Redundancy-Aware Routing with Limited Resources

Yang Song
University of Massachusetts, Amherst
ysong@ecs.umass.edu

Katherine Guo
Bell Labs Research, Alcatel-Lucent
kguo@bell-labs.com

Lixin Gao
University of Massachusetts, Amherst
lgao@ecs.umass.edu

***Abstract*—Network load is reduced upon elimination of redundant data transfers. Redundancy elimination techniques can be applied on a per-packet basis, and provide benefit regardless of application. While it is straightforward to apply redundancy elimination on a per-link basis, network cost can be further reduced by applying redundancy elimination network-wide: by routing potentially redundant packets (identified using a redundancy profile) onto common links.**

Constructing redundancy aware routes is challenging: it might not be economically viable to deploy redundancy elimination on every link. Also, to preserve end-to-end performance and control signaling cost, routes cannot be determined on a per packet basis. Routes have to be determined independent of packet content.

In this paper, we propose a redundancy-aware routing algorithm. Our protocol can cope with limited resources (in terms of number of links that can support redundancy elimination) and is feasible (it does not require per-packet routing decisions). We evaluate our algorithm using detailed simulations, based both on synthetic traffic and trace data captured from large enterprise networks. Unlike previous studies, our studies consider data from multiple sources. Our results show that a small number (less than five in our experiments) of routers equipped with redundancy elimination, coupled with our routing algorithm, is sufficient to achieve reduction in network load close to when redundancy elimination is applied at every router.

I. Introduction

Identifying and eliminating duplicate bytes in subsequent packets reduces communication cost. Prior work has described techniques for efficiently storing byte-string signatures [15]. identifying duplicate byte ranges in packets and reducing their transfer cost [19]; and evaluated the efficacy of these techniques in a network-wide setting [7].

These "redundancy elimination" (RE) [1] techniques can be applied on a per-link basis. Indeed, the early motivation for the research described in [19] was to reduce congestion on a campus-ISP link. Other single-link deployments include commercial "WAN aceleration" products that eliminate redudancy over end-to-end virtual links in enterprise VPNs [1], [2], [3], [4]. Results from WAN acceleration have shown that significant amount of traffic across enterprise sites can be eliminated if devices are installed in individual sites to offer end-to-end redundancy elimination between pairs of sites [5].

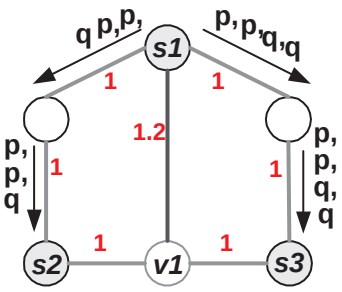
RE can also be applied network-wide. In such a setting, packets that are likely to be duplicates (or are likely to contain large duplicate strings) are routed away from their default path onto common links that support RE.

Figure 1 shows an example of how such re-routing might lead to overall savings. For ease of exposition, in this example, and in the rest of the paper, we assume that entire packets are duplicated. Each packet incurs the full link cost if it is transmitted in its original form. However, if a duplicate packet is detected, then only a pointer to the original packet needs to be transmitted. In practice, such a pointer, which identifies an original packet and byte ranges both in the original and the current packet can be encoded using 12 bytes [19]. Thus, the cost to transmit a duplicate packet is usually much less than an entire packet, and is represented using d. In the figure, original packets that traverse a link are shown in black, while duplicates are shown in gray. In this example, the cost for transmitting packets over all links in the network reduces from 14 (in the default case) to (8+6d) if duplicates are suppressed end-to-end without changing the default routes. Re-routing enables further savings since the two original packets only have to transmitted once. The total cost, with re-routing, reduces to (6.4+9d). In our example, packets are re-routed through the router $v1$, but $v1$ did not have to maintain a packet cache or re-construct compressed packets. In the general case, however, entire packets are not duplicated, but parts of packets *with arbitrary destinations* are. In these cases, interior nodes, such as $v1$, need to reconstruct compressed packets on-the-fly.

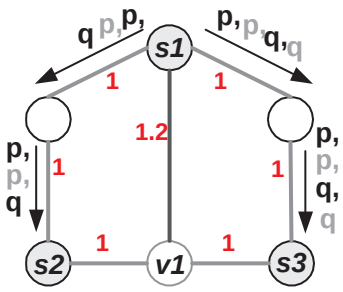
Network-wide RE and re-routing was originally proposed by Anand et al [7]. In their setting, all routers in a ISP network cache packets and eliminate duplicates. RE can be applied beyond a single ISP, for example in enterprise VPNs that span multiple ISPs. In this setting, VPN nodes themselves must maintain the packet cache and compress outgoing packets before they are encrypted. It is relatively straightforward to implement end-to-end RE in VPNs. Interestingly, VPN traffic from a single source to multiple destinations can have significant commonality. (Our own measurement study, described in Section V-A, shows that nearly 50% of traffic from one VPN node to a pair destinations were duplicative.) Hence, we expect RE with re-routing to provide at least some benefit for multi-site enterprise VPNs.

Unfortunately, the current models for RE with re-routing are infeasible to implement in practice. There are main two problems: First, existing work maximizes RE benefit by re-routing every packet independently. While theoretically sound, implementing such a scheme would incur prohibitive control plane overhead. Performance of existing protocols, including

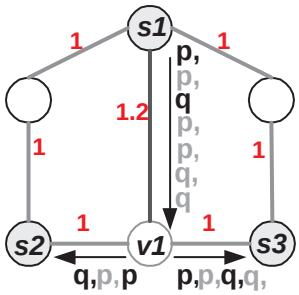
[1]In this paper, we use the term redudancy *elimination*. In practice, when duplicate packets or byte streams are detected, a pointer to a previous packet has to be transmitted, such that the receiver can reconstruct the duplicate packet.



(a) **Shortest Path** (SP) routing transmits 7 original packets, each over two unit cost hops, total network cost is 14.



(b) **End-to-end redundancy elimination with shortest path** (SP-E2E) transmits 4 original packets (two on each link) and 3 duplicate packets over the shortest paths. Network cost is (8+6d).



(c) **Redundancy aware routing** routes packets through v_1 for both destinations, and transmits only 2 original packets and 5 duplicate packets, reducing the network cost to $1.2*(2+5d)+1*(2+d)+1*(2+2d)$=(6.4+9$d$).

Fig. 1: Benefits of end-to-end and network-wide redundancy. Original packets (in black) are unit size, and the links in the figure are annotated with traversal cost. Using redundancy elimination, transmission of a duplicate packet (shown in gray) incurs cost $d \times$ link cost.

TCP, would suffer unduly due to packet re-ordering [10], [11]. As a result, even though the network cost would be provably minimized, network performance would degrade unacccceptably. The second problem is more pragmatic: Existing work assumes that *every* router in an ISP or enterprise network will be able to implement the RE protocols. Upgrading the entire network is likely to be cost prohibitive since the RE protocols require specialized hardware to be effective, including copious memory for the packet cache, fast cache lookups, and RE compression/decompression at line-speeds.

These two problems motivate our work. Our goal is to design a re-routing-based RE algorithm that provides tangible benefits using only limited resources. In particular, our scheme does not require per-packet re-routing or universal deployment of RE-capable routers. Our contributions are as follows:

- We extend the formulation of RE-aware re-routing problems to the resource-limited scenario. Unsurprisingly, the optimal solution remains NP-Hard; we propose a polynomial-time heuristic for deriving redundancy aware routes.
- We propose a simple model for accurately capturing traffic redundancy characteristics. In particular, our model captures redudancy in traffic between a single source-destination pair, and also in traffic between a single source and multiple destinations.
- We present measurement data from a multi-site VPN and empirically show the existence of redudancy in traffic from a single source to multiple destinations. Our data set captures traffic from a very large corporate network that has more than 20000 active users. We use this trace and our traffic model to generate synthetic traces for our evaluation.
- We present an extensive simulation-based evaluation of our heuristic algorithm using both the captured and synthetic trace data. Our heuristic compares favorably to the optimal (computed brute-force). We also present the network impact of RE re-routing; in particular, we consider the extra load induced on different routers and links.

The rest of the paper are organized as follows: We begin with a description of prior and related work. In Section III, we present our problem formulatation. In Section IV, we desribe our heuristic. Section V presents our evaluation. We conclude in Section VI.

II. Prior and Related Work

Network traffic exhibits large amount of redundancy when popular content is accessed by same or different users repeatedly. Many systems have explored this fact to eliminate redundant content transfers and improve network efficiency. Most systems are based on protocol dependent, object-level caching techniques. For example, web proxy caches store static web content in order to reduce bandwidh usage for either access networks or ISP networks [6].

Recently a new class of systems have been developed to be application-level protocol independent and operate at packet-level to elinimate redundant content. Packet-level RE schemes, first proposed by Spring et al [19], are becoming more popular as many vendors are deveoping WAN optimization solutions [1], [2], [3], [4].

The scheme in [19] is based on techniques for finding similar files in [13] using Rabin fingerprints [15]. For each incoming packet, Spring's protocol computes a set of representative fingerprints by first computing Rabin fingerprints of sliding windows of 64 contiguous bytes of the payload, and then selecting a fraction 1/32 of them as representative ones. All past packets are stored in a packet cache, and all representative fingerprints are stored in a fingerprint store.

The fingerprints serve as pointers into portions of the packet payload which is used to find redundant content.

Upon receiving a packet, the scheme first generates the set of representative fingerprints for the current packet, then compares them with the fingerprint store. For each fingerprint of the packet that is matched against the store, the matching packet is identified and the matching region is expanded byte-by-byte in both directions to obtain the maximal matching region. Once all matches are identified, the matching regions in the current packet are replaced with fixed-size pointers to the cache, thereby eliminating redundancy. Cache replacement policy is simply FIFO.

There are variations regarding how representative fingerprints are chosen. The "winnowing" mechanism chooses local maxima or minima over each fixed size region in payload [17]. Our redundancy-aware routing algorithm is independent of any packet-level RE schemes, and our evaluation in Section V uses the original scheme proposed in [19].

Most recently, Anand et al have proposed to deploy these systems on all ISP routers and proposed a redundancy-aware routing protocol to reduce network untilization [7], [9], [8]. This paper extends their work by considering practical constraints such as a limited number of routers can be equipped with RE and participate in redundancy aware routing. A comprehensive study on redundancy in network traffic is reported in [8]. Our real trace study further complements this study by demonstrating properties of multi-site enterprise traffic in a mesh.

III. Redundancy-aware Routing

In this section, we develop a formalization for the RE re-routing problem. The formalization will allow us to formulate a precise problem statement (which, unfortunately though not unsurprisingly, turns out to be NP-Hard.) We show that an optimal but impractical solution derives from the well-studies Steiner tree problem. In our formulation, we assume that an overlay network is constructed atop the physical topology. Existing shortest path routing (or whichever routing the ISP chooses) is used in the underlying network. The overlay network is used to re-route packets to facilitate RE.

Network and Traffic. $G(V, E)$ denotes the overlay network on top of a physical network G'. There are N end-nodes in the system $s_1, \ldots, s_N$. All nodes in the system, including the end-nodes, support the RE protocol.

We use the term *flow* to represent traffic from node s_i to s_j. In practice, a flow may be restricted to traffic over a time interval t. We model traffic from s_i to its destinations as a set of N flows $\{f_{i,1}, f_{i,2}, ..., f_{i,i-1}, f_{i,i+1}, ...f_{i,N}\}$, where $f_{i,j}$ represents the flow from s_i to s_j.

Duplicate Packet Model. We need to define a few parameters to model traffic redundancy profile. We use M_i distinct packets $\{p_{i,1}, p_{i,2}, ..., p_{i,M_i}\}$ to model traffic originating from s_i similar to [7]. We represent traffic from s_i as duplicates of the M_i distinct packets, and each distinct packet $p_{i,m}$ can have one or more copies, and we consider the original distinct packet to be the first copy. Copies of the distinct packet $p_{i,m}$ can have distinct destinations.

We define a list of constants $cp_{i,m,j}$ such that if a copy of distinct packet $p_{i,m}$ is destined for s_j, then $cp_{i,m,j} = 1$, and 0 otherwise. In Fig 1(a), for example, distinct packet $p_{1,1} = p$ originating at s_1 has four copies, two of which are destined for s_2 and two for s_3. Then, $cp_{1,1,2} = cp_{1,1,3} = 1$. We call the packet set and its corresponding list of $cp_{i,m,j}$ within time interval t the *redundancy profile* for traffic with source s_i.

We define *intra-flow redundancy* of a flow as the volume of redundant packets within the flow. For a group of flows originating from the same source, we define their *inter-flow redundancy* as the overlap among all flows after intra-flow redundancy has been removed by end-to-end (E2E) redundancy elimination within each flow. Note the list of $cp_{i,m,j}$ captures inter-flow redundancy after intra-flow redundancy has been eliminated.

Network Cost Calculation. Our goal is to construct a minimum cost overlay network using up to K overlay routers while using redundancy profile to re-route traffic. The way to build the overlay network is to establish tunnels for each source-destination pair through zero or more overlay routers. Total network cost is the sum of the cost for each packet as it traverses from its source to destination in the overlay network. To calculate the cost for each packet, we introduce a few parameters and variables below.

Let e represent an edge in the overlay G. Edge e is composed of the shortest path between its two end nodes in the physical network below. The latency over the edge is the sum of latency over physical network edges it is composed of. For every e, we define a variable $rt_{e,i,m}$ to indicate the occurance of packet $p_{i,m}$ over that edge. If at least one copy of packet $p_{i,m}$ goes through e , then $rt_{e,i,m} = 1$, otherwise it is 0. Binary values of $rt_{e,i,m}$ ensure that we only count every distinct packet once, because duplicated copies are removed by the redundancy elimination mechanism installed on both ends.

For every source s_i, we need to find the best tunnel $T_{i,j}$ from s_i to all its destinations s_j, going through zero or more overlay nodes (including end nodes and overlay routers).

We use a variable $F_{e,i,m}$ to represent the amount of resources consumed over edge e when copies of $p_{i,m}$ passes through e. Following [7], we call it the *footprint* of packet $p_{i,m}$ over edge e. Obviously when no copy of $p_{i,m}$ passes through e, $F_{e,i,m} = 0$. Because of redundancy elimination, if one or more copies of $p_{i,m}$ passes through e, then the amount of resources consumed is for one copy only. To capture the fact that more resources are consumed if packet size is larger or link latency is larger, we define $F_{e,i,m}$ to be equal to the size of the packet multiply the latency l_e of edge e in G, or $F_{e,i,m} = l_e|p_{i,m}|rt_{e,i,m}$, where $|p_{i,m}|$ is the size of $p_{i,m}$. Note that other functions of footprint can also be used. The cost of the overlay network is therefore the sum of footprint for every packet over every edge.

Optimization Problem Statement. The optimization problem can now be stated as follows: Given a physical network

G', a redundancy profile from every source s_i, and the constraint that a maximum number of K nodes in G' can be elevated to overlay routers in G, derive a set of edges to constitute the tunnel $T_{i,j}$ for every source-destination pair s_i and s_j, such that the cost of the whole overlay network $\sum_i(\sum_m(\sum_e F_{e,i,m}))$ is minimum.

a) Redundancy-Aware Routing is NP-Hard: The optimization problem trivially reduces to the classic Steiner tree problem simply by unconstraining K (such that it is unbounded) and assuming there is only a single source in the network. The Steiner tree problem has long known to be NP-hard [14], [16], [20].

b) An Optimal Solution using Steiner Trees: In fact, the complexity of redundancy-aware routing is the same as that of Steiner trees, and an optimal solution (referred to as "RA-O") can be formulated using Steiner trees. For every packet, construct a Steiner tree from its source to all its destinations, using any algorithm to compute a Steiner tree, e.g. [12]. All the packets' Steiner trees form the optimal solution for the overlay network routing. As very packet routed through the Steiner trees uses minimum network resources, the total network cost for all packets is therefore minimum.

We note that this optimal solution is a reformulation of the optimal presented in [7], where the authors essentially compute the Steiner trees as solution to the linear programing problem formulation.

IV. RA-H: A Heuristic

In this section, we present "RA-H", a greedy heuristic for redundancy-aware routing. The input to RA-H is the physical network G' and the maximum number of overlay routers K. Further, we assume that the traffic redudancy profile, i.e., the extent of (expected) duplication between flows from a single source, is also avialable. We discuss the derivation of this profile in Section III. RA-H outputs the overlay routes and routers for each source-destination pair.

We introduce the following notation: Let $\mathcal{R}_s$ be the traffic redundancy profile for the flows originating source s to N destinations $d_1, \ldots, d_N$. $\mathcal{R}_s$ is a set of pairs: each pair maps a subset of unique flows to the fraction of duplicate packets in those flows. An element in $\mathcal{R}_s$ might be $(\{s \rightarrow d_1, s \rightarrow d_2\}, 0.8)$, which means that 80% of the packets in the two flows $s \rightarrow d_1$ and $s \rightarrow d_2$ are duplicates[2]. We refer to the left hand side of the pair $(\{s \rightarrow d_1, s \rightarrow d_2\})$ as a *flow-set*. In general, the number of such (flow-set, duplicate fraction) entries in $\mathcal{R}_s$ is exponential in the number of destinations (N). In practice, only a subset of (perhaps only the pairs) would be computed and used in RA-H.

RA-H may re-route a flow set by routing all constituent flows through an overlay node. The overlay node implements the full RE protocol, can decapsulate packets as necessary, and forward (re-constructed) packets to their destination. In RA-H, packets are re-routed through at most one overlay hop. Recall that the underlying network paths are used to route to overlay nodes.

RA-H proceeds in rounds. In each round, a new set of flows F, *that has not been re-routed yet*, is re-routed as follows:

- For each source s, consider each flow set remaining in $\mathcal{R}_s$ that has not yet been re-routed. Let the C be current flow being considered.
- Compute the reduction (if any) in network cost for re-routing C through every router r in the network, regardless of whether r is already in the overlay.
- Pick F to be the flow-set that results in the maximum (positive) reduction in network cost. If the maximum is obtained by re-routing F through a node that is not yet in the overlay, add that node to the overlay.

The algorithm terminates when one of the following conditions hold: (1) the number of overlay routers reaches K or (2) no re-routed flow-set remains or (3) a round does not result in a network cost reduction. Figure 2 shows an example execution of the RA-H algorithm on a small network.

A. Computing Redundancy Profiles

As presented, the runtime complexity of RA-H is exponential in the number of active flows in the network. This is because, for source s RA-H must consider each flow-set in $\mathcal{R}_s$. Obviously, the storage or processing for iterating over complete redundancy profiles $\mathcal{R}$ is not reasonable or feasible for even small networks. Instead, we assume that some domain-specific knowledge will be used to pare down the set $\mathcal{R}$. In our experiments, almost all benefit was obtained by only considering flow sets with only two flows (i.e., all flow pairs). We expect curtailing the algorithm to only consider pairs will generall prove to be a reasonable compromise between runtime complexity and benefit.

Finally, we note that our redundancy profiles do not capture duplication between flows from different sources. While the algorithm could be extended to potentially consider this case, analysis of our traces show essentially zero redudancy in flows from different sources. Similar results are also reported in [8]. Hence, RA-H does not consider redudancy in flows that originate at different sources.

V. Evaluation

We evaluate RA-H and compare it to alternates in this section. We begin with a description and analysis of our measurement study, which is used both to motivate RA-H and to parametrize our synthetic traces used for simulations.

A. Measurement Study

We describe the key properties of content redundancy observed in real traces. We collected full packet traces at WAN access links for 5 sites of a large corporate network in North America. Our conservative estimate or the number of total unique network users is 25,000. Three of the sites (S_a, S_b and S_c) are corporate campuses, and the remaining two sites (D_a, D_b) are data centers. The hosts in the data centers are corporate

[2]Note that $\mathcal{R}$ is computed after the intra-flow redudancy in each flow has already been removed. In practice, the $\mathcal{R}$ may be computed over fixed time intervals, and the volume of duplicate bytes substituted for duplication ratio without affecting the algorithm.

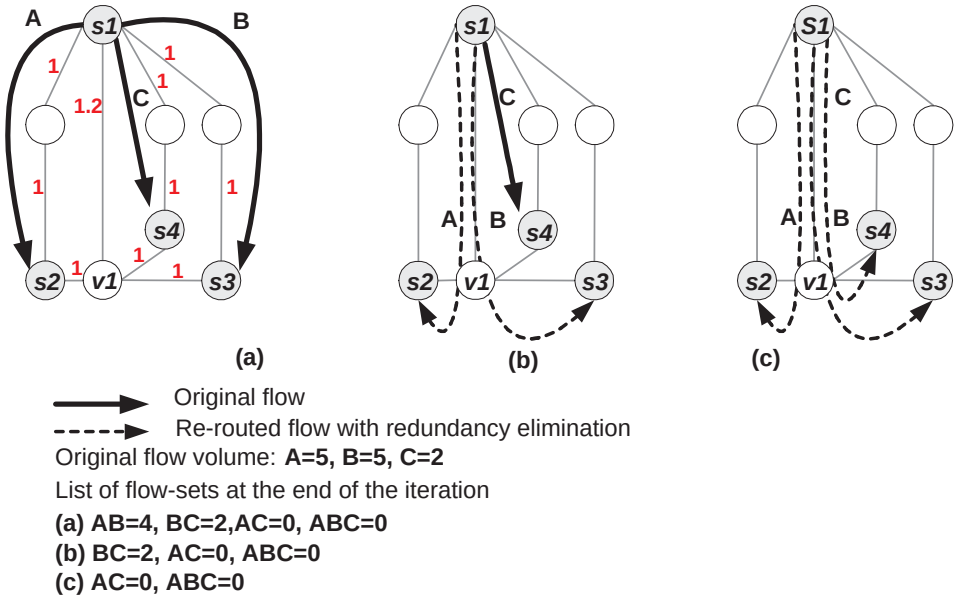


Fig. 2: Assume unit edge cost on all edges, except cost is 1.2 for edge s_1-v_1. (a) Start: flows A,B, and C following their shortest path routes. (b) Iteration 1: find the best node to re-route flow-set $\{A, B\}$. Node v_1 is chosen. Cost reduction=2.8. (c) Iteration 2: find the best node to re-route flow-set $\{B, C\}$. Node v_1 is chosen. Cost reduction =2.

TABLE I: Intra-flow Redundancy in the Real Trace

src	dst	volume	Intra-RR	src	dst	size	Intra-RR
Da	Sa	43.4%	32.6%	Db	Sa	0.6%	46.2%
Da	Sb	22.5%	43.1%	Db	Sb	1.0%	7.9%
Da	Sc	16.2%	33.5%	Db	Sc	0.5%	40.1%
Da	Db	0.3%	8.9%	Db	Da	1.2%	37.1%
Sa	Da	6.7%	40.1%	Sa	Db	0.3%	10.6%
Sb	Da	2.9%	28.8%	Sb	Db	0.2%	12.2%
Sc	Da	4.1%	16.6%	Sc	Db	0.1%	2.0%

Exchange Servers hosting e-mail, voice mail, RSS, SMS, and IM services. Our data collection captured the full mesh traffic among the 5 sites. Our collection consists of multiple 200 second snapshots every hour from 11:30am to 5:30pm EDT on July 2, 2009. The total volume of the trace data exceeds 24 Gigabytes, with more than 125 Million packets captured.

We use the algorithm described in Section II with 64 byte fingerprint window size and a cache size 5GB to analyze content redundancy. Table I shows the intra-flow redundancy for each flow; per-source volume is reported as the percentage of the total traffic volume. We define the *intra-flow redundancy ratio* (Intra-RR) as the ratio of the redundant content over the total volume of the flow. Different flow types have different redundancy, and the Intra-RR ratio ranges from 2.0% to 46%.

Our traces also contained significant intra-flow redundancy between 2-3 flows from data center D_a. 49% of f_{D_a,S_b} also appeared in f_{D_a,S_a}. However, inter-flow redundancy decreases once we eliminate intra-flow redundancy. For example, only 11% of f_{D_a,S_b} also appeared in f_{D_a,S_a}, and 6% of f_{D_a,S_b} appeared in both f_{D_a,S_a} and f_{D_a,S_c}.

We are conservative in computing inter-flow redundancy

TABLE II: Inter-flow Redundancy in the Real Trace

src	dst	Inter-RR	src	dst	Inter-RR
Da	Sa,Sb	6.7%	Db	Sa,Sb	1.8%
Da	Sa,Sc	8.5%	Db	Sa,Sc	1.2%
Da	Sb,Sc	1.4%	Db	Sb,Sc	0.5%
Da	Sa,Sb,Sc	1.2%	Db	Sa,Sb,Sc	0.4%

(Inter-RR in Table II). In particular, we compute inter-flow redundancy *after* eliminating all intra-flow redundancy in the constituent flows. Hence Table II tabulates how many unique byte ranges are duplicated in each flow.

Our analysis shows significant intra-flow redundancy that end-to-end RE can remove. The remaining inter-flow redundancy is non-trivial, and motivates our work. We next evaluate how well RA-H eliminates this inter-flow redundancy in comparison to the optimal algorithm, and quantify the network effects of the RA-H re-routing.

B. Synthetic Trace Model

In order to explore the relationship between various redundancy profiles, network topology, and the overall benefits offered by RE, we construct synthetic traces based on key properties of the captured data.

The process of generating flows from a source consists of two steps: (1) generating packets and (2) assigning destination to each packet. We use packets of identical size, and vary the number of packets to control traffic volume.

Following the duplicate packet model in Section III, flows from the source consists of distinct packets and copies of distinct packets. Packets from a source fall in to one of four categories:

(1) **Inter-flow Unique** (r_u) consists of packets that appear for the first time at the source. Subsequent copies of this packets will later be sent to other destinations.

(2) **Inter-flow Erasable** (r_e) consists of packet copies being sent to new destinations.

(3) **Intra-flow Unique** (R_u) consists of packets that are noted for the first time in a flow. Subsequent copies of this packet will only be sent to the same destination.

(4) **Intra-flow Erasable** (R_e) consists of packet copies for packets that are duplicated, but only to a single destination.

Note that whether a packet is inter- or intra-flow unique can only be determined post-hoc, after all the destinations for copies have been noted. Further, end-to-end mechanisms can eliminate all intra-flow eraseable packets, whereas inter-flow eraseable packets require in-network support.

Each packet must be classified into one of these four mutually-exclusive categores, and we use the variables r_u, R_u, r_e, R_e as defined above to denote the probability that a particular packet that is generated is of a specific type (inter/intra unique/eraseable). By construction, $r_u + R_u + r_e + R_e = 1$. We further define the overall ratio of erasable traffic as $O_e = R_e + r_e$, and overall ratio of unique traffic is therefore $1 - O_e = R_u + r_u$.

We then generate packets by assigning different values to the four probabilities. For each unique packet, we chose the destination uniformly at random. Each unique packet is marked either intra-flow unique or inter-flow unique with probability $R_u/(R_u + r_u)$ and $r_u/(R_u + r_u)$ respectively. For each intra-flow eraseable packet n, we pick a intra-flow unique packet (say u) unformly at random and set n's destination equal to that of u. For each inter-flow eraseable packet, we again start with a inter-flow unique packet. However, we choose the destinations such that the probability of a single packet going to multiple destinations reduces exponentially. This behavior (of an exponentially fewer packets going to larger flow-sets) reflects our observations of the captured packet trace.

C. Simulation Results

In the rest of this section, we describe simulation experiments using both our captured data and sythentically generated traces. We compare three algorithms:(1) SP-E2E which combines traditional shortest path routing with end-to-end RE; (2) RA-O is the optimal algorithm using Steiner Trees, and (3) our heuristic RA-H. aware routing. In each experiment, we calculate total network cost reduction using each method, and also compute the extra load induced on to network routers by using RA-H.

D. Evaluation using Real Trace Data

In this section, we evaluate in-network RE and the RA-H algorithm by re-playing our captured trace on to the SpringLink Tier-1 ISP topology (AS 1239). The SprintLink topology we use was generated using RocketFuel [18].

In our first experiment, we vary where within the SprintLink topology the VPN PoPs are located. In effect, we are simulating different geographic end-points for our hypothetical VPN

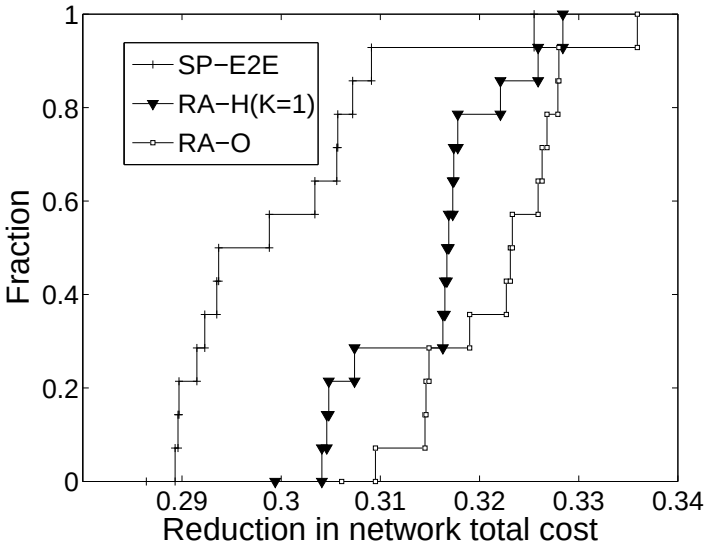


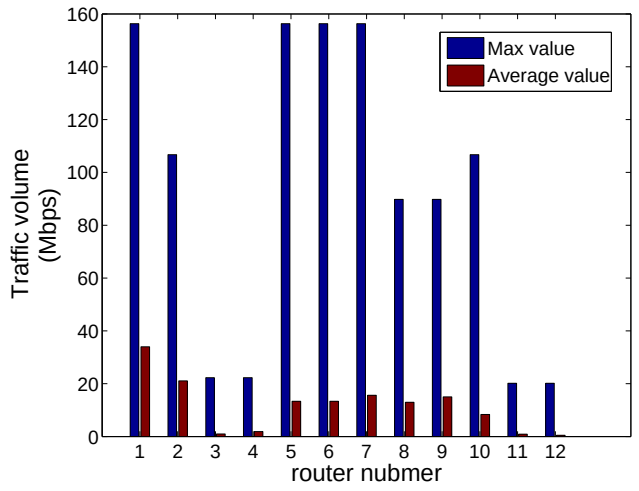
Fig. 3: Cost Reduction with different PoPs

while studying the benefits accured from RE. We ran with three different VPN configurations: star (two PoPs in the east, two in the west, and one in the center); partition (three PoPs in the east or west, remaining two west or east), and tree (one PoP in the west, four east). For each configuration, we ran with seven different runs with randomly chosen PoP locations constrained by the VPN configuration.

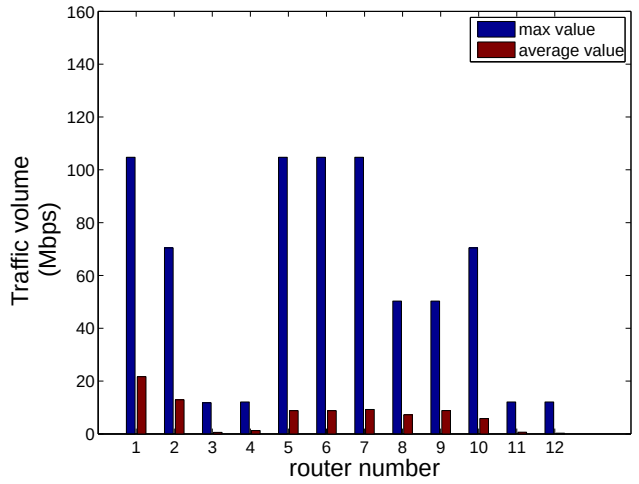
Fig 3 shows a CDF of reduction in total network cost compared to shortest path routing with RE for all 21 runs. Using end-to-end RE (SP-E2E in the figure) reduces total network cost $34 - 38\%$. RA-H with only a single overlay router provides a further (absolute) reduction of about 2%, and is within 1% of the optimal (RA-O).

We also analyzed how sensitive the reduction is to different data types and sources. Our data shows that the data center sources are much more amenable to RE (28-41% reduction) compared to the corporate campuses (15–25% reduction). Further, all ($> 99\%$) of inter-flow redundancy is also from the data center sources.

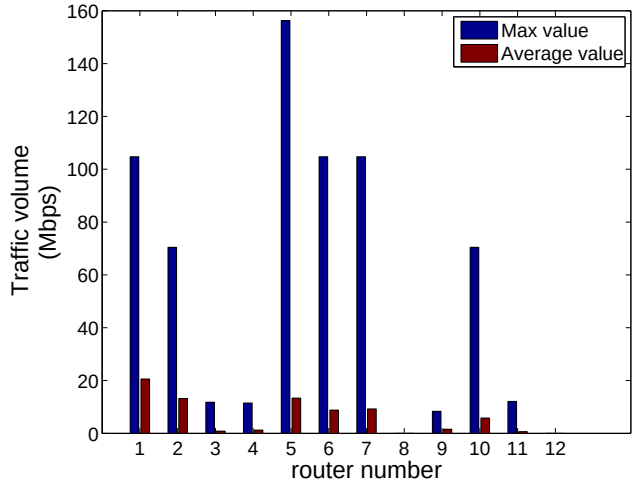
Figure 4 shows the peak and average load on each router while processing the real trace packets with and without RE. The x-axis is a router index, and refers to the same router across the histograms. The peak load is computed as the maximum of the incoming and outgoing packet rates. The results show that, as expected, SP-E2E reduces both the peak and the average transfer rate across all routers. Interestingly, even with the flow re-routing, which has the effect of concentrating packets onto the overlay routers, the load due to RA-H is lower than regular shortest path routing. This is because the RE ensures that less traffic is incident upon routers. The traffic concentration due to re-routing does not (at least in these examples) cause the packet rate to go beyond basic shortest path routing. This is an extremely encouraging and positive result that further underscores the feasibility of



(a) Shortest Path (No RE)



(b) Shortest Path with RE (SP-E2E)



(c) RA-H

Fig. 4: Router load (peak and mean) for different routing schemes

R_e \ r_e	0.001	0.01	0.1	0.2	0.35
0.001	.001, .001	.001, .002	.001, .027	.001, .055	.001, .100
0.01	.008, .008	.010, .011	.009, .034	.010, .064	.010, .110
0.1	.099, .099	.099, .100	.105, .127	.107, .161	.090, .198
0.2	.198, .198	.199, .200	.200, .228	.211, .259	.190, .294
0.35	.348, .348	.348, .349	.354, .378	.360, .441	.358, .470

TABLE III: Reduction in network cost as redundancy parameters are varied: Each pair shows reduction due to end-to-end RE (SP-E2E), and RA-H compared to Shortest Path (no-RE)

deploying RA-H.

E. Evaluation using Synthetic Traces

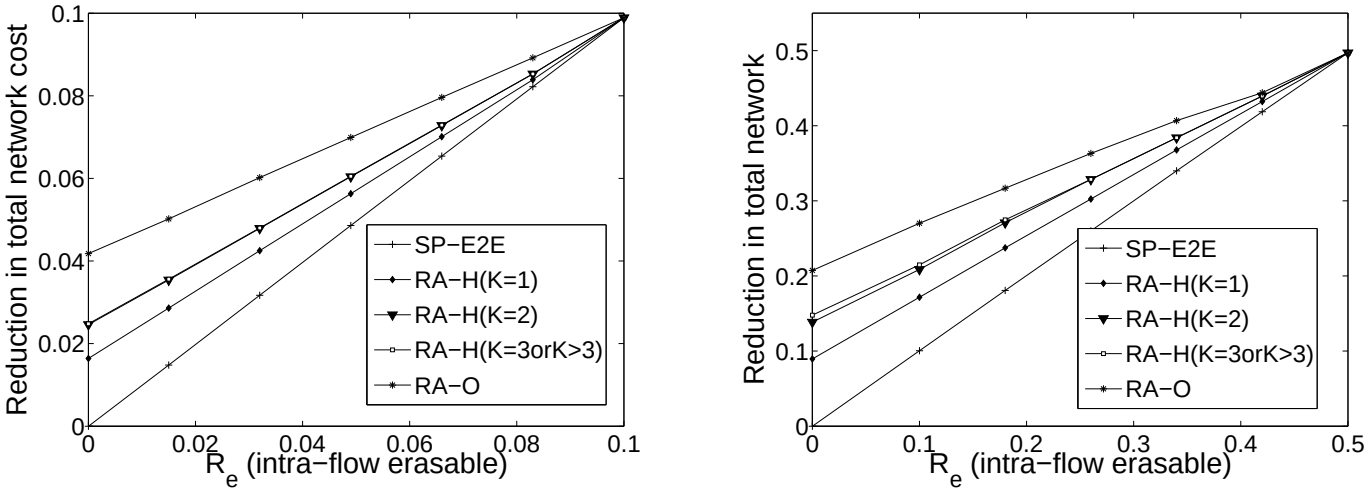
In the rest of this section, we use our generated synthetic traces using our model, and study the effects of varying different redundancy profile parameters and network topologies.

c) Different Redundancy Profiles: .

We once again use the RocketFuel SprintLink topology for our experiments with a five PoP VPN in the star configuraion (two pops east, two west, one center).

Figure 5 presents the reduction in network cost given different redundancy profiles. In each sub-figure, the overall fraction of eraseable packets (O_e) remains unchanged, and we compute the reduction from the three different algorithms as we vary the value of R_e (intra-flow eraseable) along the x-axis. Since O_e is fixed, the value of r_e (inter-flow eraseable) $(1-R_e)$ changes as well. The plot shows the reduction in network cost as a fraction of shortest-path (no RE) cost. The RA-H algorithm is remarkably stable, showing essentially linear performance gain as R_e approaches O_e. Also, almost all of RA-H's gains are realized using very few extra routers (1 or 2).

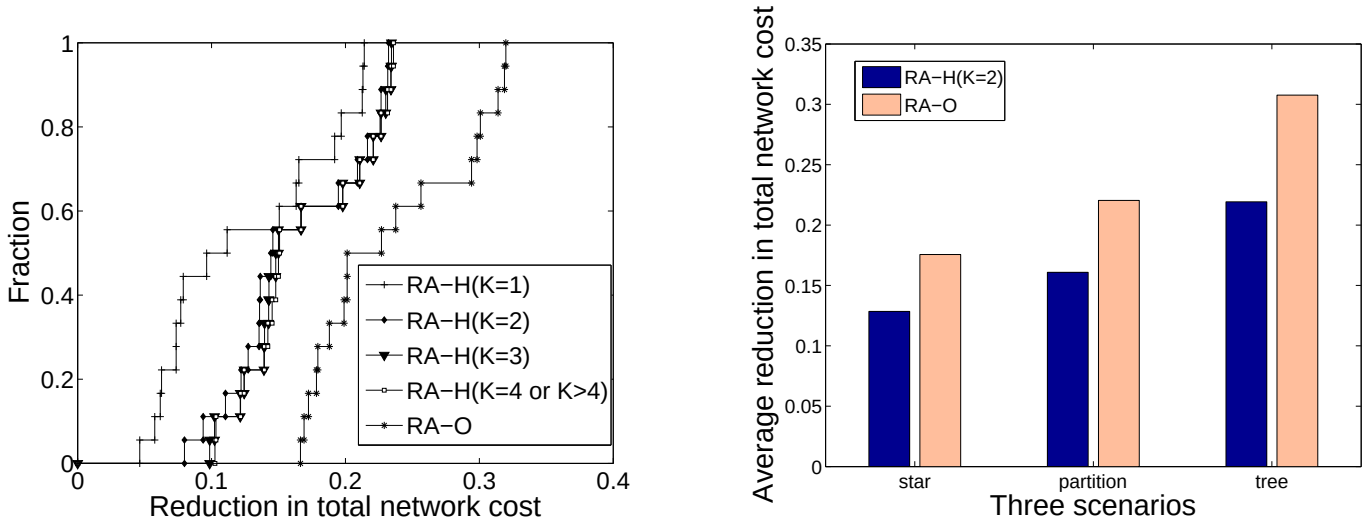
Table III shows the performance of SP-E2E and RA-H as the r_e and R_e parameters are varied independently. Each entry in the table is a pair that records the fraction reduction due to SP-E2E and RA-H (with $K = 3$) for a given value of R_e and r_e. The top rows (second number in pair) quantify the goodness of RA-H since the value of R_e is very low, meaning almost all of the redundancy is due to inter-flow duplicates. The table shows that RA-H can extract about $\frac{1}{3}$ of the maximum possible inter-flow benefit using only three overlay routers. Another way to quantify the goodness of RA-H is to consider the difference between the second and first number in each entry, which quantifies the benefit of RA-H without any input from the redundancy seen by SP-E2E. In all cases, both the absolute and the marginal benefit of RA-H increases as the value of r_e increases. Interestingly, note that the benefit from SP-E2E can sometimes *exceed* the value of R_e, e.g., when $r_e = 0.2$ and $R_e = 0.35$. This is because the duplicates are computed at the source (in this case, 35% of all source packets are intra-flow duplicates), but the cost is calculated across the entire network (and some heavy flows that have many duplicates can traverse long paths, reducing total network cost by more than 35%).



(a) Reduction in net. cost, $O_e = 0.5$

(b) Reduction in net. cost, $O_e = 0.1$

Fig. 5: Synthetic trace with varying redundancy parameters



(a) Cost Reduction CDF of different POP locations

(b) Average cost reduction with different scenarios

Fig. 6: Synthetic trace with varying PoP locations

d) Different PoP Locations: We repeated our experiments with different VPN configurations (star, partition, and tree) with different PoP locations with the synthetic traces.

For each VPN configuration, we conducted six experiments with different PoP locations as before. We generated the synthetic traces with parameter $r_e = O_e = 0.5$, which is the identical to the redundancy parametrization used in [7].

Figure 6(a) plots the CDF of cost reduction from 18 experiments. The results show that the impact of topology is significant. With the same synthetic trace, different topologies result in cost reduction ranging from 16.6% to 31.9% for RA-O, and from 10.3% to 23.6% for RA-H. RA-H offers 3.0% to 12.8% less cost reduction compared to RA-O.

Further, using RA-H, $96.2\% - 100\%$ of the cost reduction is offered by introducing the first two overlay nodes. Hence, in our experiments, a small number of overlay nodes was always sufficient to extract almost all of the gains from inter-flow redundancy.

Fig 6(b) breaks down the average cost reduction for the three configurations. We place a data center at the center of the star topologies. This causes potential detours to be relatively long, and results in comparitiavely lower network cost reduction. Conversely, in the tree topology, the data center is located at the root, and multiple flows can be merged using much shorter overlay paths, leading to a comparatively higher network cost reduction.

e) Different ISP Topologies: . Finally, we ran experiments on topologies other than the SprintLink topology used

in the other experiments. These topologies were also Tier-1 ISP topologies obtained using RocketFuel. In all cases, these ISPs had at least one PoP in North America. The figure shows network cost reduction with $O_e = r_e = 0.5$ using 5 PoPs placed randomly. The benefits from both RA-H and the optimal are relatively stable: RA-H benefits vary by less than 5% across topologies, whereas the optimal solution varies by about 7%.

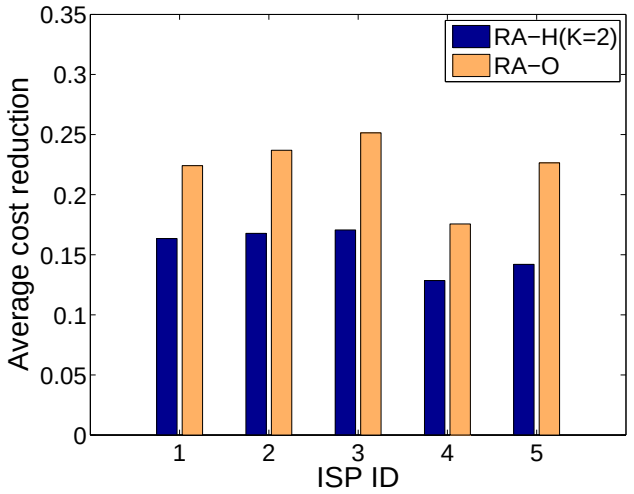


Fig. 7: Cost Reduction across multiple ISPs

VI. Conclusion

In this paper, we have discussed the problems associated with the deployment of in-network redundancy elimination. In particular, prior work requires per-packet re-routing and RE-aware routers to be deployed at each hop. We have argued that both assumptions are unteneable, and have developed a heuristic (RA-H) that can lead to practical deployment of in-network redundancy elimination.

Our evaluation is rooted on a large-scale measurement study. We use this study to motivate our problem and our solution approach. We use the study data to parametrize a simple traffic model, which we use to generate synthetic data to further evaluate our RA-H. Our evaluation is extensive, and considers both the benefit (in terms of network cost reduction) and the cost (in terms of router load) of RA-H. Our results show that RA-H can obtain much of benefit of an exponential-time optimal solution using only a minimal deployment of new hardware. Further, RA-H performance is stable across a wide-range of redundancy, topology, and deployment parameters.

Our current work assumes that the "redundancy profile", that captures the duplication among different flows, is available as an input to the RA-H algorithm. Naiive profiles require exponential time and space to construct and process; we have briefly discussed how such profiles can be scalably constructed. In our ongoing work, we are investigating methods for constructing these profiles on-the-fly, and re-structuring network paths as the redundancy profiles change.

References

[1] Netequalizer bandwidth shaper. http://www.netequalizer.com/.

[2] Packeteer wan optimization solutions. http://www.packeteer.com/.

[3] Peribit wan optimization. http://www.juniper.net/.

[4] Riverbed networks wan optimization. http://www.riverbed.com.

[5] Riverbed steelhead appliance performance brief: File transfer protocol. http://www.riverbed.com.

[6] Squid web proxy cache. http://www.squid-cache.org.

[7] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker. Packet caches on routers: the implications of universal redundant traffic elimination. In V. Bahl, D. Wetherall, S. Savage, and I. Stoica, editors, *SIGCOMM*, pages 219–230. ACM, 2008.

[8] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramachandran. Redundancy in network traffic: Findings and implications. *ACM SIGMETRICS*, Jun 2009.

[9] A. Anand, V. Sekar, and A. Akella. SmartRE:an architecture for coordinated network-wide redundancy elimination. *SIGCOMM*, Aug 2009.

[10] C. Arthur, A. Lehane, and D. Harle. Keeping order: Determining the effect of tcp packet reordering. In *ICNS'07, Proceedings of Third International Conference on Networking and Services*, 2007.

[11] E. Blanton and M. Allman. On making tcp more robust to packet reordering. *ACM Computer Communication Review*, 32:2002, 2002.

[12] S. L. Hakimi. Steiner problem in graphs and its implications, 1971.

[13] U. Manber and U. Manber. Finding similar files in a large file system. In *in Proceedings of the USENIX Winter 1994 Technical Conference*, pages 1–10, 1994.

[14] L. Nastansky, S. M. Selkow, and N. F. Stewart. Cost-minimal trees in directed acyclic graphs. 1974.

[15] M. Rabin. Fingerprinting by random polynomials. In *Harvard University, Technical Report*, pages TR–15–81, 1981.

[16] S. Ramanathan. Multicast tree generation in networks with asymmetric links. *IEEE/ACM Transactions on Networking*, 4:558–568, 1996.

[17] S. Schleimer, D. S. Wilkerson, and A. Aiken. Winnowing: Local algorithms for document fingerprinting. In *Proc. 2003 ACM SIGMOD Int. Conf. on Management of Data*, pages 76–85, San Diego, CA, Jun 2003.

[18] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring isp topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12(1):2–16, February 2004.

[19] N. T. Spring and D. Wetherall. A protocol-independent technique for eliminating redundant network traffic. In *In Proceedings of ACM SIGCOMM*, pages 87–95, 2000.

[20] L. Zosin and S. Khuller. On directed steiner trees. In *In 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 59–63, 2002.