pipe between sites. Service providers use a form of resource overbooking in order to minimize over-allocation of resources based on peak usage. Admission control is based on worst-case provisioning. This entails allocating the maximum capacity required on each link of a VPN pipe. One can think of this as performing CBR allocation based on the peak rate.

With overbooking, one may be able to achieve statistical multiplexing gains compared with peak-rate based allocation. The resource is considered to have a larger capacity (e.g., a factor of 2 to 6 greater) than the physical resource, and this larger capacity is used for admission control purposes. However, when the capacity of an interior link is exceeded, no information on this event is available at the edge router. This limits the actions that can be performed at the edge router in response. In contrast, with the capacity resizing at the edge, more information is available concerning the lack of capacity on the internal links of the pipe. For example, if blocking of voice calls was an option, the service provider may choose to block calls when there is insufficient capacity to satisfy the renegotiation request.

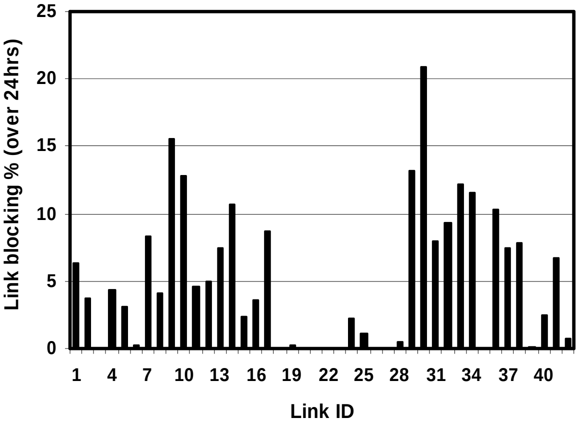

Fig. 12. BLOCKING DUE TO OVERBOOKING OF VOICE TRAFFIC: Blocking probability over 24 hour period on each of 42 links in the network.

As a way of quantitatively estimating the benefits of resizing the pipe in comparison to overbooking, we look at the blocking probability of voice calls with our traces with overbooking and compare it to that obtained with the capacity resizing mechanism, for the same capacity of the links involved in a pipe.

Table III illustrates the blocking probabilities with varying resizing intervals, ranging from 1 minute to 30 minutes. For example, with 168 hoses, the blocking probability is of the order of 0.98 when the resizing interval is 5 minutes. Also relevant is the variability in the resizing gain we observe with dynamic resizing. This suggests that the amount of capacity required on each of the provider-pipes varies on a link by link basis; and hence applying a uniform overbooking factor to all links is unlikely to provide acceptable performance.

Figure 12 shows the blocking probability at each link in the network carrying 168 customer pipes. Overbooking is modeled by reducing each link capacity to 40% of its original capacity; this emulates overbooking by a factor of 2.5. The blocking probability is measured at each link over a 24 hour period. For this fixed overbooking level, the blocking probability on each of the 42 links in the network varies considerably. For example, at several of the links (3, 18, links 20 through 23, etc.) there is no blocking at all. However, there are other links where the blocking probability is unacceptably high (nearly 21% at link 30). This confirms that applying a fixed overbooking factor uniformly across all links in the network is unlikely to provide acceptable performance. Furthermore, it is unlikely that the use of traffic models, measurement, and/or historical trending will provide sufficient information to enable the provider to apply

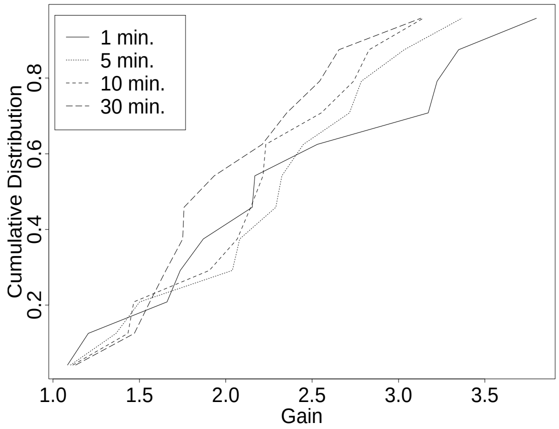

Fig. 10. HOSE GAIN FOR EFFECTIVE BANDWIDTH ON ACCESS LINKS FOR DATA TRAFFIC: CDF (over access links) of ratio of maximum pipe requirement to maximum hose requirement, according to renegotiation interval.

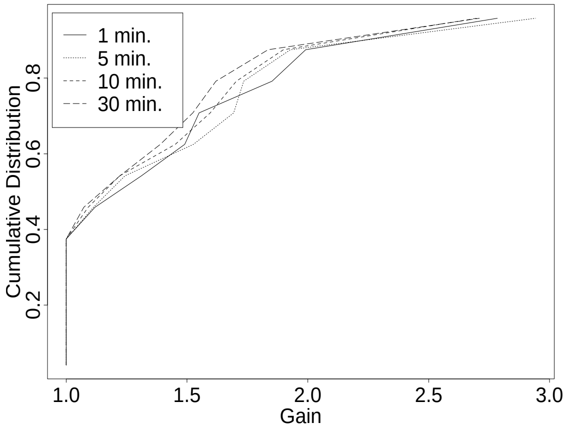

Fig. 11. HOSE GAIN FOR EFFECTIVE BANDWIDTH ON INTERNAL LINKS FOR DATA TRAFFIC: CDF (over internal links) of ratio of maximum pipe requirement to maximum hose requirement, according to renegotiation interval.

individual link overbooking factors on a daily basis. In contrast, dynamically resizing the provider pipe adapts to the dynamic capacity requirements on the internal links on the timescale of minutes.

VIII. CONCLUSIONS AND FUTURE WORK

VPNs are undergoing dramatic change owing to at least three interrelated factors:

- rapid progress in IP network technologies (in overall capacity and the development of diverse network access technologies);
- progress in IP security (in flexible, dynamic methods for establishing security associations);
- rapid change in the diversity and dynamics of communication and collaboration patterns at work and at home.

As a result, communication patterns are evolving from fairly static, predictable flows between endpoint *pairs* to dynamic, difficult to forecast traffic between *sets* of endpoints.

Accordingly, we have proposed a more powerful, easier to specify service model for the customer, termed a **hose**. A hose is characterized by the aggregate traffic to and from one endpoint in the VPN to the set of other endpoints in the VPN, and by an associated performance guarantee. A hose allows a customer to simply buy a logical *access* link and use it to send traffic to any one of the remote hose endpoints, with reliable QoS, and with the rates of the customer access links the only limitation. In addition, hoses naturally allow the customer to take advantage of aggregation of the flows to and from access links, reducing required access link capacities. Though it would appear that hoses present greater resource management challenges for the provider, these difficulties can be addressed by statistical multiplexing or resizing techniques, applied separately or in combination.

Using trace-driven simulations, we examined the effectiveness of the hose interface and the benefit of resizing. We looked at traces of telephone calls over the AT&T national long distance network, as well as traces of data traffic on a large corporate private network. Our simulations showed significant capacity savings by using techniques to improve statistical multiplexing and resizing:

- On access links, we find there is considerable potential for statistical multiplexing. It achieves a factor of 2 to 3 in capacity savings over statically provisioned customer-pipes. On network internal links, we found that statistical multiplexing, by hose, typically provides small benefit because the number of distinct destinations reachable on the link for a given hose is small. We suspect that with a richer network topology, especially with less aggregation in each customer-pipe, the provider may achieve greater benefits even on network internal links. On the other hand, statistical multiplexing by VPN, provides significant benefit.
- Similarly, resizing provides about a factor of 2 in savings in access link capacity when resizing the hose just once a minute. A simple predictor based on a local Gaussian approximation was used to calculate the expected load. The gains are much higher with data traffic because of the higher variability. On network internal links, resizing is similarly effective.
- Combining statistical multiplexing and resizing, provides additional benefit, over applying either separately.

We believe the VPN service model presented here will naturally and economically fit emerging business practices in an increasingly IP networked environment. In this paper, we have just begun to address VPN performance issues within the IP con-

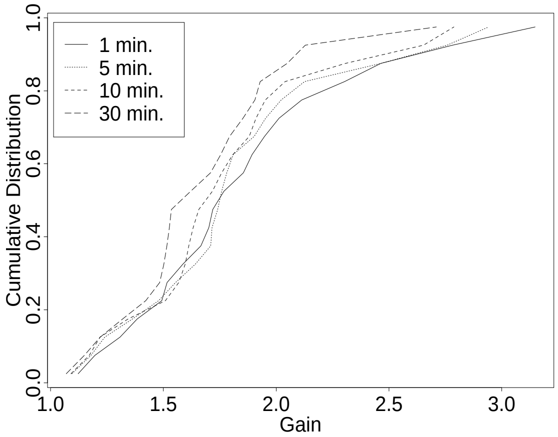

Fig. 13. VPN GAIN FOR EFFECTIVE BANDWIDTH ON INTERNAL LINKS FOR DATA TRAFFIC: CDF (over internal links) of ratio of maximum pipe requirement to maximum VPN requirement, according to renegotiation interval.

text. It is very important to understand the detailed technical specifications of the QoS attributes for hoses, i.e., the assurances in the SLA related to delay, loss, and jitter. In general, we expect these specifications to be looser for hoses than for customer-pipes. Thus, we expect both paradigms will play important roles.

REFERENCES

[1] S. Fotedar, M. Gerla, P. Crocetti, and L. Fratta, “ATM Virtual Private Networks,” *Communications of the ACM*, vol. 38, pp. 101–109, Feb 1995.

[2] S. Rooney, J. E. van der Merwe, S. Crosby, and I. Leslie, “The Tempest, a Framework for Safe, Resource Assured, Programmable Networks,” *IEEE Communications Magazine*, vol. 36, pp. 42–53, October 1998.

[3] E. Rosen and Y. Rekhter, “Bgp/mpls vpns.” RFC 2547, available as http://www.ietf.org/rfc/rfc2547.txt, March 1999.

[4] S. Kent and R. Atkinson, “Security Architecture for the Internet Protocol.” RFC 2401, November 1998.

[5] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, J. van der Merwe, N. Doraswamy, and S.Jagannath, “A performance oriented service interface for virtual private networks.” Unpublished. Available as http://www.ietf.org/proceedings/98dec/I-D/draft-duffield-vpn-qos-framework-00.txt.

[6] B. Davie and Y. Rekhter, *MPLS: Technology and Applications*. Morgan Kaufmann, 2000.

[7] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener, “Algorithms for provisioning virtual private networks in the hose model,” in *ACM SIGCOMM*, 2001.

[8] G. Italiano, R. Rastogi, and B. Yener, “Restoration algorithms for virtual private networks in the hose model,” in *IEEE INFOCOM*, 2002.

[9] I. Norros, “A storage model with self-similar input,” 1994.

[10] A. Feldmann, A. C. Gilbert, W. Willinger, and T. Kurtz, “Looking Behind And Beyond Self-similarity: On Scaling Phenomena in Measured WAN Traffic,” in *Proceedings of the 35th Allterton Conference on Communication, Control and Computing*, 1997.

[11] M. Grossglauser and D. N. Tse, “A Framework for Robust Measurement-Based Admission Control,” in *Proceedings ACM SIGCOMM 97*, (Cannes, France), 1997.

[12] R. Muirhead, *Aspects of Multivariate Statistical Theory*. Wiley, 1982.

[13] M. Schervish, *Theory of Statistics*. New York: Springer, 1995.

[14] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the Self-Similar Nature of Ethernet Traffic (Extended Version),” *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, 1994.

[15] V. Paxson and S. Floyd, “Wide-Area Traffic: The Failure of Poisson Modeling,” *IEEE/ACM Transactions on Networking*, vol. 3, no. 1, pp. 226–244, 1995.

[16] Cisco NetFlow. For more information see `http://www.cisco.com/warp/public/732/netflow/`.

[17] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, and J. E. van der Merwe”, “A Flexible Model for Resource Management in Virtual Private Networks,” *SIGCOMM’99 - Computer Communication Review*, vol. 29, pp. 95–108, October 1999.

[18] E. Knightly and N. Shroff, “Admission Control for Statistical QoS: Theory and Practice,” *IEEE Network*, vol. 13, no. 2, pp. 20–29, 1999.

Nick Duffield received a B.A. in natural sciences in 1982, and the Certificate of Advanced Study in Mathematics in 1983, both from Cambridge University, UK, and a Ph.D. in mathematical physics from the University of London, UK, in 1987. He subsequently held post-doctoral and faculty positions in Heidelberg, Germany and Dublin, Ireland. He is currently a Technology Leader in the Internet and Networking Research group at AT&T Labs–Research in Florham Park, NJ, USA. His current research focuses on Internet performance measurement, inference, and analysis.

Pawan Goyal received the Ph.D degree in computer sciences from the University of Texas at Austin in 1997 and B.Tech in computer sciences from Indian Institute of Technology at Kanpur in 1992. He is currently a research staff member in the storage systems department at IBM Almaden Research Center. He was previously at Ensim Corporation and AT&T Labs Research. His research interests are in storage systems, computer networking, operating systems and multimedia systems.

Albert Greenberg received the B.A. degree in mathematics from Dartmouth College in 1978, and the M.S. and Ph.D. degrees in computer science from the University of Washington in Seattle in 1981 and 1983, respectively. He joined AT&T Bell Labs Mathematics Research Center in 1983, and became a Dept. Head in the Network Services Research Center in 1995. In 1996, he moved to AT&T Labs-Research, where he heads the IP Network Management and Performance Dept., a group which conducts research in networking, with an emphasis on large scale IP networks and systems, and emerging Internet technologies. His research interests include Internet traffic measurement, modeling and engineering, policy based networking, and optical networking (IP/WDM). In collaboration with with several others in AT&T Labs, he is developing a unified toolkit to manage IP networks.

Partho Mishra received a B Tech in Computer Science from Indian Institute of Technology, Kharagpur, in 1988 and a PhD in Computer Science from the University of Maryland, College Park in 1993. He was at AT&T Bell Labs from 1993-1996, AT&T Labs from 1996-1999 and Gigabit Wireless from 1999-2001. He is currently Director, Embedded Software at Woodside Networks. His technical interests include Wireless Networking, Traffic Engineering, Quality-of-Service and Networking Software.

K.K. Ramakrishnan is a Technology Leader at AT&T Labs Research. He received the BS in Electrical Engineering from Bangalore University in India in 1976, the M.S. degree in Automation from the Indian Institute of Science in 1978 and the Ph.D. in Computer Science from the University of Maryland, College Park, in 1983. From 1983 to 1994, he was with Digital Equipment Corporation. Between 2000 and 2002 he was a founder and Vice President at TeraOptic Networks, Inc. His research interests are in design and performance of computer and communication network protocols and algorithms. Dr. Ramakrishnan has been an editor for the IEEE/ACM Transactions on Networking and IEEE Network Magazine.

Jacobus E. Van der Merwe is a Principal Technical Staff Member at AT&T Labs-Research in Florham Park, New Jersey. He currently works on content distribution networking and multimedia streaming. He received bachelor and masters degrees in electronic engineering from the University of Pretoria in South Africa, and a PhD in computer science from the University of Cambridge in England. He is a member of the IEEE.