Avoiding Oscillations due to Intelligent Route Control Systems

Ruomei Gao, Constantine Dovrolis, Ellen W. Zegura
gaorm, dovrolis, ewz@cc.gatech.edu
Networking and Telecommunications Group, Georgia Tech, Atlanta 30332

***Abstract*— Intelligent Route Control (IRC) systems are increasingly deployed in multihomed networks. IRC systems aim to optimize the cost and performance of outgoing traffic, based on measurement-driven dynamic path switching techniques. In this paper, we first show that IRC systems can introduce sustained traffic oscillations, causing significant performance degradation instead of improvement. This happens, first, when IRC systems do not take into account the *self-load effect*, i.e., when they ignore that the performance of a path can change after additional traffic is switched to that path. Second, oscillations can take place when different IRC systems get synchronized due to significant overlap of their measurement time windows. We then propose measurement methodologies and path switching algorithms that can effectively deal with the previous two issues. The proposed IRC techniques use available bandwidth estimation to avoid the self-load effect, and they introduce a random component in the path switching decision or time scale. We evaluate the proposed techniques under diverse traffic conditions. When the background traffic is stationary, IRC systems should switch paths conservatively, only upon major traffic fluctuations. With nonstationary background traffic and congestion periods that last for a time scale T_w, IRC systems improve performance only if they can detect congestion and switch paths much faster than T_w; otherwise, they cause oscillations and hurt performance. We also show that the gradual deployment of randomized IRC systems, in the presence of traffic from deterministic IRC systems, can play a stabilizing role and benefits early adopters.**

Keywords: Multihoming, Routing, Stability, Synchronization, Network Measurements.

I. Introduction

Multihoming is the connection of a stub network to more than one Internet provider [1]. Networks that focus on reliability and availability have been using multihoming as a form of redundancy for years. The use of multihoming has seen a dramatic increase in the last few years. The widespread proliferation of multihoming is due to several reasons. First, as more and more enterprises rely heavily on the Internet for their transactions, reliability and availability are of primary importance. Second, multihoming is often used to drive down the cost of Internet access. This is the case when the multihomed network can use a lowest-cost ISP for bulk traffic and a higher-cost but better ISP for performance-sensitive traffic.

Multihoming capabilities have expanded tremendously with the development of *Intelligent Route Control* (IRC) products. IRC systems allow a stub network to automatically switch some of the egress traffic (typically in the granularity of prefixes) from one provider to another, driven by cost and/or performance considerations. A number of vendors have developed such systems (for a representative but incomplete list see [2], [3], [4], [5], [6], [7], [8], [9]). Even though most commercial multihomed-IRC systems do not expose deep technical information about their internal operation, one of them is described with significant detail in a research publication [10]. Another good description and evaluation of an operational multihoming-IRC system is given in [11]. These two publications, as well as several white papers and high-level descriptions from vendors, allowed us to understand the key features of existing IRC systems.

In the research domain, IRC systems have become the subject of thorough investigation only recently [12], [13], [14], [15], [16]. An experimental study, based on measurements from the Akamai content distribution network, showed that multihoming can lead to significant benefits in terms of both reliability and performance for both ingress and egress traffic [12]. They also showed that up to four providers are typically enough to gain the full benefit of multihoming. Another experimental work that demonstrated similar benefits is reported in [16]. In a more theoretical thread, the authors of [14] designed IRC algorithms that can optimize cost and performance for multihomed networks. [14] also examined the equilibrium performance of competing IRC systems through simulations, arguing that a multihomed network can improve its own performance without adversely affecting other users. Two interesting pricing problems related to multihoming, namely the optimal set of ISPs that a network should subscribe to and how ISPs can react to that optimal subscription, have been recently investigated in [15]. Experimental comparisons between IRC and routing overlays have been described in [13], [16]. The measurement-based comparison of [13] suggests that IRC systems may be capable of offering almost the same performance as routing overlays, but in a much simpler and cost-effective way. An investigation of the stability of IRC systems, focusing on routing overlays, has recently appeared in [17]. That study is similar with our work, as it also shows that IRC systems can cause oscillations and stability, but it does not move in the direction of solving the problem.

In this paper, we consider a multihomed destination ("sink") network D with m ingress links. D receives traffic from several source networks, including the multihomed networks

This work was supported by the National Science Foundation (CAREER CNS-0347374 and CNS-0519756) and by a URP fund from Cisco Systems Inc.

$S_1, S_2, \ldots S_n$. The latter deploy IRC systems and they can reach D through some or all of its ingress links. We assume that a significant fraction of the total traffic destined to D originates from these IRC-capable networks. Even though this is not the case today, it is certainly a plausible scenario for the near future. Each IRC system uses measurements to monitor the performance of the paths to D, and switches its traffic towards D to the path that offers the required performance level. The performance metrics that we focus on are loss rate and path switching frequency. An important point about our model is that the network measurements are *not* instantaneous; instead, as it always happens in practice, they take place over a time window and they consist of several samples. The IRC model is described in more detail in Section II.

We first show that IRC systems can cause sustained traffic oscillations (Section III). Interestingly, as a result of these oscillations, IRC systems can lead to a significant performance degradation, instead of improvement. We then identify two key factors that generate such oscillations. First, if IRC systems do not take into account the *self-load effect*, meaning that the performance of a path can drop significantly after we switch additional traffic to it, then IRC traffic can keep switching back and forth between the two paths. Second, oscillations can take place when different IRC systems get synchronized because there is significant overlap in their measurement time windows. When such synchronization occurs, independent IRC systems start behaving as a "herd", observing the same performance difference between two or more paths and making the same sequence of path changes.

In the second part of the paper, we propose measurement methodologies and path switching algorithms that can avoid the previous two oscillation factors (Section IV). First, we propose the use of available bandwidth measurements to avoid the self-load effect. Second, we introduce a random component in the path switching decision, or in the switching time scale, to avoid synchronization. We evaluate the proposed IRC techniques with simulations under diverse traffic conditions. As expected, the performance of IRC systems is intimately related to the variability of the background traffic. When the background traffic is stationary, IRC systems should switch paths conservatively, only upon major traffic fluctuations (Section V). With nonstationary background traffic and congestion periods that last for a time scale T_w, IRC systems improve performance only if they can detect congestion and switch paths much faster than T_w; otherwise, they cause oscillations and hurt performance (Section VI). Finally, we investigate the gradual deployment of the proposed randomized IRC systems in the presence of traffic from deterministic IRC systems (Section VII). Interestingly, randomized IRC systems play a stabilizing role, reducing the overall loss rate. Furthermore, they are beneficial for the networks that adopt them, as the latter observe better performance than networks using deterministic IRC.

II. IRC MODEL

In this section, we describe the model of an IRC system, as well as the network and traffic environment in which we assume that IRC systems operate.

A. Network model

We consider a single destination network D that is multihomed, as well as a set of source networks that send traffic to D. We refer to the source networks that either have only one egress link, or that always use a specific egress link to reach D, as *Default Route Control* (DRC) sources. On the other hand, multihomed networks that can dynamically switch between different egress paths to reach D are referred to as *Intelligent Route Control* (IRC) sources.

We assume that the underlying Internet routes to D are stable relative to the time scales of IRC path switching. Hence, the path from a DRC source to the destination network D will traverse a particular ingress link of D, as determined by BGP. In contrast, the path from an IRC source to the destination network D will be determined by both BGP *and* by the choice of egress link at the source. When two sources use the same ingress link at D, their paths intersect, minimally at the ingress link, and possibly further upstream as well. We model this path interaction as occurring only at the ingress link, rather than developing a more complex model for upstream path interaction. The justification for this assumption is twofold. First, to an approximation, we can consolidate the interaction over multiple upstream links as interaction over a single link. Second, for many enterprise stub networks today it is their access link to the Internet that is often the end-to-end path bottleneck.

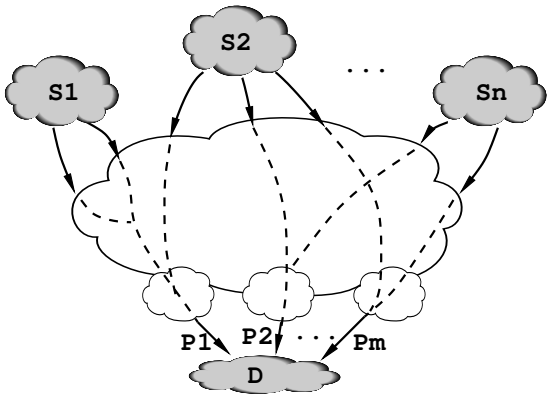


Fig. 1. Network model.

As illustrated in Figure 1, we assume that n IRC sources $S_1, S_2, ..., S_n$ send traffic to destination D through m ingress links (DRC sources are not shown). The figure illustrates that, in general, some IRC sources may not be able to access all ingress links of D (source S_n has this characteristic, for instance). Also, some IRC sources may always traverse the same destination ingress link, regardless of the selected source egress link, because of how the underlying Internet routes merge before reaching D (source S_1 has this characteristic,

for instance). From this point on, however, we assume that all IRC sources can traverse any of the m ingress links at D.

B. Traffic model

We use the term *flow* to refer to the aggregation of all traffic from a single source network to D. When a flow's path is determined by IRC, the flow is an *IRC flow*; otherwise it is a *DRC flow*. We assume that network D receives a significant fraction of traffic from IRC flows. This is a plausible assumption for the near future. We also assume that the *statistical characteristics* (including the average rate) of a flow remain constant over the time scales of interest. We do consider traffic burstiness, however, assuming that each flow follows the Fractional Gaussian Noise (FGN) model. Further, we assume that a flow's average rate does not change in response to changes in the path characteristics (e.g., congestion). This is a reasonable assumption as long as the arrival rate and size of new connections at the corresponding source network do not depend on network conditions (i.e., exogenous connection arrivals).

As previously mentioned, we model each traffic flow as an FGN fluid process [18]. The FGN process is self-similar and long-range dependent, and therefore it is considered an appropriate model for aggregate Internet traffic, especially in larger time scales where packet-level effects can be ignored. An FGN fluid process is determined by three parameters: the Hurst parameter that controls the degree of self-similarity, the average rate, and the variance over a given time scale. In the following, the Hurst parameter is set to 0.7, which is consistent with earlier measurement results [18]. For normalization purposes, we use the same coefficient of variation (CoV) for all flows. To pick a realistic CoV value, we fitted the FGN model to packet traces from an OC-3 university access link, at the time scale of 100 msec. The CoV in those traces varies between 0.09 to 0.16, and so we set the CoV of each flow so that the aggregate traffic at the ingress links of D has a CoV that is around 0.1.

Regarding the average rate of IRC flows, we use two distributions: a constant value for all flows and the Zipf distribution. We expect that these two extreme distributions will give us the right insight on how IRC performance depends on the flow homogeneity. DRC flows have the same average rate regardless of the IRC flow rate distribution.

C. IRC processes

We model an IRC system as performing periodically a five-stage process. The five stages are: idle, measurement, performance estimation, routing decision, and path switching (see Figure 2). T_r denotes the *routing period*, i.e., the time to complete a cycle through all five stages. An IRC system starts the cycle with the idle stage, which is optional. Then, in the measurement stage, the system collects performance samples for all candidate paths, using active probing or passive monitoring. Next, for each candidate path, the IRC system uses the previous measurements to estimate the performance of each path. In the routing decision stage, the IRC system determines whether it should switch to a different path, based on the specified performance objectives. Finally, if needed, the IRC system reroutes its outgoing traffic towards D to the chosen egress link. That link will be used at least during the next routing period.

We assume that the performance estimation, routing decision, and path switching stages can be executed instantaneously, as they only involve simple calculations and local routing table changes. Thus, from a timing point of view, the routing period T_r consists of the idle period and the measurement period T_m. For faster response to network congestion, the idle period can be minimized or even avoided. Hence, we assume that the routing period is as short as possible, only bounded by the length of the measurement period, i.e., $T_r = T_m$. In the following simulations, we set the routing period to T_r=1 second. Even though this is a short routing period compared to existing IRC systems, we believe that the requirements of interactive and transaction-based applications will gradually push network operators to reduce T_r as much as possible.

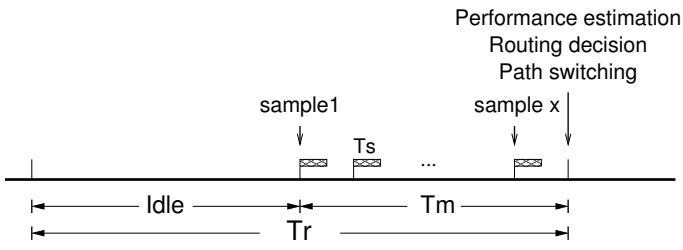


Fig. 2. The timeline of an IRC routing period.

D. Measurement process

A central component of an IRC system is the measurement process. In earlier work on multihoming and IRC, network measurement has been modeled as simple meter reading, i.e., an instantaneous and accurate action. That model however does not capture important characteristics of real network measurement, which can affect the stability and performance of IRC systems. Specifically, in practice network measurements take time, they are subject to estimation errors, and they often rely on sampling rather than continuous-time monitoring.

In our model, as illustrated in Figure 2, we assume that the performance of each path is estimated through periodic active probing. Each probing event collects a sample of the monitored path's performance. In a realistic environment with bursty network traffic, multiple samples are necessary for reasonably accurate estimation. Also, we need to consider that each probing event takes some time T_s; for instance, the time between sending a probing packet and receiving the corresponding ICMP response or acknowledgment. The path performance is estimated at the end of the measurement period, averaging the collected samples. In the following simulations the measurement period T_m consists of 10 samples, with a sampling period of 100msec.

E. Performance estimation

We consider three end-to-end metrics for evaluating the performance of a network path: queueing delay, loss rate, and available bandwidth. The queueing delay of a path is the difference between the measured Round-Trip Time (RTT) and the minimum observed RTT; the latter is typically due to propagation and transmission delays. The loss rate of a path is the fraction of lost probing packets. Finally, the available bandwidth of a path is defined as the minimum residual capacity among all links in the path. Most IRC systems today use the first two metrics, or straightforward variations of these metrics. Measurement techniques for available bandwidth have been developed only recently and it seems that they are not used by commercial IRC systems yet [19].

In the following results, we do not simulate the measurement process with individual probing packets. Instead, the previous three metrics are estimated as follows.

First, the simulated queueing delay d at an ingress link of capacity c is given by the following non-decreasing function of the instantaneous offered load r:

$$d = \begin{cases} \frac{1}{c-r}, & (r \leq c - \epsilon) \\ d_{max}, & (r > c - \epsilon) \end{cases} \quad (1)$$

where ϵ is a small positive constant. Note that the upper bound d_{max} models a finite buffer size ($d_{max} = 1/\epsilon$ for continuity).

Second, the simulated loss rate l at an ingress link of capacity c is determined by the fluid model:

$$l = \begin{cases} \frac{r-c}{r}, & (r \geq c) \\ 0, & (r < c) \end{cases} \quad (2)$$

In other words, we assume that the link does not drop traffic unless it is saturated.

Third, the simulated available bandwidth A of an ingress link is given by

$$A = \begin{cases} c - r, & (r \leq c) \\ 0, & (r > c) \end{cases} \quad (3)$$

Note that when comparing the currently used path p with another path p' in terms of available bandwidth, an IRC system has to consider the offered load r_f of its flow. Specifically, p' is better than p only if $A_{p'} > A_p + r_f$.

F. Routing decision and path switching

How does an IRC system compare paths and determine whether to perform path switching? We consider two cases: first, that the IRC system can only measure delay and loss rate (the most common case today), and second, that it can also measure available bandwidth.

When measuring only queueing delay and loss rate, we consider the latter as more important in terms of network performance. So, the path with the lowest loss rate is considered the best. This is consistent with typical Service Level Agreements today, which consider losses as more detrimental than queueing delay. If there are several paths with the same loss rate (for instance, l=0), the best path is the path with the minimum queueing delay. When the IRC system can also measure available bandwidth, the best path is that with the maximum available bandwidth. Note that if a path has some available bandwidth ($A > 0$), then its loss rate l is zero (Equation 2). If all paths are saturated (i.e., A=0), then the best path is chosen based on loss rate and/or queueing delay.

Taking into account that measurements are error-prone, we consider the performance of two paths as equal if the corresponding performance metrics are close, say within 10%. Specifically, if A and B are measurements over two paths p and p' respectively, we consider that p is better than p' if $0.9A > 1.1B$.

Knowing that there exists a better path than the currently used path does *not* mean that the IRC system should switch to the former. Specifically, we consider two path switching policies: *choose-best* and *choose-good*. With the former, the IRC system always switches to the best path. With the latter, the IRC system switches to the better path only if the current path is congested, i.e., if the loss rate is larger than zero.

With choose-best, an IRC system may switch paths even when there is no congestion in the current path, and so the path switching frequency increases. On the other hand, choose-best provides IRC traffic with a larger safety margin to random traffic variations and measurement errors. For this reason, we focus on the choose-best policy. The main conclusions of our study, however, are also valid for the choose-good policy when the ingress links of D are not overprovisioned.

III. IRC-induced traffic oscillations

In this Section, we discuss two conditions under which IRC systems can cause traffic oscillations. The first problem appears when using network measurements and performance metrics that do not take into account the load of the switched traffic ("self-load effect"). The second problem is the synchronization of different IRC systems when their measurement time windows overlap significantly.

A. Self-load effect

As explained in Section II, the IRC routing decisions are based on estimates derived from path measurements. The problem, however, is that the performance of a path p can be significantly affected after the IRC system switches some traffic to or from that path. For example, consider an IRC system and two candidate paths p_1 and p_2. Initially, the IRC system routes its traffic over path p_1, and the measured loss rates are 1% for p_1 and zero for p_2. This does not necessarily mean that p_2 is better. After the IRC system switches its traffic to p_2, the loss rate at the latter can become larger than 1%, due to the additional load that the IRC traffic imposes, and the loss rate at p_1 can become zero. Hence, the IRC system can start oscillating between the two paths. The same scenario can take place when the path switching decisions are based on queueing delay measurements. Note that it is not possible to predict the loss rate or queueing delay at a path after a load shift without an accurate characterization of the queueing behavior in that path, and such a characterization is quite difficult in practice.

With available bandwidth measurements, on the other hand, the previous problem can be avoided. The reason is that the available bandwidth shows how much additional traffic a path can carry before it is congested. For example, if p_1 has available bandwidth A_1=2Mbps and it carries an IRC flow of 5Mbps, while path p_2 has available bandwidth A_2=4Mbps, then the IRC system should prefer p_1 because if the flow is switched to p_2 it will certainly experience congestion.

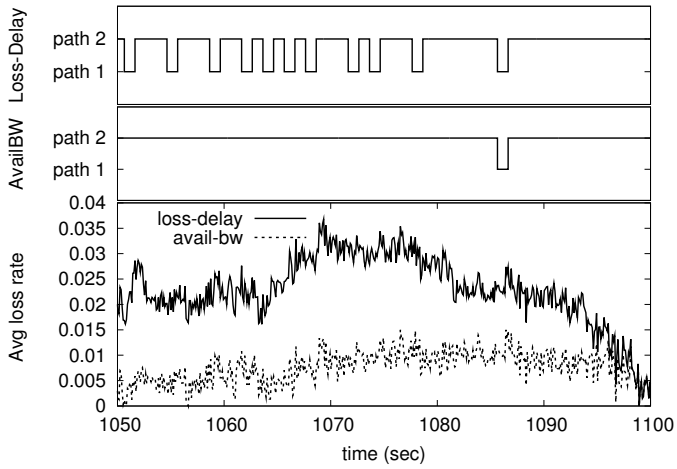


Fig. 3. Path switching events (top and middle graphs) and loss rate (bottom graph) when using delay and loss rate versus available bandwidth measurements.

Figure 3 shows the results of two simulations with the same configuration, except that one uses the loss-delay metric while the other uses the available bandwidth metric. Both experiments have four flows, one IRC flow and three DRC flows, and two identical ingress links at the destination network D. The four flows have the same average rate, and the aggregate capacity of the ingress links is equal to twice the total load of the four flows. The experiments start with two flows on each link. In Figure 3, the top two plots show the path the IRC flow uses as a function of time. When the IRC flow switches from one path to the other, it is shown as a downward or upward step. The number of downward and upward steps is the number of switching events of the IRC flow during that time period. The bottom plot of Figure 3 shows the running average of the loss rate of the IRC flow, over a moving window of 30 seconds.

The path switching plots clearly show that when the IRC flow uses the delay-loss metric it experiences many more path switching events than using the available bandwidth metric. Also, the loss rate that the IRC flow experiences is significantly reduced when using the available bandwidth metric, because the flow stays at the best possible path.

B. Synchronization of IRC systems

Another issue with IRC systems is that different IRC flows can get synchronized, oscillating between two or more paths. The fundamental cause for the synchronization is the possible overlap of the measurement time windows of different IRC systems. To understand this effect, consider a simple example with two identical ingress links and two identical IRC flows, f_1 and f_2. To avoid the self-load problem, assume that the flows use the available bandwidth metric. Also, suppose that both IRC flows have equal routing and measurement periods with $T_r = T_m$ and that they take 10 available bandwidth samples in each measurement period. Let us assume that the timing between the two flows is such that the routing decision of f_1 occurs one sample period earlier than that of f_2. f_1 and f_2 start at the same path p_1, and after the first routing period f_1 detects greater available bandwidth on p_2 and switches to that path. At that point, f_2 has already collected nine out of its 10 samples, and even though the available bandwidth in p_2 is now equal to that in p_1, it may also estimate that the *average* available bandwidth, across all 10 samples, is larger in p_2. So, f_2 will switch to p_2, where it will overlap with f_1 for 80% of the next measurement period. In the next routing period, f_1 and f_2 will move back to p_1 in the same fashion, hence producing a persistent oscillation between the two paths.

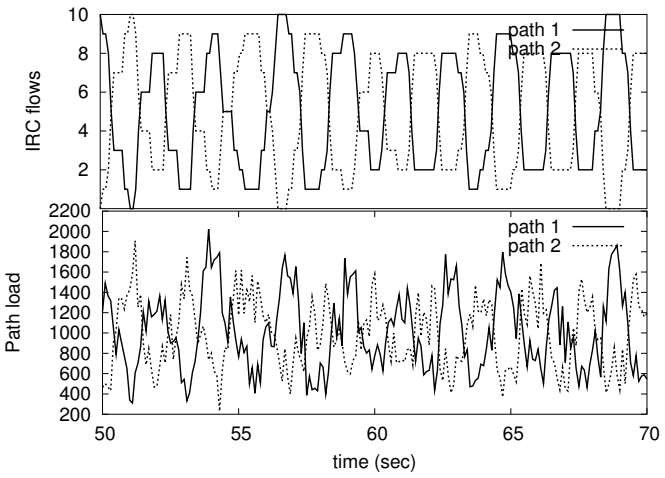


Fig. 4. Oscillations due to synchronization of different IRC systems.

In practice, even a lower degree of overlap between the measurement periods of IRC systems can still cause synchronization. In Figure 4, we show the type of oscillation described above but under a less rigid configuration. In this simulation, the destination has two ingress links. Initially, there are 10 IRC flows and 10 DRC flows on each link. All flows have the same average rate and equal routing/measurement periods $T_r = T_m$=1sec. The start times of the IRC flows (and thus their route decision events) are uniformly distributed over the length of a measurement period (one second). The graphs show the number of IRC flows on each path (top) and the traffic load (bottom) over time. Note the persistent path and traffic oscillations, as a result of IRC synchronization. The oscillation period is two seconds, which agrees with the fact that IRC flows perform a routing decision in every second.

One may think that such oscillations can be avoided if IRC systems use the last measured sample, instead of an average

across all samples, to estimate the performance of a path. This is not practical however, given that the performance of a path can vary significantly with time and the measurements are prone to errors. Consequently, the measurement period has to include several samples, and this implies that the measurements windows of different IRC systems can overlap in time causing synchronization. In the next section, we propose several path switching algorithms that can avoid synchronization through limited randomization.

IV. Randomized IRC algorithms

The previous section identified two problems with IRC systems: the self-load effect, which can cause oscillations even with a single IRC flow, and the synchronization of different IRC flows due to a significant overlap in their measurement periods. The self-load effect can be avoided if the IRC system uses available bandwidth measurements. In this section, we focus on path switching algorithms that can avoid the synchronization problem.

In general, synchronization among a set of autonomous agents can be avoided with the introduction of a certain degree of randomness in the actions of these agents. In the context of IRC systems, this randomization can take several forms. First, we can add randomization in the path selection itself. This is not a good option, however, if there are only two or three paths to choose from. Second, we can add randomization in the path selection timing, i.e., in the length of the routing period T_r. Third, we can add randomization in the path switching decision, i.e., on whether the path switching should be performed or not.

A. Deterministic Path Switching (DPS)

DPS is the basic path switching algorithm that we described in Section II. It does not add any randomization in the path switching decision and it uses a fixed routing period T_r. As shown in Section III, DPS can lead to persistent oscillations when two or more IRC systems get synchronized. The following four algorithms are variations of DPS that include some form of randomization.

B. Fixed Switching Probability (FSP)

FSP switches to the best path with a probability P that we refer to as the *switching probability*. P controls the responsiveness of IRC flows to performance changes. When $P = 0$, the IRC flow behaves just like a DRC flow (static routing) and it does not respond to measurements. When $P = 1$, the IRC flow behaves as a DPS flow and it is susceptible to synchronization.

C. Adaptive Switching Probability (ASP)

ASP is a variation of FSP in which P adapts to the network conditions. The intuition is that P should be large when there is a major performance difference between the current path and the best path, while P should be much lower when the current path is almost as good as the best path. For simplicity, ASP only uses two P values: P_{hi} and P_{lo}. The algorithm uses P_{lo} when the difference between the current path and the best measured path is below a certain threshold; otherwise it uses P_{hi}. We set that threshold to 3σ, where σ is the standard deviation of the corresponding performance metric in the current path.

D. Random Routing Period (RRP)

RRP adds randomization in the routing period T_r (see Figure 5). Specifically, T_r is uniformly distributed in a range $[T_m, T_M]$, in which T_M is the maximum possible routing period, while T_m is the measurement period (1 second). When $T_M = T_m$, RRP is the same with DPS. Note that when $T_r > T_m$, the measurement period covers the last T_m time units of the routing period. A larger T_M makes it less likely that measurement periods of different IRC systems will overlap, but it also decreases the responsiveness of the IRC system.

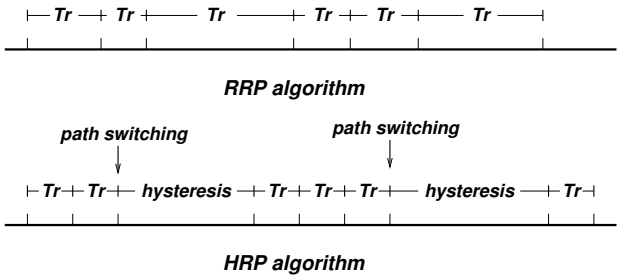


Fig. 5. RRP and HRP algorithms.

E. Hysteresis Routing Period (HRP)

HRP is a variation of RRP in which a hysteresis period is added after each path switching event. Figure 5 illustrates the difference between the two algorithms. Contrary to RRP, which has a random routing period, HRP uses a fixed T_r, which is equal to the measurement period T_m. However, after each path switching event HRP inserts a random hysteresis period. The purpose of the hysteresis period is to break any synchronization that may result after the last path switching. At the same time, HRP allows quick response to congestion, because it keeps T_r to its minimum. The length of the hysteresis period is uniformly distributed in a range $[0, T_H]$, where $T_r + T_H$ is the largest routing period.

V. Evaluation of IRC algorithms - stationary load

In the previous two sections, we illustrated that the basic path switching algorithm (DPS) can lead to synchronization and proposed four algorithms (FSP, ASP, RRP, HRP) that rely on randomization to avoid such synchronization. In this section, we evaluate these algorithms in terms of loss rate and switching frequency under stationary load. By stationary load, we mean that the average rate of the DRC flows, i.e., the background traffic, on each ingress link of the destination D remains time-invariant. We emphasize that stationarity does not mean constancy; on the contrary, both DRC and IRC traffic flows have highly variable rates, driven by the FGN model.

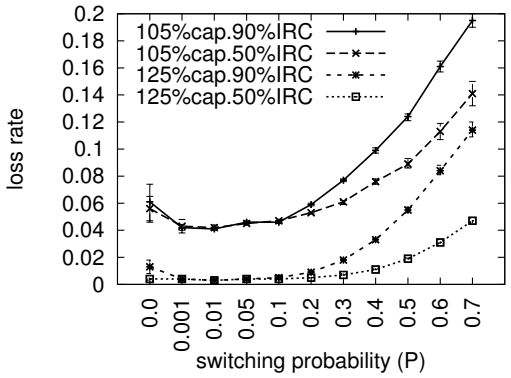
A. Simulation setup

The topology of the experiments is based on the network model of Section II in which n=100 sources send data to the destination network D through m ingress links. Among the n sources, n_I of them are IRC sources, while the remaining n_D are DRC flows (background traffic). The ratio n_I/n_D controls the fraction of IRC traffic relative to the statically routed background traffic. The traffic of each flow follows the FGN fluid model, as described in Section II, with a rate change every 100 msec. For the DPS, FSP, ASP, and HRP algorithms, the routing period T_r is set to one second, equal to the measurement period T_m. The start time of each flow is randomly chosen within the routing period. The first 50 seconds of simulation time are discarded to avoid any transient effects.

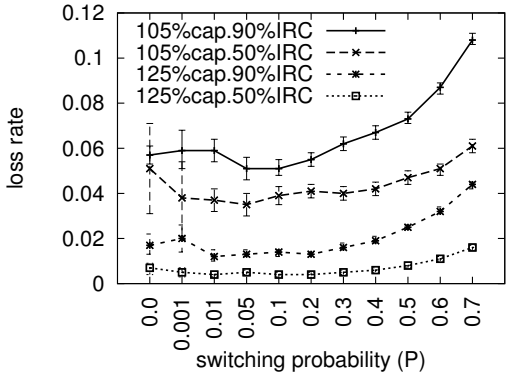
We use two metrics to evaluate the performance of IRC flows: loss rate and path switching frequency. Both metrics are defined over a time period. The loss rate of an IRC flow is defined as the fraction of the total traffic volume that is lost in that time period. The (path) switching frequency of an IRC flow is defined as the total number of path switching events during a time period divided by the length of that period. The loss rate evaluates the effectiveness of IRC in improving the performance of its traffic. The switching frequency, on the other hand, evaluates stability. The latter is important both for IRC flows (frequent path changes cause packet reordering and increased jitter) and for the underlying network performance (e.g., effectiveness of traffic engineering). The reader should distinguish between these two continuous-time metrics that are calculated from the simulator and the three metrics (queueing delay, loss rate, available bandwidth) that an IRC flow estimates during a measurement period using sampling.

We examine the effect of four important factors, with two values per factor, giving us 16 different simulation configurations. First, the ratio of average IRC load to the aggregate traffic load: 50% and 90%. Second, the distribution of average rates for the IRC flows: homogeneous (i.e., the same for all flows) and Zipf (shape parameter=1). Third, the aggregate capacity of ingress links relative to the total offered load: 105% and 125%. And fourth, the number of ingress links m: two and four. Note that we only examine what happens when the ratio of IRC traffic is significant (50% or more). If there is only a small fraction of IRC traffic, the synchronization effects that we examine are of minor significance for the aggregate traffic, but they can be important for the IRC flows. Also, we do not simulate greatly overprovisioned or underprovisioned links because such conditions lead to either zero loss rate and stable path selections (overprovisioning) or persistent losses at all paths (underprovisioning). In other words, IRC is not needed if there is plenty of capacity in the ingress links, and also it cannot avoid congestion if there is not enough aggregate capacity. It is the range in the middle that is interesting and important in practice, because that is the most cost-effective operating regime.

B. Performance of FSP



(a) Homogeneous flow rates



(b) Zipf flow rates

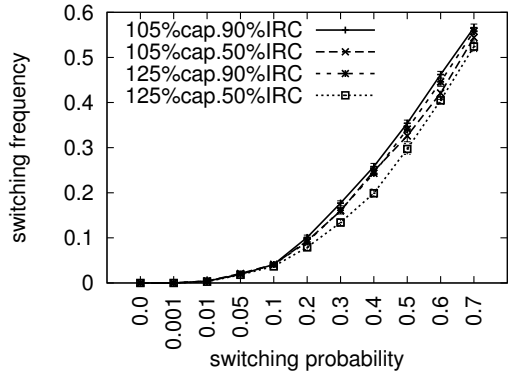
Fig. 6. Loss rate with FSP as a function of P (four ingress links).

We first study the performance of FSP under different values of the switching probability P. Figure 6 shows the loss rate with FSP as a function of P, for the homogeneous and Zipf flow rate distributions and for m=4 links. Each plot shows four curves, for all combinations of capacity (105% and 125%) and IRC traffic ratio (50% and 90%). The curves show the average loss rate across all IRC flows, as well as the 99% confidence intervals. We show more points at the low end of P (below 0.1) to illustrate the trend in that range.

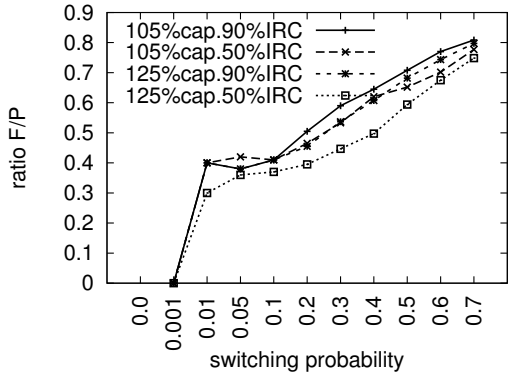
First note that as P increases above 20%, the loss rate increases significantly. This is due to synchronization of IRC flows. The synchronization becomes more prevalent with higher P because flows tend to switch paths more aggressively. With DPS (P=1.0, not shown here), the loss rate would be excessively large. Second, practically any small value of P, between 0.001 and 0.1 is sufficient to avoid synchronization and path switching (shown in Figure 7(a)), and to result in the minimum possible loss rate. Thus, FSP is robust in the selection of P. Another interesting observation is that even without any path switching (P=0) the loss rate is close to

minimum. This is because, with such stationary background traffic, IRC flows distribute themselves uniformly across the ingress links and then they rarely see that the performance is significantly better in another path. As will be shown in the next section, the situation is very different when we consider nonstationary traffic load and rapid congestion events.

Some more observations from Figure 6 follow. First, as expected, larger capacity leads to lower loss rate. Second, reducing the fraction of IRC traffic decreases the loss rate because the synchronization of IRC flows has lesser impact on the aggregate load at the ingress links. Third, the Zipf distribution results in a lower average loss rate across all flows, because it is mostly the few large flows that experience major losses when there is synchronization. This also explains why the loss rate confidence intervals are much wider with the Zipf distribution. Due to space constraints we do not present results for the case of two links; the results are similar.



(a) Switching frequency F



(b) Ratio F/P

Fig. 7. Switching frequency with FSP as a function of P (four ingress links and homogeneous flow rate distribution).

Figure 7(a) shows the switching frequency F of FSP in simulations with homogeneous flow rate distribution and four ingress links. The results are similar for other configurations. First, note that F is almost zero when P is less than 10%, validating that even a small switching probability can avoid synchronization. Second, F increases significantly as the path switching probability increases beyond 10%. There are two reasons for this. First, increasing P means that IRC flows will get the opportunity to switch paths more often, if they need to. So, if the traffic variations at the ingress links remained the same as we increase P, then we would observe that F increases linearly with P. On the contrary, Figure 7(b) shows that the ratio F/P is not constant, and that it increases with P. This means that as we increase P, not only IRC flows get the opportunity to switch paths more often, but they also need to switch paths more often. This is explained as follows: increasing P causes further synchronization among IRC flows, which causes larger fluctuations in the traffic of the ingress links. These traffic fluctuations trigger further IRC path switching and even greater synchronization. In other words, aggressive path switching creates a positive feedback loop between the synchronization of IRC flows and the traffic fluctuations in the underlying bottleneck links.

C. Parameterization of ASP, RRP and HRP

We conducted a similar study for ASP, RRP and HRP. In this section, we summarize the simulation results that resulted in the best parameters for these three algorithms, while the next section compares all path switching algorithms.

The ASP algorithm depends on two probabilities, P_{hi} and P_{lo}. We examined the following pairs of (P_{hi}, P_{lo}): (0.7, 0.3), (0.8, 0.2), (0.9, 0.1), (0.95, 0.05), and (1.0, 0.0). The simulation results show that the loss rate and switching frequency decrease as the difference (P_{hi}-P_{lo}) increases. When we reach the extreme pair (1.0, 0.0), the loss rate slightly increases under several configurations. We thus choose (0.95, 0.05) as the parameters for ASP under stationary load. This setting means that ASP switches to another path almost certainly when the current path is much worse, but it stays in the current path, almost always, otherwise.

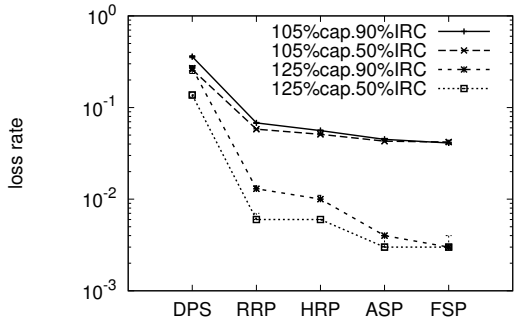
For the RRP algorithm, the parameter T_M controls the range of the maximum routing period. We examined the T_M range from T_m to $10T_m$. The simulations results show that the loss rate and switching frequency decrease rapidly as T_M increases away from T_m, but then they flatten out as T_M becomes larger than $4T_m$. The exact transition point depends on the simulation configuration. We set $T_M = 7T_m$, because the loss rate and switching frequency show diminishing returns for larger values, while the algorithm becomes less responsive to dynamic load changes as T_M increases.

For the HRP algorithm, the parameter T_H controls the range of the hysteresis period. We examined the T_H range from 0 to $40T_m$. Similar to RRP, the loss rate and switching frequency decrease rapidly at the beginning, when T_H is small. After about $5T_m$, however, the curves flatten out. We set T_H to $20T_m$, as a trade-off between performance and responsiveness.

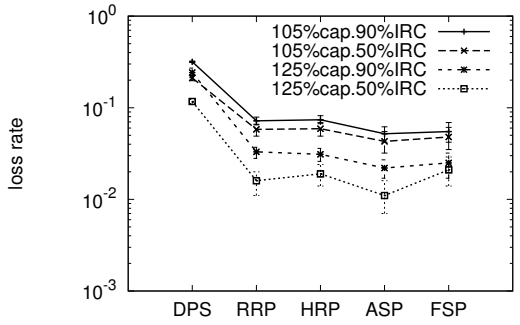
D. Comparison of DPS, FSP, ASP, RRP and HRP

In this section, we compare the performance of the five IRC algorithms we consider under stationary load conditions, using

the parameters given previously.



(a) Homogeneous flow rates



(b) Zipf flow rates

Fig. 8. Comparison of IRC algorithms– loss rate (four ingress links).

Figures 8 and 9 show the loss rate and the switching frequency of the five IRC algorithms in the m=4 link configuration. The results with other configurations, not shown here, have similar trends. First, note that in terms of loss rate the four probabilistic path switching algorithms (FSP, ASP, RRP, HRP) do much better than deterministic path switching (DPS), as they decrease the loss rate by an order of magnitude or more. Second, the randomized algorithms perform similarly, with ASP and FSP being slightly better than RRP and HRP.

The switching frequency results show similar trends. Without randomization, IRC path switching can cause major synchronization and oscillations. The four randomized algorithms have a clear difference in terms of switching frequency, however. FSP is the most stable, ASP comes next, while RRP and HRP are the least stable.

E. Summary

We found that introducing some randomness in the IRC path switching process can avoid synchronization and dramatically improve performance and stability compared to deterministic path switching. We also observed that, under stationary load, optimal performance and stability result from very conservative path switching. Finally, the exact form of randomization,

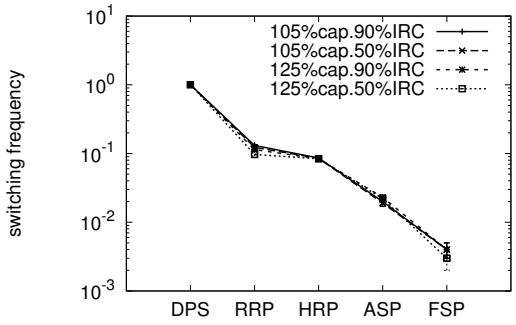


Fig. 9. Comparison of IRC algorithms– switching frequency (four ingress links, homogeneous flow rates).

or its parameters, do not seem to matter significantly, especially in terms of the resulting loss rate.

VI. Evaluation of IRC algorithms - nonstationary load

The simulation results of Section V examined the performance of IRC algorithms under stationary conditions, where the average DRC traffic load, i.e., the background traffic with which IRC flows share the links of D, remains constant. We also need to understand, however, the performance of IRC systems under dynamic network conditions in which the background traffic varies rapidly due to random events such as BGP rerouting, link/router failures, flash crowds, arrival/departure of major flows, etc. Instead of evaluating such dynamic conditions with simulations of individual load changes, we prefer instead to investigate the behavior of IRC in the presence of cyclostationary background traffic. The latter is a special form of nonstationary traffic in which the average rate varies periodically. The benefit of this approach is that it allows us to examine the performance of IRC as we vary the time scale in which congestion persists. In other words, the following evaluation resembles a frequency-domain analysis of IRC behavior, rather than a time-domain transient analysis.

A. Simulation setup

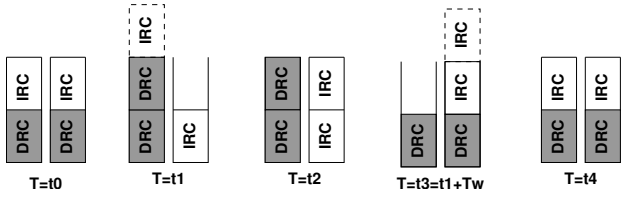


Fig. 10. The pattern of periodic load changes between two ingress links. At $t1$ a block of DRC traffic moves from the right link to the left. At $t3$, T_w time units later, an equally sized block moves back to the right link.

The simulations in this section refer to a two-link configuration with homogeneous DRC and IRC traffic flows. The total capacity is barely enough to carry the aggregate traffic load. The periodic pattern of the load changes at the two links is shown in Figure 10. Initially (t_0), the two links are equally loaded with IRC and DRC traffic. IRC flows do not switch

paths because the two links offer the same performance. At some time t_1, the DRC traffic of link-2 (at the right) moves to link-1. This causes major congestion at the latter, and so the IRC flows of link-1 gradually move to link-2 (t_2). The delay $t_2 - t_1$ depends on the aggressiveness (or responsiveness) of the IRC path switching algorithm. For FSP, a higher value of P will reduce the delay $t_2 - t_1$. If P=1 (DPS), $t_2 - t_1$ will be as short as one routing period T_r. Then, at some time $t_3 = t_1 + T_w$, half of the DRC traffic at link-1 moves to link-2. Again, this causes major congestion at the latter, and so half of the IRC traffic gradually moves to link-1 (t_4), getting us back to where we started at t_0. Note that the term "gradually" is only true here if P is much lower than one. Otherwise, the IRC traffic can experience oscillations before reaching the load-balanced configuration shown at t_4. This pattern can repeat periodically, with a period of $2T_w$, if the events at t_1 and t_3 occur every $2T_w$ time units.

Even though the previous traffic pattern is very artificial, it produces a periodic load variation that allows us to examine the performance of various IRC algorithms as a function of the period T_w. In particular, we are interested in the relation between T_w, which is the time scale in which congestion emerges, and T_m=1 second, which is the minimum time scale in which an IRC flow can detect congestion and react to it.

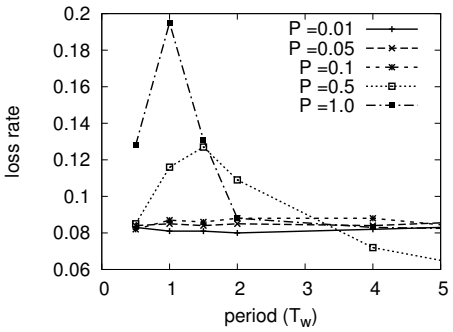
We examined the performance of the five IRC algorithms (DPS, FSP, ASP, RRP and HRP) as T_w varies from 0.5 seconds to 400 seconds. This range is sufficient to show the three important modes in the behavior of IRC. Each simulation has 100 flows, half of which are IRC flows. The total capacity of the two identical links is set to 105% of the aggregate traffic load. The rest of the parameters are as in Section V.

B. Performance of FSP under nonstationary load

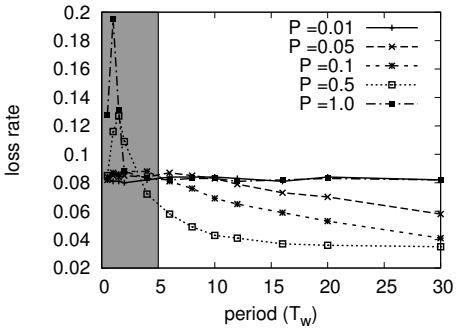
Due to space constraints, we present more detailed results only for FSP with P set to 0.01, 0.05, 0.1, 0.5, and 1.0 (DPS). The other three randomized algorithms perform similarly. We identify three distinct ranges of T_w in terms of the resulting loss rate and trends.

First, Figure 11(a) shows what happens when T_w is less than 5 seconds. Recall that the routing period as well as the measurement period of FSP are set to 1 second, meaning that in this range of T_w congestion emerges almost with the same frequency with which FSP can detect whether it should switch paths. Note that IRC does very poorly when P is high, 0.5 or higher, and T_w is less than 2-3 seconds. The reason is that in that range FSP tries to "catch its tail", switching between paths with almost the same frequency with which the background traffic moves between these paths. On the contrary, IRC does best when it rarely switches paths, i.e., when P is very low or zero. Hence, IRC techniques will not improve performance (they will actually hurt performance) if they detect and react to congestion in the same time scales in which congestion emerges.

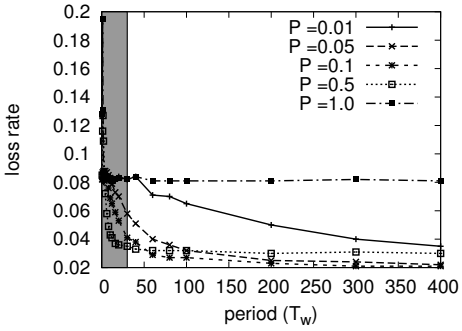
Second, Figure 11(b) (the non-shaded part) shows what happens when T_w is larger than 5 seconds and lower than 30 seconds. In this case, congestion emerges much less frequently



(a) Tw = (0, 5)



(b) Tw = (0, 30)



(c) Tw = (0,400)

Fig. 11. Loss rate with FSP in three ranges of T_w, for different values of P.

compared to the IRC time scale (one second), but nevertheless it does emerge periodically and it is significant. The results are quite different now. The loss rate can be significantly reduced with IRC compared to static routing or P=0. Also, an aggressive form of FSP with P=0.5 performs better than the conservative path switching probabilities 0.1 and 0.05. Deterministic path switching, on the other hand, does not perform better, because it still causes synchronization of IRC flows. Hence, when IRC techniques are fast enough to detect congestion when the latter is still at its onset, IRC can

significantly improve performance. In that regime, IRC can also be more aggressive in terms of path switching compared to stationary load conditions.

Finally, Figure 11(c) (the non-shaded part) shows what happens when T_w is larger than 30 seconds and lower than 400 seconds. In this case, the background traffic goes through major fluctuations only rarely. Thus, this case is not very different than the stationary load evaluation of Section V. Indeed, we see that as T_w increases, conservative path switching does better. Eventually, as T_w tends to infinity, the best choice for P becomes 0.01. Hence, when congestion is a rare event, IRC is still beneficial but it should be quite conservative in terms of path switching.

C. Comparison of FSP, ASP, RRP and HRP under nonstationary load

Here, we compare the performance of the four randomized IRC algorithms (FSP, ASP, RRP and HRP) under nonstationary traffic load. In this set of simulations, the parameters of the ASP, RRP and HRP algorithms are as in Section V. For FSP, we set P=0.5.

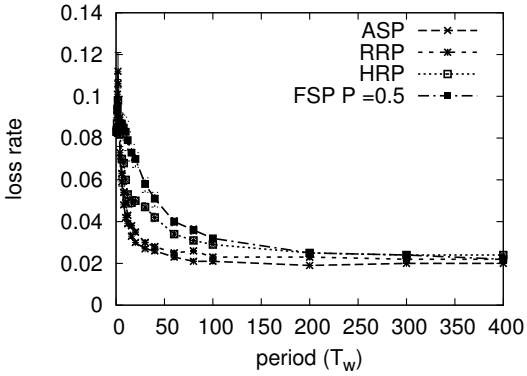


Fig. 12. Loss rate with FSP, ASP, RRP and HRP as we vary T_w.

The results are shown in Figure 12. As was the case with FSP, all IRC algorithms perform poorly when T_w is very short, below 5 seconds. Also, the algorithms do not show major differences when T_w is very large, say above 200 seconds, which is consistent with our earlier comparisons in Section V. The more interesting range is when T_w falls between 10 to 100 seconds. In that case, the ASP and RRP algorithms perform best, HRP follows, and FSP comes last. We should note however that these comparisons are somewhat dependent on the parameterization of the four algorithms.

D. Summary

The simulations of this section revealed some interesting observations. First, the effectiveness of IRC techniques, in general, depends on the time scale in which such techniques can detect and react to congestion relative to the time scales in which congestion persists. If congestion appears rapidly and it only lasts for a few seconds, it would be very hard for IRC systems to avoid it given that they also need a few seconds to detect congestion through measurements. In that case, it may be better to stay at the same path and deal with congestion through other means. On the other hand, if congestion lasts for many seconds, we expect that IRC systems can be fast enough to detect it and switch to another path.

VII. Heterogeneous IRC sources

In the previous sections, we showed that the four randomized path switching algorithms perform well when all IRC flows use the same algorithm. In this section, we investigate how these algorithms perform in a heterogeneous environment. We also investigate the coexistence of deterministic path switching with randomized path switching. This is a critical question for the gradual deployment of the latter.

A. Heterogeneous IRC algorithms

In practice, we expect that different IRC systems will be using different path switching techniques, with diverse forms and parameters of randomization. To examine such a heterogeneous environment, we simulated several configurations where different IRC flows use FSP, ASP, RRP and HRP, each algorithm adopted by the same number of flows.

The results are only summarized here due to space constraints. First, we observed that in such a heterogeneous environment the difference, in terms of loss rate, between the four randomized algorithms is further decreased (i.e., heterogeneity causes assimilation). The reason is that, since all IRC flows share the same bottleneck links, they will all observe, on the average, the same loss rate. On the other hand, the IRC flows still do much better than the deterministic path switching flows as the latter experience persistent oscillations. Hence, we expect that even if different IRC vendors adopt different path switching techniques, the resulting traffic will be stable as long as there is some degree of randomization in their switching techniques.

B. Coexistence of deterministic and randomized path switching

In practice, some networks may continue to use deterministic path switching techniques, while others gradually deploy randomized path switching. The critical question in such an environment is whether IRC users have the incentive to switch to randomized IRC or whether they will do better for themselves being more aggressive.

We investigate this question simulating a fraction of randomized IRC traffic with the rest of the traffic doing deterministic path switching (DPS). Figure 13 shows the resulting loss rate when the randomized IRC algorithm is FSP with P=0.01. The results with other algorithms show similar trends. We start from 5% FSP and 95% DPS traffic and gradually increase the fraction of FSP traffic to 95%.

First, note the large gap between the loss rate of DPS and FSP flows. When the ratio of FSP to DPS traffic is 5:95, the loss rate of the former is only about 40% of the loss rate of the latter. This performance difference indicates that when FSP and DPS flows coexist at a network, the FSP algorithm gives a major advantage to its users over the DPS algorithm.

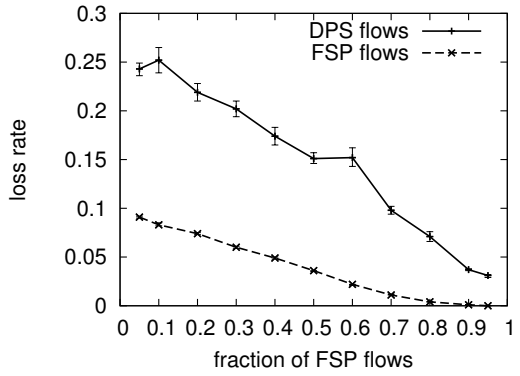


Fig. 13. Loss rate when FSP and DPS traffic coexist (four ingress links, homogeneous flow rates).

We also observe that with the increase of FSP traffic, the loss rates for both FSP and DPS flows decrease. This improvement of the overall performance results from the decreased fraction of DPS traffic, which leads to reduced synchronization.

These results are encouraging for two reasons. First, the substantial difference between FSP and DPS shows that multihomed networks will have a strong incentive to use IRC systems with randomized rather than deterministic path switching. Second, the decreasing loss rate as the fraction of FSP traffic increases suggests that the gradual deployment of randomized IRC systems will benefit all traffic sharing the same bottlenecks.

VIII. Conclusions

Oscillations due to load-driven routing are certainly not a new or hypothetical risk. For a long time, Internet routing protocols have avoided load-driven routing exactly because of this risk. With the deployment of technologies such as IRC at multihomed networks and routing overlays, semi-static routing in the network core no longer provides protection against oscillation. Individual stub networks are already using IRC systems, without a thorough evaluation of what could happen if the amount of IRC traffic, or the number of independent IRC systems, becomes significant.

In this paper, we first showed that IRC systems can cause sustained traffic oscillations under reasonable conditions, that can easily occur in practice. We hope that this negative result will motivate further research in the appropriate design of IRC systems, as well as in measurement studies at multihomed networks that use IRC. We also showed that some simple randomization techniques in the path switching algorithm, as well as the use of available bandwidth measurements, can be effective in avoiding IRC-induced traffic oscillations. Nevertheless, the fact that even the randomized IRC algorithms can experience oscillations when the traffic changes very quickly relative to the IRC time scales means that the problem is not entirely solved. We anticipate that this area will attract significant research interest in the near future.

References

[1] T. Bates and Y. Rekhter, "Internet RFC 2260: Scalable Support for Multi-homed Multi-provider Connectivity," January 1998.
[2] InterNAP, "Premise-Based Route Optimization."
[3] Route Science, "Adaptive Networking Software."
[4] f5 networks, "BIG-IP Link Controller."
[5] Radware, "Peer Director."
[6] Rainfinity, "RainConnect."
[7] Stonesoft, "StoneGate Multi-Link Technology."
[8] FatPipe, "WARP."
[9] Cisco Systems, "Optimized Edge Routing (OER)."
[10] F. Guo, J. Chen, W. Li, and T. Chiueh, "Experiences in Building A Multihoming Load Balancing System," in *Proceedings of IEEE INFOCOM*, 2004.
[11] A. Akella, S. Seshan, and A. Shaikh, "Multihoming Performance Benefits: An Experimental Evaluation of Practical Enterprise Strategies," in *Proceedings of USENIX Annual Technical Symposium*, 2004.
[12] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, "A Measurement-Based Analysis of Multihoming," in *Proceedings of ACM SIGCOMM*, 2003.
[13] A. Akella, J. Pang, A. Shaikh, B. Maggs, and S. Seshan, "A Comparison of Overlay Routing and Multihoming Route Control," in *Proceedings of ACM SIGCOMM*, 2004.
[14] D. Goldenberg, L. Qiu, H. Xie, Y. Yang, and Y. Zhang, "Optimizing Cost and Performance for Multihoming," in *Proceedings of ACM SIGCOMM*, 2004.
[15] H. Wang, H. Xie, L. Qiu, A. Silberschatz, and Y. Yang, "Optimal ISP Subscription for Internet Multihoming: Algorithm Design and Implication Analysis," in *Proceedings of IEEE INFOCOM*, 2005.
[16] S. Tao, K. Xu, Y. Xu, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z. Zhang, "Exploring the Performance Benefits of End-to-End Path Switching," in *Proceedings of IEEE ICNP*, 2004.
[17] R. Keralapura, C.-N. Chuah, N. Taft, and G. Iannaccone, "Can Coexisting Overlays Inadvertently Step on Each Other," in *Proceedings of IEEE ICNP*, 2005.
[18] V. Paxson, "Fast Approximation of Self-Similar Network Traffic," *ACM SIGCOMM Computer Communications Review*, vol. 27, no. 5, pp. 5–18, oct 1997.
[19] M. Jain and C. Dovrolis, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 537–549, Aug. 2003.