# 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 transfer. Redundancy elimination (RE) techniques can be applied on a per-packet basis, and provide benefit regardless of application. While it is straightforward to apply RE on a per-link basis, network cost can be further reduced by applying RE 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 RE over every link. Also, to preserve end-to-end performance and control signaling cost, routes cannot be determined on a per packet basis.**

**We propose a redundancy-aware routing algorithm. Our approach can cope with limited resources (in terms of number of routers that can support RE ) and is feasible (not requiring for per-packet routing decisions). We evaluate our algorithm using detailed simulations, based on both synthetic traffic and trace captured from large enterprise networks. Unlike previous studies, our studies consider data from multiple sources. Our results show that a small number of RE equipped routers, coupled with our routing algorithm, are sufficient to achieve reduction in network load close to the unreachable upper bound.**

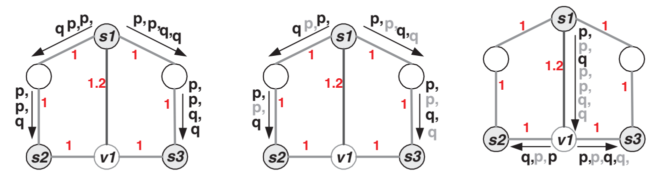

## I. Introduction

Identifying and eliminating duplicate bytes in subsequent packets reduces communication cost. Prior work has described techniques for efficiently storing byte-string signatures [1], identifying duplicate byte ranges in packets and reducing their transfer cost [2]; and evaluated the efficacy of these techniques in a network-wide setting [3].

These *redundancy elimination* (RE) techniques can be applied on a per-link basis. Indeed, the early motivation for [2] was to reduce congestion on a campus-ISP link. Other single-link deployments include commercial *WAN acceleration* products that eliminate redundancy over end-to-end (E2E) virtual links in enterprise VPNs [4], [5], [6], [7]. Results have shown that significant amount of traffic across enterprise sites can be eliminated if devices are installed in individual sites to offer E2E RE between pairs of sites [7].

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, 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 in both




(a) **SP** cost is 14. (b) **SP-E2E** cost is (8+6$d$). (c) **RA routing** cost is (6.4+9$d$).

Fig. 1: Benefits of E2E and network-wide RE. Original packets (in black) are of unit size, and the links are annotated with traversal cost. Using RE, transmission of a duplicate packet (shown in gray) incurs cost $d \times$ link cost where $0 < d < 1$.

original and current packet can be encoded using 12 bytes [2]. Thus, the cost to transmit a duplicate packet is usually much less than an entire packet, and is represented using $d$. The cost for transmitting packets over all links in the network reduces from 14 (in the default case with Shortest Path (SP) routing) to (8+6$d$) if duplicates are suppressed E2E without changing the default routes (SP-E2E). *Redundancy-Aware* (RA) routing enables further savings since the two original packets only have to be transmitted once. The total cost, with RA routing, reduces to (6.4+9$d$). In our example, packets are re-routed through router $v1$, but $v1$ does 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. Therefore interior nodes, *e.g.* $v1$, need to reconstruct compressed packets on-the-fly.

Network-wide RE with re-routing, or simply RA routing was originally proposed by Anand *et al.* [3]. In their setting, all routers in an ISP network cache packets and eliminate duplicates. RE can be applied beyond a single ISP, *e.g.* 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 E2E 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 over 40% of traffic from one VPN node to a pair destinations were duplicated.) 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 routing *every* packet independently with unique packets through the shortest paths and duplicate packets through the re-routing paths. While theoretically sound, implementing such a scheme would incur prohibitive control plane overhead. In addition, multiple paths between a source and destination pair lead to packet re-ordering. Performance of existing protocols, including TCP, would suffer unduly due to packet re-ordering [8]. The second problem is more pragmatic: existing work assumes that *every* router in an ISP or enterprise network will be capable to implement RE. 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. Further, if the traffic to be compressed is encrypted, *e.g.* VPN or secure web traffic, implementing the current method would require all the routers to note different encryption algorithms as well. This is not feasible in practice.

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, and only requires a small number of routers to be RE-capable. Our contributions are as follows:

- We formulate the problem of redundancy-aware routing under limited resources. Unsurprisingly, the optimization problem is NP-hard; we propose a polynomial-time heuristic for deriving RA routes.
- We present measurement data from a multi-site VPN and empirically show the existence of redundancy in traffic from a single source to multiple destinations. We capture traffic from a very large corporate network that has more than 25,000 active users. We use this trace and our traffic model to generate synthetic traces for 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 RA routing; in particular, we consider the extra load induced on different routers and links.

We organize the rest of the paper as follows: We begin with a description of prior and related work. Section III presents our problem formulation. Section IV describes 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 content is accessed by same or different users repeatedly. Many systems have explored this fact to eliminate redundant 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 to reduce bandwidth usage for access or ISP networks [9].

Recently a new class of systems have been developed to be application-level protocol independent and operate at packet-level to eliminate redundancy. Packet-level RE schemes, first proposed in [2], are becoming more popular as many vendors are developing WAN optimization solutions [4], [5], [6], [7].

The scheme in [2] is based on techniques for finding similar files in [10] using Rabin fingerprints [1]. For each packet, it first computes Rabin fingerprints of sliding windows of 64 contiguous bytes of the payload, and then selects a fraction 1/32 of them as representative ones. The fingerprints serve as pointers into portions of the packet payload which is used to find redundant content. There are variations regarding how representative fingerprints are chosen. The “winnowing” mechanism chooses local maxima or minima over each fixed size region in payload [11]. Our redundancy-aware routing algorithm is independent of any packet-level RE schemes, and our evaluation in Section V uses the original scheme in [2].

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 utilization [3], [12], [13]. This paper extends their work by proposing a more feasible routing algorithm with practical constraints such as only 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 [13]. Our real trace study further complements this study by demonstrating properties of multi-site enterprise traffic in a mesh.

## III. Redundancy-aware Routing

We first formalize the problem of RA routing in overlay networks. 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. We assume that an overlay network is constructed atop the physical topology. Existing shortest path (SP) routing (or whichever routing the ISP chooses) is used between two overlay routers 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 on top of a physical network $G'$ with $N$ end-nodes $\{s_1, \ldots, s_N\} \subset V$.

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 $N-1$ 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 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 [3]. 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 multiple destinations.

We define *intra-flow redundancy* of a flow as the volume of redundant packets within the flow. For a group of flows with the same source, we define their *inter-flow redundancy* as the overlap among all flows after intra-flow redundancy has been removed by E2E RE within each flow.

**Network Cost Calculation**. Our goal is to construct a minimum cost overlay network $G$ using up to $K$ overlay routers while using redundancy profile to re-route traffic. The way to build the overlay is to establish a tunnel 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.

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 $G'$ below. The latency $l_e$ over the edge $e$ is the sum of latency over physical network edges it is composed of. For every $e$, we define a binary variable $rt_{e,i,m}$ to indicate the occurrence of packet $p_{i,m}$ over that edge. 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.

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 [3], we call it the *footprint* of packet $p_{i,m}$ over edge $e$. 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 $p_{i,m}$. The cost of the overlay 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.

**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 sending one packet to several destinations. The Steiner tree problem has long known to be NP-hard [14], [15], [16].

**An Optimal Solution using Steiner Trees**. In fact, the complexity of RA 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. [17]. As every packet routed through the Steiner trees uses minimum resources, the total cost for all packets is therefore minimum. The optimal solution offers the upper bound in network cost reduction. However, as we mentioned in section I, this method is unfeasible and unachievable in practice.

We note that this optimal solution is a reformulation of the optimal presented in [3], 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 $K$. We assume that the traffic redundancy profile is also available. RA-H outputs the overlay routers and overlay routes for each source-destination pair.

According to our measurement results in Section V-A and reports in [18], intraflow redundancy is significant. Thus, in the greedy algorithm, we first eliminate all the intraflow redundancy by implementing E2E RE on all the source and destination nodes. Next, we show how to eliminate interflow redundancy.

We introduce the following notation: Let $\mathcal{R}_s$ be the traffic redundancy profile for flows from 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[1]. We refer to $(\{s \rightarrow d_1, s \rightarrow d_2\})$ 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 small number of subsets would be computed and used in RA-H based on the extent of redundancy in real traffic.

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 decode compressed packets as necessary, and forward (re-constructed) packets to their destination.

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 $\mathcal{R}_s$ in which there are flows that have not yet been re-routed. Let the $C$ be current flows being considered.
- Compute the reduction (if any) in network cost for re-routing $C$ through every router $r$ in the network $G'$, regardless of whether $r$ is already in the overlay $G$.
- 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 by promoting it the an overlay router. The flow-set $F$ is then labeled as *settled*.

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.

Notice that, for every flow-set, we limit the number of promoted router to one in order to reduce running time from exponential to polynomial and turns the solution from optimal to heuristic. While, promoting more than one router may reduce cost further for flow-sets containing more than two flows,

[1]Note that $\mathcal{R}$ is computed after the intraflow redundancy 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.

| 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% |

TABLE I: Intra-flow Redundancy in the Real Trace.

however in practice, the cost reduction from settling two flows is greater than settling three or more flows. Analysis of trace data in both [13] and Section V-A has shown that interflow redundancy among more than two flows is significantly smaller than that among any two flows. As a result, our algorithm usually settles the flow-sets with two flows before settling flow-set with three or more flows, and thus it is still possible to promote multiple routers to settle flow-sets with more than two flows. Our evaluation in Section V shows our algorithm gives a good approximation to the upper bound solution.

## 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 for 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. 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 480 Gigabytes, with more than 2.5 Billion packets captured.

We use the algorithm in [2] with 64 byte fingerprint window size and a FIFO cache size 5GB to analyze content redundancy. Table I shows intra-flow redundancy for each flow; per-source volume is reported as percentage of total traffic volume. We define *intra-flow redundancy ratio* (Intra-RR) as the ratio of redundant content over total volume of the flow. Table I shows 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 $f_{D_a,S_c}$.

Naturally, one can define *inter-flow redundancy ratio* (Inter-RR) of a flow-set as the ratio of redundant content over total volume of the flow set. However, we are conservative, and compute 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.

| 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% |

TABLE II: Inter-flow Redundancy in the Real Trace.

Our analysis shows significant intra-flow redundancy that E2E RE can remove. The remaining inter-flow redundancy is non-trivial, and motivates our work. We next evaluate how well RA-H eliminates inter-flow redundancy in comparison to the optimal, and quantify network effects of 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 consists of distinct packets and copies of distinct packets. Any packet must fall in to one of four mutually exclusive and exhaustive categories:

(1) **Inter-flow Unique** ($r_u$) consists of packets that appear for the first time at the source.

(2) **Inter-flow Erasable** ($r_e$) consists of copies of packets in category (1) but sent to different destinations.

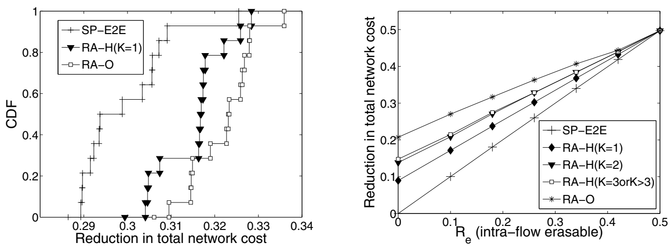
(3) **Intra-flow Unique** ($R_u$) consists of packets that appear for the first time in a flow.

(4) **Intra-flow Erasable** ($R_e$) consists of copies of packets in category (3) and sent to the same 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, E2E RE can eliminate all intra-flow erasable packets, whereas inter-flow erasable packets require in-network support.

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/erasable). 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 generate unique packets with probability $1 - O_e$ and erasable packets with probability $O_e$. For each unique packet, we chose its 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 erasable packet $p$, we pick a intra-flow unique packet (say $q$) uniformly at random and set $p$'s destination equal to that of $q$. For each inter-flow erasable packet, we again start with an 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 observation of captured trace.




(a) Cost reduction with different PoPs. (b) Reduction in net. cost, $O_e = 0.5$.

Fig. 2: Experiment results.

### C. Simulation Results

We now describe simulation experiments using both captured and synthetically generated traces. We compare three algorithms:(1) SP-E2E which combines traditional shortest path routing with E2E RE; (2) RA-O, the optimal algorithm using Steiner Trees, and (3) RA-H, our heuristic. In each experiment, we calculate total network cost reduction using each algorithm, and compute the extra load induced on routers.

We evaluate our RA-H against SP-E2E and RA-O by replaying our captured trace on to the SprintLink Tier-1 ISP topology (AS 1239) generated using RocketFuel [19]. 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 while studying the benefits accrued 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 conduct five different runs with randomly chosen PoP locations constrained by the VPN configuration.

Fig 2 (a) shows the CDF of reduction in total network cost for the three algorithms for all 15 runs. SP-E2E reduces total network cost $28.6 - 32.5\%$. RA-H with only a single overlay router provides a further (absolute) reduction upto $2.3\%$, and is within $1\%$ of the optimal (RA-O).

We also analyze 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.

We examine the peak and average load on each router while processing real trace packets with different algorithms. Our results show that, as expected, SP-E2E reduces both peak and average transfer rate across all routers comparing with implementing shortest path without RE. Further, even with flow re-routing, which has the effect of concentrating traffic onto overlay routers, peak load of all the routers with RA-H is the same as SP-E2E. As the flows switch, the average traffic volume of some routers increases. Interestingly, among all the routers, there exists only one router whose average volume is larger than the amount in the shortest path without RE case, and the increasing percentage is $22.9\%$. This is a positive result that further underscores the feasibility of deploying RA-H.

| | $r_e$ | | | | |
|---|---|---|---|---|---|
| $R_e$ | 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 SP-E2E and RA-H($K = 3$) compared to Shortest Path (no-RE).

### D. 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.

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

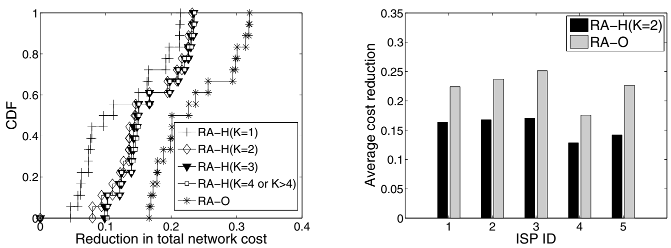
**Different Redundancy Profiles**. We once again use the RocketFuel SprintLink topology with a five PoP VPN in the star configuration.

Figure 2(b) presents network cost reduction given different redundancy profiles. The overall fraction of erasable packets ($O_e = 0.5$) remains unchanged, and we compute the reduction from the three different algorithms as we increase the value of $R_e$ (intra-flow erasable) along the $x$-axis. Since $O_e$ is fixed, the value of $r_e$ (inter-flow erasable) ($O_e - R_e$) reduces. 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). We conduct the experiments with various values of $O_e$ and notice similar trend [20].

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%).

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

For each VPN configuration, we conduct six experiments by




(a) Cost reduction of diff. locations. (b) Cost reduction across mult. ISPs.

Fig. 3: Experiment results with synthetic traces.

varying PoP locations as before. We generate synthetic traces with parameter $r_e = O_e = 0.5$.

Figure 3(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 cost reduction is offered by introducing the first two overlay nodes. Hence, a small number of overlay nodes are sufficient to extract almost all of the gains from inter-flow redundancy in our experiments.

**Different ISP Topologies**. Finally, we run experiments on Tier-1 ISP topologies obtained using RocketFuel, other than the SprintLink topology. In all cases, each ISP has at least one PoP in North America. Figure 3(b) shows network cost reduction with $O_e = r_e = 0.5$ using 5 PoPs placed randomly. The benefits from both RA-H and RA-O are relatively stable: RA-H benefits vary by less than 5% across topologies, whereas the optimal RA-O benefits vary by about 7%.

**Long Term Traffic Variation**. Practical in-network RE solutions rely on traffic redundancy profiles. These profiles are collected periodically. As profile varies, the location of RE routers and routes will change. Results from another set of traffic traces captured at the same set of corporate VPN sites on Aug 6, 2009 indicate that less than 2% variation in inter-flow redundancy variation. This variation does not lead to any change in overlay routers or routes in our setup. This gives some indication of the stability of selection of the overlay RE routers. We plan to study time variation of redundancy profile with more data sets.

## VI. Conclusion

In this paper, we have discussed the issues associated with deployment of in-network redundancy elimination (RE). 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 untenable, 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 extensive evaluation 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. Naive 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.

## VII. Acknowledgments

This work is partially supported by NSF grant CNS-0626618 and NSF grant CNS-0917078.

## References

[1] M. Rabin, "Fingerprinting by random polynomials," in *Harvard University, Technical Report*, 1981.
[2] N. Spring and D. Wetherall, "A protocol-independent technique for eliminating redundant network traffic," in *SIGCOMM*, 2000.
[3] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker, "Packet caches on routers: the implications of universal redundant traffic elimination.," in *SIGCOMM*, 2008.
[4] "Netequalizer bandwidth shaper." http://www.netequalizer.com/.
[5] "Packeteer wan optimization solutions." http://www.packeteer.com/.
[6] "Peribit wan optimization." http://www.juniper.net/.
[7] "Riverbed steelhead appliance performance brief: File transfer protocol." http://www.riverbed.com.
[8] C. Arthur, A. Lehane, and D. Harle, "Keeping order: Determining the effect of tcp packet reordering," in *ICNS*, 2007.
[9] "Squid web proxy cache." http://www.squid-cache.org.
[10] U. Manber, "Finding similar files in a large file system," in *USENIX*, 1994.
[11] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local algorithms for document fingerprinting," in *SIGMOD*, (San Diego, CA), 2003.
[12] A. Anand, V. Sekar, and A. Akella, "SmartRE:an architecture for coordinated network-wide redundancy elimination," *SIGCOMM*, 2009.
[13] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramachandran, "Redundancy in network traffic: Findings and implications," *SIGMETRICS*, 2009.
[14] L. Nastansky, S. M. Selkow, and N. F. Stewart, "Cost-minimal trees in directed acyclic graphs," *Mathematical Methods of Operations Research*.
[15] S. Ramanathan, "Multicast tree generation in networks with asymmetric links," *IEEE/ACM Trans. on Networking*, 1996.
[16] L. Zosin and S. Khuller, "On directed steiner trees," in *Annual ACM-SIAM Symposium on Discrete Algorithms*, 2002.
[17] S. Hakimi, "Steiner problem in graphs and its applications," 1971.
[18] M. Roughan, M. Thorup, and Y. Zhang, "Performance of estimated traffic matrices in traffic engineering," 2003.
[19] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," in *SIGCOMM*, 2002.
[20] Y. Song, K. Guo, and L. Gao, "Redundancy-aware routing with limited resources," http://rio.ecs.umass.edu/˜ysong/redundancy.pdf.