MON: Mission-optimized Overlay Networks

B. Spang, A. Sabnis, R. Sitaraman, D. Towsley
College of Information & Computer Sciences
U. Massachusetts - Amherst
Amherst, MA 01003
{bspang,asabnis,ramesh,towsley}@cs.umass.edu

B. DeCleene
BAE Systems & Technology Solutions
Burlington, MA 01803
brian.decleene@baesystems.com

***Abstract*—Large organizations often have users in multiple sites which are connected over the Internet. Since resources are limited, communication between these sites needs to be carefully orchestrated for the most benefit to the organization. We present a Mission-optimized Overlay Network (MON), a hybrid overlay network architecture for maximizing utility to the organization. We combine an offline and an online system to solve non-concave utility maximization problems. The offline tier, the Predictive Flow Optimizer (PFO), creates plans for routing traffic using a model of network conditions. The online tier, MONtra, is aware of the precise local network conditions and is able to react quickly to problems within the network. Either tier alone is insufficient. The PFO may take too long to react to network changes. MONtra only has local information and cannot optimize non-concave mission utilities. However, by combining the two systems, MON is robust and achieves near-optimal utility under a wide range of network conditions. While best-effort overlay networks are well studied, our work is the first to design overlays that are optimized for mission utility.**

I. Introduction

Large organizations have users in multiple sites that are connected over the Internet. A business may have multiple offices around the world which need to communicate with each other. A defense organization may have personnel deployed at multiple sites, who need to communicate and fulfill specific mission goals. A retailer may have multiple shops, warehouse locations, and offices. One traditional approach to facilitating communication between distributed sites of an organization is to deploy a *private* enterprise network with dedicated infrastructure to fulfill the organization's communication requirements. However, an alternate approach is to build an *overlay* on top of the *public* Internet, avoiding the need for dedicated infrastructure.

Overlays have been studied and built for the past 25 years [1], [9], [24], [26], [29]. Large CDNs such as Akamai [21] have built overlays for delivering web and video content since the late 1990's [25]. These overlays are "best-effort", in that they are concerned with providing higher reliability and performance than what the native Internet can offer for all traffic using the overlay. Such best-effort overlays include caching overlays for Web content [7], routing overlays for reliably transporting live video streams [3], [17], P2P overlays for downloads [24], [26], [29], and security overlays for preventing DDoS attacks [25]. However, best-effort overlays do not explicitly optimize the "mission goals" of the organization that operates the overlay.

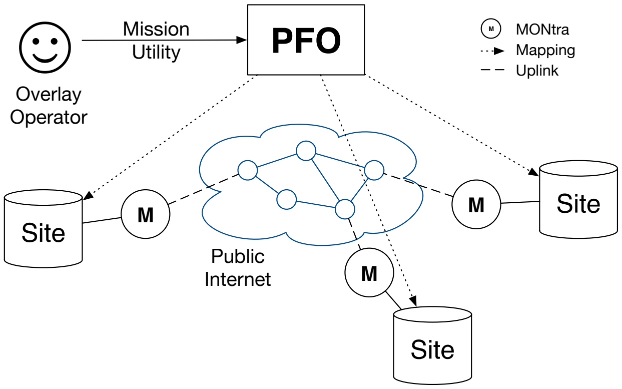


Fig. 1. MON Architecture Diagram

In this paper, in contrast to best-effort overlays studied in prior work, we propose and study overlays that are driven by explicitly stated mission goals. For instance, consider a multi-site defense organization. The goals for the overlay are set by an operator who dictates the relative utility of various types of communication that occur between the different sites. Note that the mission goals may vary with time, e.g., an urgent all-hands video conference watched by users in all the sites may take higher precedence than downloads, VOIP and other traffic classes. A Mission-optimized Overlay Network (MON) dynamically allocates available overlay resources to the traffic between sites to maximize overall mission utility, enabling the goals of the organization to be met.

A. MON Functionality

MON takes as input the (time-varying) mission goals set by the operator and routes traffic on the overlay network to meet these goals (see Figure 1). Each site is connected to the public Internet through a transport controller which performs overlay routing.

We group the traffic between sites into a set of classes K, where each traffic class represents a set of end-user sessions of a specific type (such as video, downloads, VOIP, etc...) between a specific source and destination site. Each traffic class may use a set of overlay routes. The mission goals are captured by mission utility functions specified by the overlay operator. MON determines a set of sessions from each class and a rate for each session so as to maximize the cumulative mission utility.

MON is designed to continually adapt to change. The mission utility functions can change as mission goals change. The number of sessions in each class that need to be routed can change with user demand. The underlying Internet could suffer from failures that require traffic to be rerouted. To deal with these changes, MON continually adapts the number of sessions n_k and rates x_k for each class $k \in K$ to optimize the mission utility.

B. Our Contributions

We propose a novel two-tiered overlay architecture for MON that combines offline and online tiers. The offline tier is called the Predictive Flow Optimizer (PFO), and it periodically performs a global optimization of the cumulative mission utility. The output of PFO is "mapped" to a lower-level online network transport mechanism called MONtra which performs the actual routing of traffic in the network using proportionally-fair utility functions (see Figure 1). We prove that MON's two-tiered architecture converges to an optimal cumulative mission utility. An interesting aspect of our work is a mapping process that allows us to implement arbitrary non-decreasing mission utility functions using logarithmic transport utilities which are well-studied and have desirable properties such as proportional fairness.

To establish the real-world feasibility of MON, we implement a prototype within the Deterlab [19] testbed. We show that the PFO implementation using a bilinear global optimizer, in combination with MONtra implemented on the Deterlab nodes, is able to send traffic at rates that converge to a solution that achieves optimal mission utility. We also show that the system is robust to changes in the network, such as those caused by network partitions or congestion. Further, we show that the system is robust to changes in the number of sessions, such as those caused by flash crowd events. We also empirically evaluate MON when the network and traffic demands are not precisely known. In this case, we show that MON degrades gracefully and still provides a near-optimal mission utility.

C. Roadmap

We give an overview of the MON architecture and a detailed description of each component in Section II. We describe our prototype and experimental setup in detail in Section III. We implement MON and present our empirical results in Section IV. We compare MON to prior work in Section V and then conclude in Section VI.

II. The MON Architecture

MON dynamically allocates available overlay resources to traffic between sites to maximize overall mission utility. In order to do this, we need a precise definition of the mission utility maximization problem. We group the traffic between sites into a set of classes K, where each traffic class $k \in K$ represents a set of sessions of a specific type (e.g. video or VOIP) between a specific source and destination site. Each class uses a set of overlay routes ρ_k. Mission goals are captured by mission utility functions specified by the overlay operator. Specifically, associated with each class $k \in K$ is a function $U_k(x_k)$ that corresponds to the value of one session of class k receiving a rate of x_k. MON chooses a number of sessions n_k for each class $k \in K$ and routes each chosen session of class k at a rate of x_k so as to maximize the cumulative mission utility expressed as $\sum_k n_k U_k(x_k)$.

Symbol	**Meaning**
K	Set of traffic classes
N_k	Maximum number of sessions for class k
ρ_k	Set of routes usable by k
L	Set of underlay links
$\hat{L}$	PFO's estimate of L
C	Set of underlay link capacities
$\hat{C}$	PFO's estimate of C
$U_k(x_k)$	Per-class mission utility
n_k	Number of admitted sessions for class k
x_f	Rate assigned to a flow
x_k	Aggregate class rate ($x_k = \sum_{f \in \rho_k} x_f$)
w_f	MONtra's weight for class k on flow f
$V_f(x_f)$	MONtra's utility for class k on flow f
γ	MONtra's stability constant

Fig. 2. Table of Notation

MON has a two-tiered architecture which combines a non-real-time global optimizer (PFO) with a distributed real-time transport protocol (MONtra) to optimize mission utility. It is a novel application of the divide-and-conquer principle in network design. We use predicted global knowledge to periodically "push" the overlay network into an optimized state. We maintain the network in a near-optimal state, even in the presence of sudden network changes (such as partitions or congestion), using a mission-aware distributed transport protocol. In this section, we describe the following three major aspects of the architecture (see Figure 1):

- The Predictive Flow Optimizer (Section II-A) solves an optimization problem to come up with a plan for routing traffic in the MON. It solves a non-concave bilinear optimization problem periodically using the predicted network state and projected future traffic conditions.
- MONtra (Section II-B) solves an online optimization problem to react to changes in the network. Using ideas from network utility maximization [12], it adjusts the sending rates of each site to solve a convex optimization problem.
- A mapping between PFO and MONtra (Section II-C) ensures that when PFO has full knowledge of the network, MONtra will converge to PFO's target rates. Our main result is that this convergence happens if MONtra has the same gradient as PFO at the target rates.

A. Predictive Flow Optimizer

The Predictive Flow Optimizer (PFO) outputs a routing plan for the network that maximizes mission utility. It runs periodically using a prediction of future network conditions and traffic demands.

PFO performs "call admission" by choosing a number of sessions n_k to admit in each class $k \in K$. It can decide to admit no sessions at all for a given class k by setting n_k to zero. In addition, PFO chooses a per-flow rate x_f to provide to each admitted session of class $k \in K$ along a route $f \in \rho_k$ in the network. The output of PFO is then used to set the parameters of the MONtra controllers, a process we call "mapping". Thus, PFO solves a hard global optimization problem, albeit in a non-realtime fashion using predicted traffic and network states.

The Optimization problem: PFO runs periodically and solves the following optimization problem to route a predicted set of sessions on the overlay. PFO takes as input a set of traffic classes K. Each traffic class $k \in K$ has a set of N_k sessions that need to be routed from a specific source site to a specific destination site. For instance, a traffic class could be all the VOIP phone calls made from a given site to another given site.

PFO has the option to send traffic along different paths in the network. For example, it might send traffic directly from one site to another, or send it indirectly via a number of enclaves. We say that a *flow* corresponds to the unique pair of a traffic class and a route through the network. For each class k, PFO has a set of ρ_k of possible flows. Each flow $f \in \rho_k$ starts at the source and ends at the destination associated with the class, using zero or more sites as intermediate nodes. Let ρ denote the set of flows for all classes.

PFO uses a model of the underlying network to pick a feasible set of rates for the flows in each class. Let L be the set of underlay network links, and C the set of link capacities. The capacity of link $l \in L$ is C_l. For convenience, we will write $l \in f$ to denote the links used by flow f and $l \ni f$ to denote the flows that use link l. PFO uses an estimate of the set of links $\hat{L}$, and an estimate $\hat{C}$ of the link capacities.

PFO is said to have *full knowledge* of the network if $L = \hat{L}$ and $C = \hat{C}$. At a minimum, PFO knows the uplinks for each site and their capacities, i.e., $\hat{L}$ is the set of uplinks from the MONtra nodes to the public Internet (see Figure 1). Between these two extremes, PFO may incorporate partial knowledge of the links and capacities using tools from network tomography (e.g. [5]).

In addition to the above, the overlay operator provides PFO a mission utility function $U_k(x_k)$, for each traffic class k, representing the value of giving one session of class k a rate of x_k. We assume that the mission utility functions are from $\mathbb{R}^+ \to \mathbb{R}$, are subdifferentiable everywhere, and are non-decreasing. We do *not* assume that they are concave, in order to incorporate mission utilities for inelastic traffic [10].

For each traffic class $k \in K$, PFO outputs a number of allowed sessions n_k. For each class k and possible flow $f \in \rho_k$, PFO outputs a target rate x_f which corresponds to the amount of traffic MON sends for a *single* session along flow f. To do so, PFO solves the following non-concave optimization problem:

$$\max_{n,x} \sum_{k\in K} n_k U_k(\sum_{f\in\rho_k} x_f) \tag{1}$$

$$\begin{aligned}
\text{subject to } & \sum_{k\in K}\sum_{f\in\rho_k;l\ni f} n_k x_f \le \hat{C}_l && \forall l \in \hat{L} \qquad (2)\\
& n_k \le N_k && \forall k \in K\\
& x_f \ge 0 && \forall k \in K, f \in \rho_k\\
& n_k \in \mathbb{Z} && \forall k \in K
\end{aligned}$$

Solving the Optimization Problem: The above optimization problem is NP-Hard, since the number of sessions must be an integer and the mission utility functions are not concave. However, for certain mission utility functions there are optimization techniques which make solving this problem more feasible. For instance, if the mission utility functions are piecewise linear, the problem becomes a bilinear program that can be solved efficiently using the ANTIGONE solver [20]. Other approaches for solving non-concave network utility maximization problems offline are described in the literature (e.g. [10]).

B. MON Transport Control (MONtra)

MONtra works at the transport layer of MON, and is responsible for reacting rapidly to changes in the underlying network. MONtra consists of weighted proportionally-fair congestion controllers, which route session traffic to match the rate chosen by PFO.

For each overlay route, MONtra's controllers optimize the transport-layer utility function $V_f(x_f) = w_f \log x_f$ (which should not be confused with the mission utility function $U_k(x)$). In Section II-C, we will describe how to choose weights for these controllers so that they provably converge to PFO's target rates. In an attempt to make the mapping easier to understand, we will model MONtra as using one controller per session on a flow. It's possible to extend the model to combine all the sessions on a flow into one controller. MONtra solves the following optimization problem, where L is the set of links in the network and C_l is the capacity of a link:

$$\begin{aligned}
& \max_{x} \sum_{f\in\rho} n_{k(f)} V_f(x_f)\\
\text{subject to } & \sum_{k\in K}\sum_{f\in\rho_k;l\ni f} n_k x_f \le C_l && \forall l \in L\\
& x_f \ge 0 && \forall k \in K, f \in \rho_k
\end{aligned}$$

We solve this optimization problem using techniques from Network Utility Maximization [13]. We initialize the controllers to the the rate selected by PFO, then adapt the rate based on network feedback. After each success/loss signal, we adjust rates according to the following update rules, where γ is a constant chosen for stability:

$$\begin{aligned} x_f &\leftarrow x_f + \gamma \cdot w_f && \text{(after each successful packet)} \\ &\leftarrow x_f - \gamma \cdot x_f && \text{(for each loss)} \end{aligned}$$

Note that unlike existing multipath TCP research (e.g. [11], [13], [15], [22], [28]), MONtra uses uncoupled controllers. We would like our controllers to exactly match the target rates chosen by PFO, which implies the transport optimization problem should have a unique optima. Unfortunately, the coupled controllers in the multipath literature allow multiple optima.

Instead of using a window-based controller, we use a rate based controller which sends packets at a rate of x_f. To do this, we generate delays between each packet so that the packet sending process is Poisson with rate x_f.

C. Mapping

The mapping layer is responsible for ensuring that MONtra converges to the set of rates chosen by PFO. Intuitively, we would like the transport utility function to act like the mission utility functions in the vicinity of the target rates selected by PFO. If we could ensure that MONtra sends at the same rate as PFO *and* has the same derivative at PFO's target rates, MONtra might behave in the same way as PFO even if there were slight changes to the network. The following theorem uses similar intuition and allows us to prove that MONtra converges to PFO's target rates.

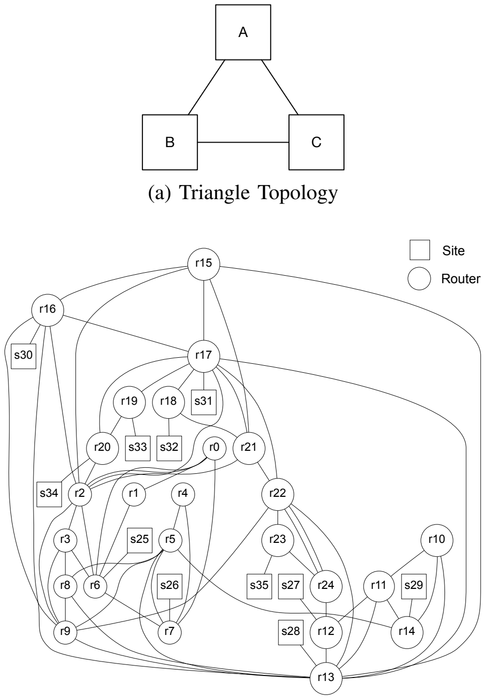
Theorem 1. *Suppose PFO has full knowledge of the network and selects a set of rates A and a number of sessions n_k for each class k that maximizes the mission utility function $U(A)$. For each link l, let λ_l be the dual variable associated with the capacity constraint for link l. Fix the number of sessions for each class in the transport layer to n_k. Using the following transport utility functions for a given flow f with class k, MONtra's rates will converge to A:*

$$\begin{aligned} V_f(x_f) &= w_f \log x_f \\ w_f &= n_k \big(\sum_{l \in f} \lambda_l \big) A_f \end{aligned}$$

Proof: See Appendix A ■

Note that if the only active constraint in the PFO solution is constraint (2), this mapping confirms the earlier intuition that we should match the PFO gradient at the target operating point. Since only the rate-related constraints are active for PFO, by the PFO KKT conditions, $\frac{\partial}{\partial x_f} U(A) = (\sum_{l \in f} n_k \lambda_l)$. Therefore, our mapping simplifies to $w_f = \frac{\partial}{\partial x_f} U(A) A_f$. At the target operating point, the partial derivative of the MONtra utility function with respect to a flow f is $\frac{w_f}{A_f} = \frac{\partial}{\partial x_f} U(A)$. So in addition to matching the rate, in this case we would also expect the MONtra utility functions to approximate the PFO utility functions close to the operating point.

This mapping theorem also works for any implementation of MONtra and other formulations of PFO optimization problem. For instance, MONtra could use other classes of concave transport utility functions, or another method of distributed optimization such as backpressure routing. The PFO optimization problem could use another way of combining per-flow rates, e.g. by summing the *weighted* rates across all paths instead of summing the rates across all paths.



(a) Triangle Topology

(b) AT&T Topology

Fig. 3. Experimental Topologies

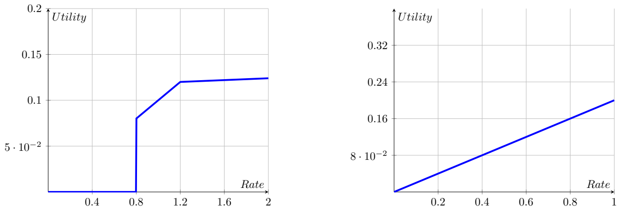
III. Evaluation Methodology

To show how MON performs in a realistic setting, we implemented MON and ran it on a set of network topologies, traffic scenarios, and mission utility functions as outlined below.

Network Testbed. We ran the experiments on Deterlab [19], which allowed us to allocate physical linux machines for each site and router in the network. We used Linux's traffic control system to set the network bandwidth. We used token bucket filters with a burst size of $100kb$ and a maximum queue latency of $5ms$, which provides stable throughput when we transfer files between the hosts.

Network Topologies. We emulated the small triangle topology shown in Figure 3(a) to illustrate MON behavior in an easier to understand context. We also emulated several large topologies from [16] (AT&T USA, Bell Canada, BTN, and Abilene). We present the results for the AT&T topology shown in Figure 3(b) in most of our experiments.

Mission Utility Functions. We use the following two mission utility functions to illustrate the behavior of MON, though



(a) Class A

(b) Class B

Fig. 4. Mission Utility Functions for Traffic Class A and B

our system works for arbitrary mission utility functions:

$$U_A(x) = \begin{cases} 0 & \text{if } x \leq 0.8 \\ \min(0.1x, 0.005x + 0.114) & \text{otherwise} \end{cases}$$

$$U_B(x) = 0.2x$$

The type A mission utility function has non-concavity and monotonically increasing utility with diminishing returns. The type B mission utility function increases linearly with rate without a point of diminishing returns. These functions are shown in Figure 4.

PFO Implementation. Since our approach reduces the PFO optimization to a bilinear program, we used the ANTIGONE solver [20] to efficiently solve the optimization problem. The solver uses branch-and-bound techniques and convex relaxations such as McCormick's envelopes which allow bilinear optimization problems to be solved efficiently.

MONtra Implementation. The MONtra implementation is based on multipath network utility maximization theory from [13]. We used the controllers described in Section II-B with per-packet acks to detect congestion. We gave each host an infinite backlog of data to send. We always used the number of sessions chosen by PFO. We set the stability constant $\gamma = 0.001$ and further improved stability by dividing γ by the largest weight (i.e. $\max_{r \in \rho} w_r$).

IV. Evaluation Results

A. *Does the overlay optimize mission utility?*

Our first experiment shows that MON optimizes mission utility for simple scenarios. We give PFO full knowledge of the network topology and capacities, and show that MONtra converges to PFO's target rates. We show this for both a simple topology and a more realistic one.

For the simple topology, we set up three nodes on Deterlab using the triangle topology shown in Figure 3(a). We set the capacity between Node B and Node C to 5Mbps and set the capacity of all other links to 10Mbps. We have two traffic classes, one between Node A and Node C and one between Node B and Node C. PFO assigns a rate of 10 Mbps between Node A and Node C and a rate of 5 Mbps from Node A to Node C via Node B. Table 5(a) shows the rates MONtra converged to for the two flows. In this simple example, there are no shared links between the flow. Note that the actual rates achieved by MONtra is close to the target rate set by PFO.

Path	Target	Actual
$A \rightarrow C$	10.0	9.71
$A \rightarrow B \rightarrow C$	5.0	4.84

(a)

Path	Target	Actual
$s35 \rightarrow s26$	10.0	9.24
$s29 \rightarrow s31$	10.0	9.24
$s33 \rightarrow s35$	10.0	9.24
$s31 \rightarrow s33$	10.0	9.23
$s34 \rightarrow s32$	6.0	5.9
$s26 \rightarrow s34$	6.0	5.68
$s32 \rightarrow s29$	6.0	4.4
$s32 \rightarrow s34$	4.0	3.55
$s26 \rightarrow s32$	4.0	3.4
$s34 \rightarrow s29$	4.0	3.24

(b)

Fig. 5. MONtra rates converged to the optimal target rates set by PFO for both the triangle topology and the AT&T topology

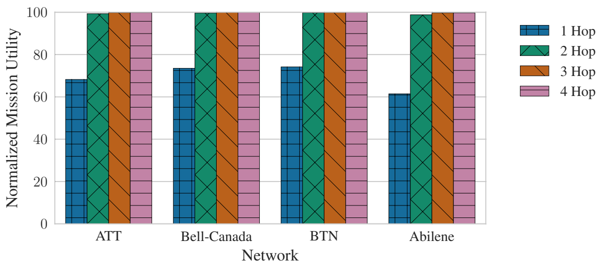


Fig. 6. Mission utility increases with the number of allowable hops, though two hops are sufficient to obtain most of the benefits.

We also ran this experiment on the more realistic AT&T network topology. We set the capacity of all links to 10Mbps, and ran PFO using the actual topology and capacities. Unlike the triangle experiment, PFO shared links between flows. Table 3(b) shows the rates MONtra converged to, which are close to the rates computed by PFO.

B. *Are overlay paths useful? If yes, how many? How does random path selection compare with choosing the best paths?*

Two sites can communicate directly with each other, or they can communicate indirectly via a series of other sites. The impact of overlay routing on reliability and performance for traditional best-effort overlays are well-known [1], [2], [23]. Here we ask analogous questions for mission-optimized overlays by ascertaining the benefits of overlay paths for mission utility maximization.

To study the impact of overlay paths, we ran PFO on a variety of real network topologies from [16] (AT&T USA, Bell Canada, BTN, and Abilene). We set uplink capacity to 30Mbps and the capacity between routers to 10Mbps. To simulate a partially loaded overlay, we created traffic classes between half the sites. We first restricted PFO to use only direct, one-hop paths, then to using one-hop and two-hop paths, and then

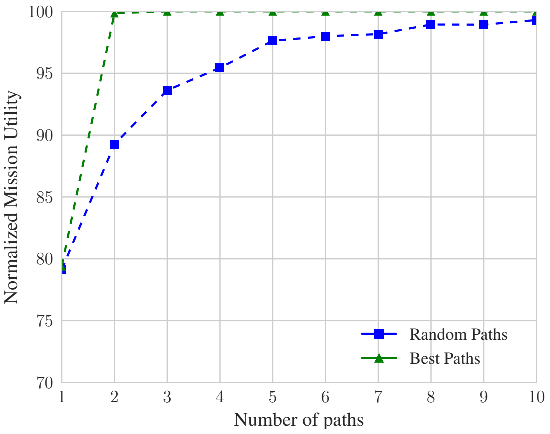


Fig. 7. The impact of using additional indirect paths on mission utility.

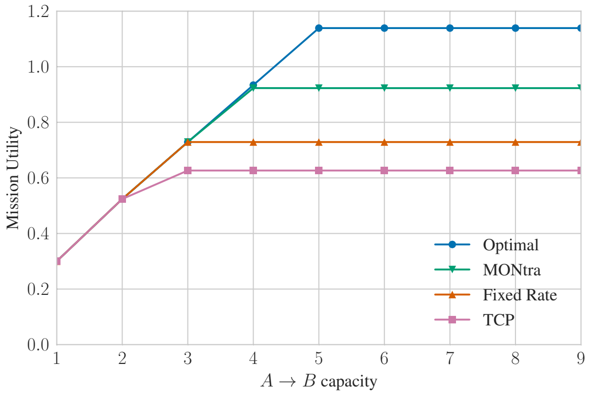


Fig. 8. How MONtra reacts when PFO has incorrect topology knowledge. We ran MONtra with PFO outputs for an $A \to C$ capacity of 3 Mbps, and plotted the rates it converged to for various values of $A \to C$ capacities.

to three- and four-hop paths. Note that the optimal mission utility cannot decrease when we allow a greater number of hops, since more hops corresponds to a larger feasible region. Figure 6 shows the results of this experiment. In all networks, there was more than a 20% increase in mission utility from using two hop paths over just the one-hop direct path. In this particular experiment, there was no benefit to using paths longer than two hops.

Next, we consider the impact of the number of allowed paths (i.e. $|\rho_k|$) on mission utility. It may be useful to not have to consider all overlay paths when running PFO. As shown in Figure 7, adding just one well-chosen indirect path for each class can be sufficient to obtain the maximum mission utility. However, we generally do not know the single best path before running PFO. Including a few random paths is also sufficient to improve mission utility. Just one randomly picked indirect path gave 80% of the optimal mission utility, and adding four random indirect paths gave 95% of optimal.

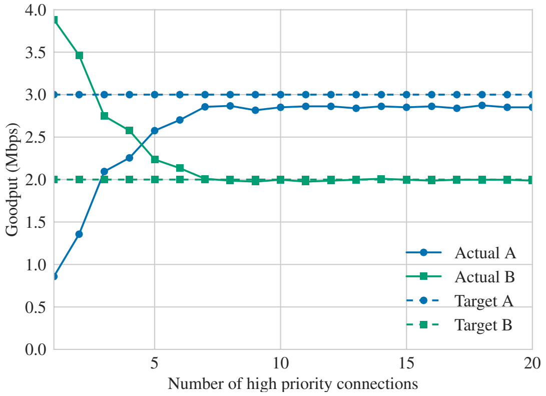


Fig. 9. How MONtra reacts when demand changes. PFO chose 11 sessions for the high priority flow, but we varied the number of sessions from 1 to 20.

C. *How does MONtra react to slight changes?*

Our next experiment shows that MONtra optimizes mission utility better than existing transport layer implementations, even when the network is slightly different from what PFO expects. We compare MONtra to a transport layer which uses Scalable TCP [14] without trying to match PFO's rates, and a transport layer which always sends at PFO's target rates.

This experiment used a mathematical model of the triangle network. We consider flows over links $A \to B$ and $B \to C$. We set link $B \to C$ to 5Mbps, and run PFO with the capacity on $A \to B$ set to 3Mbps. We then used those PFO rates for different capacities on $A \to B$. We tested capacities of 1Mbps to 10Mbps in intervals of 1Mbps.

Figure 8 shows the result of this experiment. When the capacity is 3Mbps and PFO has correct knowledge of the network, MONtra and the fixed-rate implementation match PFO's target rates. When link $A \to B$'s capacity is between 1 and 4 Mbps, MONtra was also able to match PFO's rates. Although MONtra had different rates between 5 and 10Mbps, it had a higher mission utility than the other two implementations.

D. *What if demand changes?*

In our previous experiments, we allowed PFO to choose the number of sessions used by MONtra. In a real system, the number of sessions might be less predictable; PFO may choose to admit n_k sessions and then many more sessions could arrive. In this experiment, we show how MONtra reacts when the number of sessions is more or less than what PFO chooses.

This experiment uses the triangle network. We set capacities $A \to C$ to 3Mbps, $B \to C$ to 5Mbps, and $A \to C$ to 10Mbps. PFO split the capacity of the $B \to C$ link between a flow of class A and a flow of class B. PFO gave the flow of class A 3Mbps for eleven sessions, and gave the flow of class B the remaining 2Mbps for one session.

We then vary the number of sessions for the flow of class A from 1 to 20. Figure 9 shows the rates that MONtra converged to. For 7 sessions and up, MONtra converged to PFO's target rate. Below 7 sessions, the two systems diverged and MONtra sent more of flow B than PFO.

E. How does MON react to failures?

This experiment shows that MON adapts to sudden, large changes in the network. At the start of the experiment, we provide PFO full knowledge of the network and feed its output to MONtra. We then cause a link failure, and allow MONtra to react. We finally re-run PFO with the updated topology, and it picks a new, more optimal set of rates to route around the failure.

In the triangle network, we set the capacity between nodes B and C to 5Mbps, and set all other capacities to 10Mbps. MONtra begins with the rates selected by PFO. At 60 seconds, we set the capacity between nodes A and B to 1Mbps. At 140 seconds, we re-run PFO and update MONtra to use the new flows and rates. Figure 10(a) shows the rates over time for the experiment, and Figure 10(b) shows the mission utility over time. When the network failure occurs, MONtra adjusts its sending rate to compensate. When PFO is re-run, it selects a new set of rates that increase the overall mission utility. The drop in mission utility at 140 seconds is because our prototype doesn't gracefully switch flows after PFO runs.

In the AT&T network, we set the capacity on all links to 10Mbps. At 40 seconds, we removed all links connected to the two routers with the highest degree ($r2$ and $r13$). Just before 150 seconds, we re-ran PFO to select a new set of target rates. Figure 11 shows the mission utility over time. Again, MONtra reacts to the network failure, and PFO selects new target rates that increase mission utility. This shows that MON is resilient to network failures.

V. Related Work

Overlays have been studied and built for the past 25 years. Large CDNs such as Akamai [21] have built overlays for delivering web and video content since the late 1990's [25]. However, these overlays are "best-effort" overlays that attempt to provide higher reliability and performance than what the native Internet can offer. Best-effort overlays come in many flavors, including caching overlays for Web content [7], routing overlays for reliably transporting live video streams [3], [17], P2P overlays for downloads [24], [26], [29], and security overlays for preventing DDoS attacks [25]. However, such overlays do not attempt to explicitly optimize the "mission goals" of the enterprise that operates the overlay, the focus of our work.

There has been prior work on overlay networks driven by quality-of-service (QoS). Networks that focus on QoS tend to guarantee each flow a particular performance metric such as rate, packet loss, jitter etc [8], [27]. In contrast, MON works by optimizing the cumulative mission utility of the overlay traffic. In particular, MON might sacrifice the QoS of some (lower-priority) traffic flows to enhance the QoS of other (higher-priority) flows.

MON uses the network utility maximization (NUM) framework, first pioneered by Kelly [12] who described distributed algorithms for optimizing concave utilities. Non-concave NUM problems are one of the major open problems in the field (e.g. [6]; sec. V-E). The general approaches are to solve the problem offline (e.g. [10]), which means the system adapts slowly at best or not at all at worst to network failures, or to approximate the problem with a distributed algorithm (e.g. [18]), which may not converge to a globally optimal point. Our two-tiered approach of periodically performing a global non-concave optimization to drive real-time transport controllers with logarithmic utilities is a novel approach to this classical problem.

VI. Conclusions

In this paper, we proposed a Mission-optimized Overlay Network which routes traffic to explicitly optimize the goals of an organization. By incorporating utility into its routing decisions, MON's decisions are more useful than prior, best-effort overlays. We proposed a novel, two-tiered architecture for this overlay. The higher-tier PFO periodically performs a global optimization of cumulative mission utility, allowing it to maximize the complex, non-concave utility functions that occur in practice. The lower-tiered MONtra transport protocol uses PFO's target rates to keep the overlay network near the optimal operating point by responding to network and traffic events in real-time. This architecture also opens up a new approach to investigating non-concave Network Utility Maximization (NUM) problems which has not been studied in the prior literature.

We implemented a prototype of our architecture on physical hardware, and showed that MON converges quickly to a state where the cumulative mission utility is maximized. By using mission utility information, MON can respond gracefully to network failures and changes in demand.

Acknowledgments

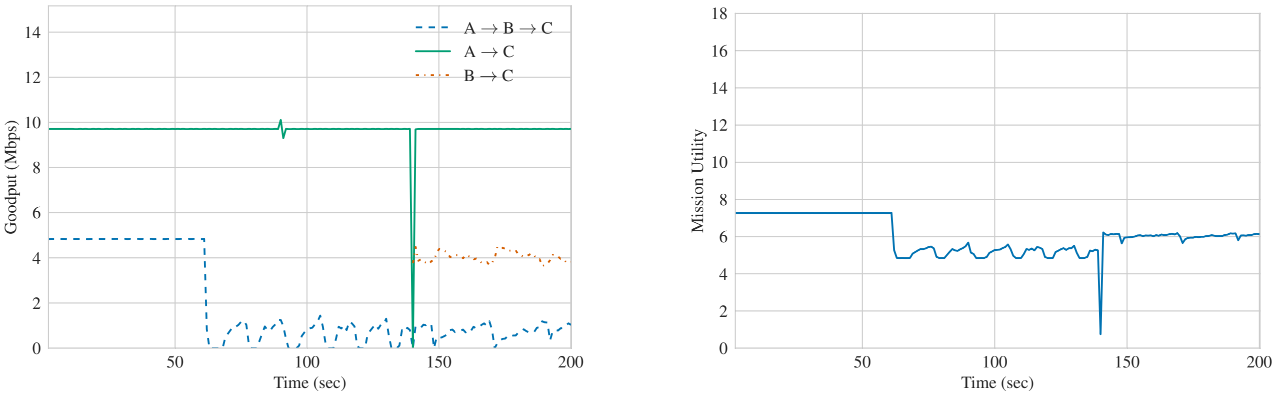
The research reported in this paper was sponsored by DARPA under Contract No. N66001-15-C-4045. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies of DARPA and the the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. Approved for public release; unlimited distribution.

References

[1] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks. *SIGOPS Oper. Syst. Rev.*, 35(5):131–145, Oct. 2001.

[2] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan. Best-path vs. multi-path overlay routing. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, IMC '03, pages 91–100, New York, NY, USA, 2003. ACM.

[3] K. Andreev, B. M. Maggs, A. Meyerson, and R. K. Sitaraman. Designing overlay multicast networks for streaming. In *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*, pages 149–158. ACM, 2003.



(a) Rates on triangle topology

(b) Mission utility on triangle topology

Fig. 10. MONtra rates and utilities with a link failure. We started MONtra with the correct PFO outputs. At 60 seconds, we reduced the capacity of the $A \to B$ link. We allowed MONtra to adapt, then at 140 seconds, we re-ran PFO to correct the rates. The overall utility went down after the failure, though PFO was able to improve the performance over only MONtra.

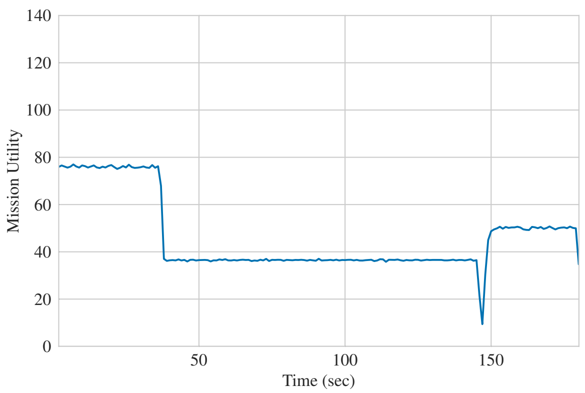


Fig. 11. Mission utility over time for the partition experiment on the AT&T network. At 40 seconds, we removed all links connected to the two most connected routers. At 150 seconds, we re-ran PFO which increased mission utility.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[5] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network tomography: recent developments. *Statistical science*, pages 499–517, 2004.

[6] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, Jan 2007.

[7] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl. Globally distributed content delivery. *Internet Computing, IEEE*, 6(5):50–58, 2002.

[8] Z. Duan, Z.-L. Zhang, and Y. T. Hou. Service overlay networks: Slas, qos, and bandwidth provisioning. *IEEE/ACM Trans. Netw.*, 11(6):870–883, Dec. 2003.

[9] H. Eriksson. Mbone: The multicast backbone. *Communications of the ACM, 37(8)*, 1994.

[10] M. Fazel and M. Chiang. Network utility maximization with nonconcave utilities using sum-of-squares method. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1867–1874, Dec 2005.

[11] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley. Multipath tcp: a joint congestion control and routing scheme to exploit path diversity in the internet. *IEEE/ACM Transactions on Networking (TON)*, 14(6):1260–1271, 2006.

[12] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 1997.

[13] F. Kelly and T. Voice. Stability of end-to-end algorithms for joint routing and rate control. *SIGCOMM Comput. Commun. Rev.*, 35(2):5–12, Apr. 2005.

[14] T. Kelly. Scalable tcp: Improving performance in highspeed wide area networks. *SIGCOMM Comput. Commun. Rev.*, 33(2):83–91, Apr. 2003.

[15] R. Khalili, N. Gast, M. Popovic, U. Upadhyay, and J.-Y. Le Boudec. Mptcp is not pareto-optimal: performance issues and a possible solution. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2012.

[16] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan. The internet topology zoo. *IEEE Journal on Selected Areas in Communications*, 29(9):1765–1775, October 2011.

[17] L. Kontothanassis, R. Sitaraman, J. Wein, D. Hong, R. Kleinberg, B. Mancuso, D. Shaw, and D. Stodolsky. A transport layer for live streaming in a content delivery network. *Proceedings of the IEEE*, 92(9):1408–1419, 2004.

[18] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Non-convex optimization and rate control for multi-class services in the internet. *IEEE/ACM Transactions on Networking*, 13(4):827–840, Aug 2005.

[19] J. Mirkovic, T. V. Benzel, T. Faber, R. Braden, J. T. Wroclawski, and S. Schwab. The deter project: Advancing the science of cyber security experimentation and test. In *In Technologies for Homeland Security (HST), 2010 IEEE International Conference on*, page 7, 2010.

[20] R. Misener and C. A. Floudas. ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. *Journal of Global Optimization*, 2014. DOI: 10.1007/s10898-014-0166-2.

[21] E. Nygren, R. K. Sitaraman, and J. Sun. The akamai network: a platform for high-performance internet applications. *SIGOPS Oper. Syst. Rev.*, 44:2–19, August 2010.

[22] Q. Peng, A. Walid, J. Hwang, and S. H. Low. Multipath tcp: Analysis, design, and implementation. *IEEE/ACM Transactions on Networking (ToN)*, 24(1):596–609, 2016.

[23] H. Rahul, M. Kasbekar, R. Sitaraman, and A. Berger. Towards realizing the performance and availability benefits of a global overlay network. In *Proc. of Passive and Active Measurement Conference*. Citeseer, 2006.

[24] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '01, pages 161–172, New York, NY, USA, 2001. ACM.

[25] R. K. Sitaraman, M. Kasbekar, W. Lichtenstein, and M. Jain. Overlay networks: An akamai perspective. *Advanced Content Delivery, Streaming, and Cloud Services*, pages 305–328, 2014.

[26] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup

protocol for internet applications. *IEEE/ACM Trans. Netw.*, 11(1):17–32, Feb. 2003.

[27] L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz. OverQoS: An Overlay Based Architecture for Enhancing Internet QoS. In *1st Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, March 2004.

[28] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath tcp. In *NSDI*, volume 11, pages 8–8, 2011.

[29] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: A fault-tolerant wide-area application infrastructure. *SIGCOMM Comput. Commun. Rev.*, 32(1):81–81, Jan. 2002.

Appendix A
Proof of mapping theorem

Theorem 1. *Suppose PFO has full knowledge of the network and selects a set of rates A and a number of sessions n_k for each class k that maximizes the mission utility function $U(A)$. For each link l, let λ_l be the dual variable associated with the capacity constraint for link l. Fix the number of sessions for each class in the transport layer to n_k. Using the following transport utility functions for a given flow f with class k, MONtra's rates will converge to A:*

$$\begin{aligned} V_f(x_f) &= w_f \log x_f \\ w_f &= n_k\big(\sum_{l \in f} \lambda_l\big) A_f \end{aligned}$$

Proof: Recall the PFO solves the following optimization problem:

$$\begin{aligned} \max_{n,x} \quad & \sum_{k \in K} n_k U_k\Big(\sum_{f \in \rho_k} x_f\Big) & \\ \text{subject to} \quad & \sum_{k \in K} \sum_{f \in \rho_k; l \ni f} n_k x_f \leq \hat{C}_l & \forall l \in \hat{L} \\ & n_k \leq N_k & \forall k \in K \\ & x_f \geq 0 & \forall k \in K, f \in \rho_k \\ & n_k \in \mathbb{Z} & \forall k \in K \end{aligned}$$

And MONtra solves the following optimization problem:

$$\begin{aligned} \max_{x} \quad & \sum_{f \in \rho} n_{k(f)} V_f(x_f) & \\ \text{subject to} \quad & \sum_{k \in K} \sum_{f \in \rho_k; l \ni f} n_k x_f \leq C_l & \forall l \in L \\ & x_f \geq 0 & \forall k \in K, f \in \rho_k \end{aligned}$$

Since we assume PFO has perfect information, $\hat{L} = L$ and $\hat{C} = C$.

Since the set of rates A, the number of admitted sessions n_k, and the dual variables for each link λ_l and for each constraint on the number of sessions λ_k are optimal for PFO, they satisfy the following KKT conditions (See [4]; Sec. 5.5.3):

$$\sum_{k \in K} \sum_{f \in \rho_k; l \ni f} n_k A_f \leq C_l \qquad \forall l \in L \tag{3}$$

$$A_f \geq 0 \qquad \forall k \in K, f \in \rho_k \tag{4}$$

$$n_k \leq N_k \qquad \forall k \in K$$

$$n_k \in \mathbb{Z} \qquad \forall k \in K$$

$$\lambda_l \geq 0, \qquad \forall l \in L \tag{5}$$

$$\lambda_l\Big(\sum_{k \in K} \sum_{f \in \rho_k; l \ni f} n_k A_f - C_l\Big) = 0 \qquad \forall l \in L \tag{6}$$

$$\frac{\partial}{\partial x_f} U(A) = \sum_{l \in f} n_k \lambda_l \tag{7}$$

$$\frac{\partial}{\partial n_k} U(A) = \sum_{k \in k} n_k \lambda_k \tag{8}$$

For a set of target rates A' and link dual variables λ'_l to be optimal for the MONtra problem, they must satisfy the following KKT conditions for the MONtra problem:

$$\sum_{k \in K} \sum_{f \in \rho_k; l \ni f} n_k A'_f \leq C_l \qquad \forall l \in L \tag{9}$$

$$A'_f \geq 0 \qquad \forall k \in K, f \in \rho_k \tag{10}$$

$$\lambda'_l \geq 0, \qquad \forall l \in L \tag{11}$$

$$\lambda'_l\Big(\sum_{k \in K} \sum_{f \in \rho_k; l \ni f} n_k A'_f - C_l\Big) = 0 \qquad \forall l \in L \tag{12}$$

$$\frac{d}{dx_f} V_f(x_f) = \sum_{l \in f} n_k \lambda'_l \tag{13}$$

Since the MONtra problem is concave, if some point A', λ'_l satisfy the MONtra KKT conditions then it is optimal and MONtra will converge to that point. We will show that PFO's optimal rates A and link dual variables λ_l satisfy these KKT conditions, so MONtra will converge to the PFO optimal point.

The MONtra KKT conditions (9) and (10) are satisfied since they are identical to the PFO KKT conditions (3) and (4) because we've fixed the number of flows in the transport to the number of flows chosen by PFO. MONtra KKT condition (11) is satisfied by PFO KKT condition (5). MONtra KKT condition (12) is satisfied by the PFO KKT condition (6). MONtra KKT condition (13) is satisfied by our choice of w_f since for flow f with class k:

$$\begin{aligned} \frac{d}{dx_f} V_f(x_f) &= \frac{w_f}{A_f} \\ &= \sum_{l \in f} n_k \lambda_l \end{aligned}$$

Therefore A, λ_l satisfy the KKT conditions for MONtra. Since MONtra is concave, it will converge to A, λ_l. ■