Designing Router Scheduling Policies: A Privacy Perspective

Sachin Kadloor[†*], Xun Gong[†*], Negar Kiyavash[‡*], Parv Venkitasubramaniam[§]
[†] ECE Department and Coordinated Science Lab.
[‡] IESE Department and Coordinated Science Lab.
[*]University of Illinois at Urbana-Champaign, Urbana, IL, USA.
[§]ECE department, Lehigh University, Bethlehem, PA, USA.
{kadloor1,xungong1,kiyavash}@illinois.edu, parv.v@lehigh.edu

ABSTRACT

We examine a queuing side channel which results from a shared resource between two users in the context of packet networks. We consider the scenario where one of them is a legitimate user and the other is an attacker who is trying to learn about the former's activities. We show that the waiting time of an adversary sending a small but frequent probe stream to the shared resource (e.g., a router) is highly correlated with traffic pattern of the user. Through precise modeling of the constituent flows and the scheduling policy of the shared resource, we describe a dynamic program to compute the optimal privacy preserving policy that minimizes the correlation between user's traffic and attacker's waiting times. While the explosion of state-space for the problem prohibits us from characterizing the optimal policy, we derive a sub-optimal policy using a myopic approximation to the problem. Through simulation results, we show that indeed the sub-optimal policy does very well in high traffic regime. Furthermore, we compare the privacy/delay trade-offs among various scheduling policies, some already widely deployed in scheduling and others suggested by us based on the intuition from the myopic approximation.

Categories and Subject Descriptors

C.2.0 [**Computer-Communications Networks**]: General—*Security and protection(e.g., firewalls)*; C.2.3 [**Computer-Communications Networks**]: Network Operations—*Network monitoring*; I.5.4 [**Pattern Recognition**]: Applications; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms

Security, Theory, Algorithms

1. INTRODUCTION

It has long been known that a shared resource in a network leads to a covert channel that can be used for communication between different processes. However, shared resources not only lead to covert but also side channels– information leaks about the activities of one process to another without the former process's cooperation. In this work, we propose to explore the side channel resulting from the queueing of packets from multiple users at a router.

We begin by a motivating example. A user, Alice, is using her computer at home to connect to the Internet. She connects by using a home DSL router, which connects to a router at her Internet service provider (ISP), connecting to the Internet. The ISP sees all the traffic that Alice sends; however, Alice is not worried because she knows that she is protected by Anti-wiretapping legislation, and she encrypts all her most sensitive information.

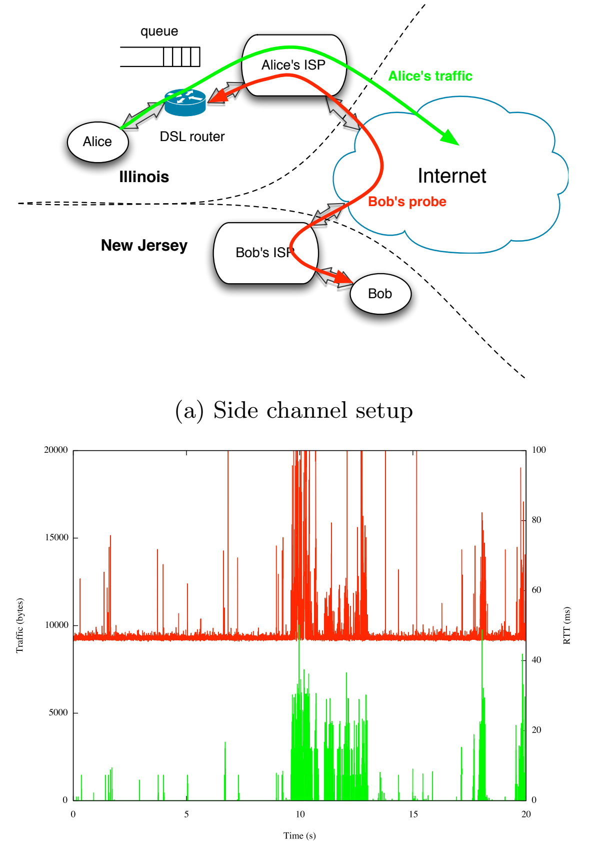
Along comes Bob, who is located at another ISP entirely, perhaps even in another country. Bob sends a probe stream to Alice's router. The probes are frequent, but small in size. Most importantly, the probes (and responses) make use of a shared queue at Alice's DSL router. As a result, the waiting time of the probes is correlated with Alice's traffic patterns. Bob, by carrying out the attack as explained below can make sure that he shares a resource which is in use by Alice: the packet queue inside her DSL router. This resource allows him to create a side channel that will leak information about Alice's traffic. This scenario is illustrated in Figure 1(a). Bob sends a low-bandwidth, but high frequency probe to the router and measures the round-trip time (RTT) of his probe packets. The DSL has an incoming and an outgoing port, for the traffics addressed to, and originating from Alice's computer. Bob's probe responses and Alice's incoming traffic share the same queue, hence the delay that Bob observes would vary based on the pattern of traffic addressed to Alice. Although these pings travel through various intermediate routers, their roundtrip time is primarily affected by Alice's traffic. This is because of two reasons. One, the intermediate routers have significantly higher bandwidth compared to the volume of the traffic flowing through them making Alice's router the bottleneck. Also, the intermediate routers carry multiple traffic flows. Therefore, the delays incurred at these routers does not change with time.

To evaluate the potential for this type of attack, we observed traffic of a home DSL user in Illinois, while simultaneously sending a ping probe from a computer in New Jersey every 10ms. Figure 1(b) shows the results: the DSL traffic is shown in green in the bottom half of the graph, and the round-trip times of the ping probe are shown in red in the upper half, showing a clear correlation between the two. The probe traffic used less than 50 Kbps of bandwidth and is unlikely to be noticed. Note that our attack can only reveal the timing and the volume of the traffic, and not the actual contents of the packets. However, recent research has shown that significant inferences can be drawn by observing just the traffic pattern of a user.

CCS'10, October 4–8, 2010, Chicago, Illinois, USA.
ACM 978-1-4503-0244-9/10/10.

Due to the high correlation between probe waiting times and Alice's traffic pattern, Bob can reliably fingerprint the websites Alice visited. The primary reason for the high correlation is the first-come-first-serve (FCFS) queuing policy of the shared DSL router. While FCFS policy is an attractive choice in terms of delays and utilization, the high correlation between Bob's waiting times and Alice's traffic pattern is highly unattractive in terms of preserving privacy. In this paper, we consider the problem of designing router scheduling policies that mitigate this traffic analysis attack. Now consider another extreme policy, namely, time division multiple access (TDMA) where a user is assigned a fix service time regardless of whether he has any packets that need to be serviced. As expected, in this case, Bob's waiting times are independent of Alice's traffic pattern. However, TDMA is a highly inefficient policy in terms of throughput and delay. This trade-off between the information leakage and efficiency (in terms of delay or throughput) is inherent to the router policy design, and the goal of this paper is to design scheduling policies that mitigate the information leakage but at the same time have good performance guarantees.

The main contribution of this work is that we analyze the router-based side channel by developing a precise queueing model for the router and the constituent flows. We propose a *correlation* metric based on this model to measure the extent of information leakage between a target user flow and an attacker's probe, for any scheduling strategy. We describe a dynamic program to compute the optimal scheduling strategy that minimizes the correlation metric. However, the size of state-space for the problem prohibits us from deriving the optimal policy. Instead, we propose a sub-optimal policy using a myopic approximation to the problem and show that indeed the sub-optimal policy does very well in high traffic regime. Through extensive experimental evaluation, we compare the privacy/delay trade-offs among various policies, some already widely deployed in scheduling and others suggested by us based on the intuition from the myopic approximation.



(a) Side channel setup

(b) Real traffic on a DSL line vs. observed probe RTTs.

Figure 1: Queueing Side Channel

2. OPTIMAL SCHEDULING POLICY: MARKOV DECISION FRAMEWORK

Our system model consists of a router serving two streams of data. Each time the router is free and there are packets waiting to be served, the router needs to make a decision on which packet to serve next. Serving packets from one stream will induce delays to the packets from the other stream, which indirectly leaks information about the size of flow in one stream to the other. As a measure of the leakage of information, we will consider the *correlation between the arrival pattern of the packets in one stream and waiting times of the packets in another stream.*

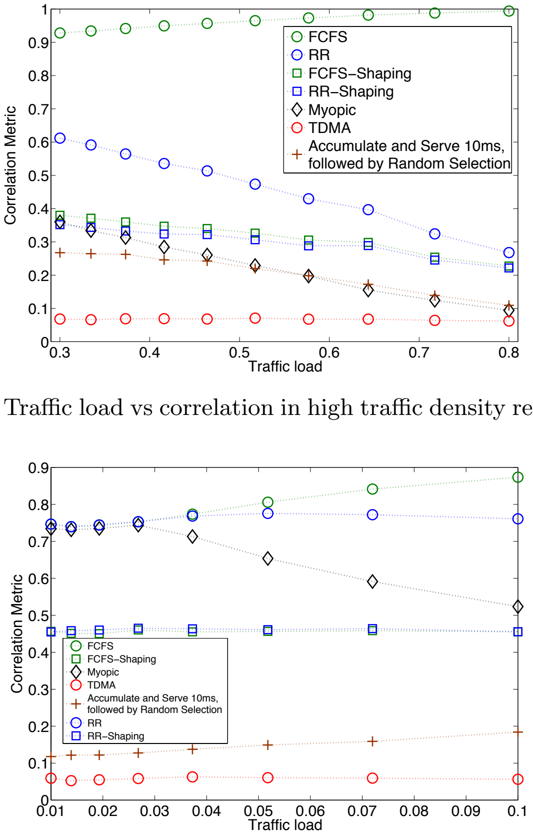
Correlation Metric: From the router's perspective, either of the streams could belong to legitimate user and hence, any metric cannot assume a specific stream to be the attacker. The *correlation metric* for a router policy ψ is therefore defined as the maximum correlation coefficient across the streams:

$$\max\left(\rho(\tilde{\underline{\mathrm{x}}}^{(1)}_{N_1}, \tilde{\underline{\mathrm{T}}}^{(1)}_{N_1}), \rho(\tilde{\underline{\mathrm{x}}}^{(2)}_{N_2}, \tilde{\underline{\mathrm{T}}}^{(2)}_{N_2})\right), \tag{1}$$

where $\tilde{\underline{\mathrm{x}}}^{(k)}_{N_i}$ is a vector of arrival volumes in stream k between consecutive arrivals of packets in the other stream, and $\tilde{\underline{\mathrm{T}}}^{(k)}_{N_i}$ is the inter-departure times of the packets in the other stream. The correlation metric captures the influence of the traffic pattern in one stream to the delays experienced by packets in the other. If the router employs a FCFS policy, then the only reason an incoming packet waits at the head of its queue is because the router is busy serving all the packets that have arrived in the other stream between its arrival time that of the previous packet in its stream. The value of this correlation is hence, close to one if FCFS policy is used by the router.

The optimal scheduling policy that minimizes the correlation metric can be formulated using a Markov Decision Process (MDP) framework. MDP provides a mathematical framework to model decision making in situations where the outcomes probabilistically depend on the actions. For the sake of brevity, we do not list the state update equations here. We can show that in order to solve for an optimal policy, the router will have to store a lot of information because of the high demensionality of the state space. The framework is thus not very useful in solving for an optimal policy, one can however use it to provide an approximate solution by considering a myopic optimization.

Myopic policy. We define a myopic policy in the above framework which performs a greedy optimization at every step of the process. More specifically, at time t, the router serves one packet from that stream for which the resulting correlation metric is lower.



(a) Traffic load vs correlation in high traffic density regime

(b) Traffic load vs correlation in the low traffic density regime

Figure 2: Correlation-traffic load for the different policies

3. EXPERIMENTAL RESULTS

The system model for the experiment is that there is a single router serving packets from two streams. For the inputs to stream one, we used synthetic traffic which was generated according to Pareto distribution. The traffic in the second stream is assumed to be a train of pings which are sent at a regular interval of 10ms. We compare the performance of different scheduling policies using the correlation metric. The schemes we compare are FCFS, Round Robin (RR), FCFS-shaping[1], RR-shaping, Myopic policy, Accumulate and Serve[2], and TDMA. Results are presented in Figure 2.

3.1 Inference from simulation results

From our experiments, some of the broad inferences: The FCFS and the TDMA policies represent two extreme points in terms of correlation, with FCFS being the largest and TDMA the least. This follows our previously described intuition that FCFS reveals maximum information about the traffic pattern whereas with TDMA, the observables and the arrival information are independent of each other.

Table 1: Results of the Classification Experiment

Policy	Accuracy	Correlation	Average delay (s)
FCFS	71.11%	0.9874	0.4450
AS_10	31.11%	0.1514	0.4576
AS_30	25.78%	0.0960	0.4843
AS_40	22.22%	0.0919	0.4988
TDMA	4.00%	0.0118	2.6108

In the high traffic regime, the myopic policy based on the dynamic program framework provides the least correlation, while the accumulate and serve has the least correlation for medium and low traffic loads. In fact at high traffic loads, the correlation performance of both these policies are close to that of TDMA with significantly lower delay[3]. Since the myopic policy is directed at optimizing the correlation metric, and the accumulate and serve is an approximation of the optimal infinite delay policy, the good correlation performance of these policies are as expected.

Classification Experiment. Here we describe the results of the experiment wherein we studied the efficacy of the accumulate and serve policy in mitigating the remote traffic analysis attack. We consider the situation where the attacker, Bob, is interested in identifying which website Alice is browsing. We assume that Alice is browsing one of the top twenty four news websites listed in `http://www.alexa.com/topsites/category/Top/News`. We carried out the attack described above by simulating the functioning of the router, and by using real traffic traces captured using tcpdump. The attacker is assumed to know which policy Alice's router is using. The classification results are presented in Table 1. A detailed account of the attack can be found in [1]. When FCFS policy was used, the attacker could correctly identify Alice's website over 70% of the time. In the other extreme, the TDMA policy resulted in a classification percentage of less than 10%. We also tested the AS policy with different accumulate times, 10ms, 30ms and 40 ms, the classification percentage for the latter being less than 25%, thus demonstrating the ability of the AS policy in preventing this attack to a great extent, relative to the commonly used FCFS policy. Table 1 also shows the the average delays experienced by Alice's packets for the policies tested. The delays for AS policies are all within 10% of delays for FCFS, hopefully a tolerable trade-off in return for improvement in Alice's privacy. Furthermore, it can be seen that the policies with lower correlation metric also have a lower classification rate. This serves as a validation to show that the correlation metric does indeed is nicely tied to the operational performance metrics such as classification error.

4. REFERENCES

[1] X. Gong, N. Kiyavash and N. Borisov, *Fingerprinting Websites Using Remote Traffic Analysis*, in preparation for conference submission, Available online at `http://www.ifp.illinois.edu/~kadloor1/attackdescription.pdf`

[1] In the shaping schemes, artificial delays are added to the input streams.

[2] Packets from both the streams are buffered for a certain amount of time after which all these buffered packets are served in a random order.

[3] The results where we compare the delay performance of these schemes is not presented here for the sake of brevity.