

Inferring a Network Congestion Map with Zero Traffic Overhead

Florin Dinu T. S. Eugene Ng
Computer Science Department, Rice University

Abstract—This paper proposes a purely passive method for inferring a congestion map of a network. The congestion map is computed using the congestion markings carried in existing traffic, and is continuously updated as traffic is received. Consequently, congestion changes can be tracked in a real-time fashion with zero traffic overhead. Unlike active congestion reporting methods, our novel passive method is more robust during periods of congestion because there are no congestion report messages that could be lost and existing congestion is never aggravated. Our solution has several applications ranging from informing IP fast re-route algorithms and traffic engineering schemes to assisting in inter-domain path selection.

I. INTRODUCTION

Information about congestion is of great importance to a network. Congestion indicates a severe degradation of the service level provided by the network and even a possible danger to the stability of the basic routing functionality [21]. Several distributed router level algorithms can benefit from congestion information. For example, IP fast re-route algorithms are currently ignorant of congestion [18][17][22]. It is important that these algorithm be extended to properly take network congestion into account because their routing decisions could cause congestion and also the efficiency of their decisions is impacted by existing congestion. Automated verification mechanisms that use multiple vantage points to verify the correct functioning of routers [19][15][24] can use congestion information to reason whether packet loss was caused by congestion or by errors in protocol implementations. Distributed traffic engineering algorithms [13] can use congestion information to balance the load in the network. At the inter-domain level, congestion information can be leveraged in the selection of suitable inter-domain routing paths.

Our novel proposition is that a congestion map of the network can be locally obtained at any router by processing the congestion information carried in the existing traffic that passes through that router. No extra traffic is added into the network. Our solution combines path level congestion information with routing information. Today, both types of information can be obtained from standardized protocols. First, the explicit congestion notification (ECN) protocol alongside an active queue management (AQM) protocol enables existing traffic to carry aggregate, path level congestion severity information. We present a complete method for inferring path level congestion information by leveraging the congestion markings from ACK and data packets. Subsequently, information from a link-state intra-domain routing protocol can be used to break down the

path level congestion information into detailed link level congestion location and severity information. At the inter-domain level our solution allows a local border router to extend its congestion map with aggregate inter-domain path congestion information. In our solution, at a border router, each reachable remote network is abstracted as a virtual link connected to the border router. The congestion inference method for a virtual link is precisely the same method used in the local autonomous system (AS). A congestion map is the union of all the link level and path level congestion information inferred at a router.

Our inference method is useful in inter-domain scenarios where, due to administrative boundaries, routers in different ASes cannot be actively queried for their congestion level. In addition, even in the local AS, our method has several important advantages compared to methods that rely on active congestion reporting. First, our method does not aggravate congestion episodes. In contrast, active congestion report messages increase the severity of congestion episodes. Even more, with our solution, routers also have the flexibility to choose different sampling granularities locally without affecting the network traffic. In contrast, to obtain fine-grained real-time congestion information, an active reporting method must resort to increasing the reporting frequency, further increasing the traffic overhead and exacerbating existing congestion. This weakness of active congestion reporting is especially detrimental because precisely during congestion periods fine-grained real-time congestion information is most useful. Second, our method is more robust during congestion periods since there are no report messages that could be lost because of congestion. Third, our method computes a fine-grained congestion map that is continuously updated in real-time as traffic is received, yet incurs zero traffic overhead.

We test the accuracy of our method against several important factors using numerical analysis and ns-2 simulations. The factors include sudden variations in the congestion level, the type of AQM used and multiple consecutive congestion points. We also analyze the effect of factors that can distort the sequences of packet markings. Despite the influence of all these factors our solution can infer congestion at a remote link with good accuracy and at fine time scales.

The rest of the paper is organized as follows. §II describes prerequisites and overviews the solution. §III discusses the inference of aggregate path level congestion. §IV details the computation of link level estimates. §VI and §V evaluate the accuracy of our method. §VII discusses our solution further, §VIII presents related work and §IX concludes.

II. OVERVIEW

Prerequisites: Our solution infers congestion inside a local AS and on the inter-domain paths that take traffic to and away from the AS. For local inference our solution uses the traffic that originates in and is destined to the same AS. For inter-domain inference the inter-domain traffic that has either the source or destination in the local AS is used. To have their congestion level inferred by our solution, local routers need an AQM algorithm [12][5][9] and the ECN protocol [20] for explicit congestion marking. The more AQM/ECN enabled routers are present in the network the more complete the inference is. The AS also needs to use a link state intra-domain routing protocol (e.g. OSPF). At the inter-domain level, as long as the routers at strategic points most susceptible to congestion (e.g. traffic exchange points) are AQM/ECN enabled, our solution will provide benefit.

Background: An AQM enabled router may mark data packets instead of dropping them. Specifically, the AQM algorithm [12][5][9] at a router computes a congestion measure for the router's outgoing links. This measure can be as simple as a function of the router queue size (e.g. RED [12]) or a more complex expression based on incoming traffic rate and available bandwidth (e.g. REM [5]). The router can then mark each outgoing data packet probabilistically, as a function of the congestion measure of the link they are sent on.

ECN [20] is the protocol that enables congestion marking. It makes use of four packet header bits: ECN-Echo (ECE) and Congestion Window Reduced (CWR) in the TCP header and ECT (ECN-Capable Transport) and CE (Congestion Experienced) in the IP header. ECN capable data packets have the ECT bit set. Congested routers can mark such packets by setting the CE bit. When receiving a data packet with the CE bit set, a TCP destination sets the ECE bit in the subsequent ACK and continues to do so for all following ACKs until it receives a data packet with the CWR bit set. The CWR bit is set by a TCP source to signal a decrease in the congestion window. This can happen as a result of receiving an ACK with the ECE bit set or for other reasons. The ECN markings on the two halves of a TCP connection are independent.

Definitions and notations: We use the term data or forward path to refer to the path taken by data packets and ACK path to refer to the path taken by ACK packets. We call a packet with the ECE bit set a *marked ACK* packet and a packet with the CE bit set a *marked data* packet. Unmarked packets do not have those bits set. As explained, a TCP receiver can mark multiple ACKs as a result of receiving one marked data packet. Consequently, the sequence and percentage of the markings in the data packets can be modified when the TCP receiver echoes the markings to the ACKs. We refer to this process as the *alteration* caused by the TCP receiver. For brevity we use *MP* to denote marking probability and *LMP* and *PMP* to denote link level and path level marking probabilities. A group of unmarked ACKs is a maximal sequence of consecutive unmarked ACKs. A group of unmarked ACKs of size zero is considered to appear whenever the echoing of markings in

ACKs finishes but has to resume immediately. This situation arises when a data packet has both the CE and CWR bits set.

Overview of the solution: Our solution allows a router to locally infer the congestion severity at other routers and network paths. In this paper we use the MP as the representation of congestion severity because the MP is common to several AQM algorithms.

Our solution applies to any network topology on a per-path basis. Routers first analyze the data and ACK markings received on separate network paths. The analysis of data packet markings leverages the percentage of marked data packets and allows congestion inference to be performed on the path taken by the data packets from the source until the analyzing router. The analysis of ACK markings relies on the size of the groups of unmarked ACKs and allows congestion inference on the entire forward path of flows. This path level analysis yields aggregate PMPs. After computing PMPs, a router can leverage link state routing information which is reliably disseminated and is sufficient to allow each router to compute the hop level path that a packet takes from a source to a destination. Using both PMPs and the hop level path description, aggregate PMPs for increasingly shorter paths can be derived. Thus, LMPs can also be computed with this approach. The set of all PMPs and LMPs inferred by a router forms a congestion map.

Inter-domain congestion inference: Aggregate congestion information for inter-domain paths can be computed with the same mechanism that applies to the problem of congestion inference for the local AS. Since one AS typically has no topology information about other ASes, we abstract the path from a local border to an external destination as one virtual link connected to the border router. Local border routers obtain aggregate congestion information for these virtual links. In this inter-domain scenario, data packet congestion markings convey congestion information about ingress routes. ACK path congestion markings give the local border routers congestion information about egress routes. The obtained congestion information can be leveraged when selecting good inter-domain paths and when advertising paths to neighbors. One advantage of our solution is that as re-routing occurs outside the local AS on an inter-domain path, our solution always tracks the congestion on the currently used path. Given the large number of destination reachable in the inter-domain, an operator may wish to restrict the number of destinations for which the inference is performed or can perform inference on demand. For the rest of the paper we consider that an AS network topology includes these virtual links. As a result of applying our solution to this topology both the detailed congestion map of the local AS as well as the aggregate inter-domain path congestion information will be obtained.

III. INFERRING PATH LEVEL CONGESTION INFORMATION

In this section, we describe the initial building block for obtaining a congestion map of the network: the inference of PMPs from packet markings. PMPs can be obtained from either data or ACK markings. However, the analysis of these two types of markings presents different challenges. In this

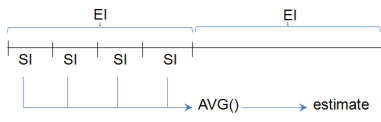


Fig. 1. Sampling and estimation intervals

paper we solve the more difficult problem of ACK based inference. ACK based congestion inference is vital because it conveys information about downstream router paths, exactly the paths that carry forwarded traffic.

A. Estimation and Sampling Intervals

Our inference solution uses two parameters: a sampling interval (SI) and an estimation interval (EI). For each SI, a single *sample* is obtained from all the markings received during the SI. Choosing a very low SI value may result in too few packet markings to compute a meaningful sample. On the other hand, a large SI value biases the sample towards the periods where bursts of packets are received. One cause for these bursts is the burstiness inherent in the use of TCP. We found that an SI on the order of the RTT of the network performs well because it works at the scale of TCP's burstiness while at the same time providing enough packets for the inference.

The EI represents the granularity at which inference is performed. The inferred LMPs and PMPs represent congestion at the scale of an EI. Figure 1 depicts the relationship between the EI and the SI. An EI is a multiple of an SI. For each EI, an *estimate* of the congestion severity is computed by averaging all the samples obtained during the EI. The trade-off present in choosing an EI value will become clear after we describe our solution.

B. Data Packet Based Inference

The data packet markings are exactly the markings set by routers. They are never altered until the data packets reach the destination. To compute a sample for a path using data packet markings a router counts the total number of data packets and the total number of marked data packets it receives across all flows traversing the path. Using these two counters, the ratio of marked data packets is obtained. This ratio serves as the sample for an SI.

C. ACK Packet Based Inference

The ACK markings are the result of the echoing performed by the TCP receiver. The challenge is to infer accurate congestion estimates despite the alteration caused by the echoing. At first glance, computing a sample using ACK markings could also leverage the percentage of marked ACKs. On closer inspection, such a solution is unfortunately inadequate because the alteration of the markings can cause a significant overestimation of the MP. In [10] we use a theoretical model that quantifies the effect of the alteration. We find that the overestimation can be severe. For a TCP window of 8 packets and a real MP of 0.15 the inferred MP overestimates by a factor of 4 [10]. In this section, we present our solution for

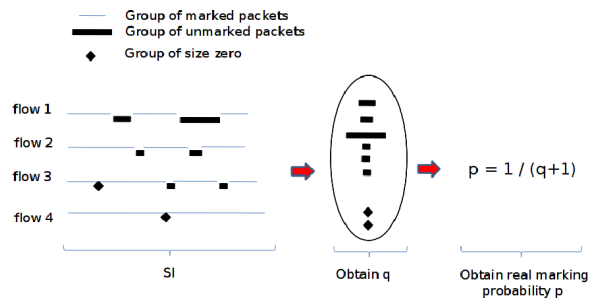


Fig. 2. The computation of a sample

inference using ACK packets. The key insight is that groups of unmarked packets can be leveraged instead of individual packet markings. Using the groups allows meaningful samples to be computed despite dealing with the altered versions of the initial congestion markings.

Figure 2 depicts our congestion inference solution. Routers monitor a limited number of TCP flows for each network path. In the figure, as a simple illustration, 4 out of many flows are monitored. During each SI, the length of each group of unmarked ACKs in each monitored TCP flow is measured. Then, an average group length is computed across all the monitored flows. Let q be this average size, p be the marking probability and n the possible size of groups of unmarked ACKs ranging from 0 to ∞ . The relation between q and p is:

$$q = \sum_{n=0}^{\infty} n * (1 - p)^n * p = \frac{1 - p}{p} \quad (1)$$

After q is obtained from counting the groups of unmarked ACKs, the sample p can be derived using equation (1).

Monitoring flows: Flows constantly begin and finish, therefore the set of monitored flows needs to be periodically refreshed. In this paper, we refresh the set of monitored flows at the beginning of every new EI by randomly selecting a new set of flows. If desired, other refresh intervals can also be used. For simplicity, in this paper, for every new EI we remove the information computed for the previous EI. A simple extension to our solution can allow information to be shared across EIs for flows that are chosen for monitoring in consecutive EIs.

At the beginning of each EI, the first few samples are not computed. Those first few SIs serve only to begin selecting a number of flows to monitor. In our evaluation we find that selecting flows during the first 2 SIs is sufficient for good accuracy. The samples from the subsequent SIs are averaged to obtain the estimate. During these subsequent SIs flows can still be chosen for monitoring until a desired maximum threshold is reached. These flows are used for inference starting with the SI following the SI where they were first encountered.

Incomplete groups: Groups of unmarked ACKs can span multiple SIs. During one EI, these groups are counted in the SI where they end. Even so, incomplete groups of unmarked ACKs may be encountered at the start or end of an EI. Such groups begin or end in a different EI. Since we treat each SI separately, the correct size of the incomplete groups cannot

be correctly measured and this can skew the results. Thus, incomplete groups are not considered for inference. It should now become clear why choosing very small EI values may impact accuracy. If for example $EI = SI$, an important percentage of groups will be incomplete and cannot be used by the inference. On the other hand, if the EI contains multiple SIs, fewer incomplete groups will appear since many larger groups will end and will be used for the inference.

Identifying groups of size zero: Our solution can leverage groups of unmarked ACKs of size zero. Recall that such groups are considered to appear when the CWR data packet that signals the TCP receiver to stop setting the ECE bit is also marked. To correctly identify groups of size zero a router remembers the sequence numbers of the last byte of the CWR packets it observes in the each of the monitored flows. The ACK corresponding to a CWR packet can be identified by its sequence number which is the first value larger than the remembered value. If both the ACK corresponding to a CWR packet and the previous ACK are marked, this signals the presence of a group of unmarked ACKs of size zero. To ensure that every ACK packet can be checked against its corresponding data packet, for each flow and EI, the sequence number of the first detected data packet is stored and only the ACKs with a greater sequence number are considered.

Benefit of groups of size zero: The use of the groups of size zero benefits the accuracy of our solution. As described, to use such groups, routers must be both on the forward and on the ACK path of a flow. However, it is important to note that our solution still works without the groups of size zero. If the groups of size zero are not used, a small change is required to (1). In that case, the MP can be computed as the inverse of the average group size of groups of positive size. The entire derivation is available in [10].

The increase in accuracy when using groups of size zero comes from the fact that in environments with high levels of congestion the number of groups of unmarked ACKs of size greater than zero alone may not provide statistical significance. The reason is that the ratio of groups of size zero is proportional to a path's MP and therefore becomes significant when the MP is large. Using groups of size zero does not, however, strictly require routing to be symmetric. If the first-hop and last-hop router for a flow are unique, as is usually the case, those routers will always be able to take advantage of the groups of size zero, irrespective of the degree of asymmetry of the entire end-to-end path.

Storage overhead: The state that routers need to store for our inference solution is small and comprised only of simple counters. For each flow, a counter keeps track of the size of the current group of unmarked ACKs. Over all flows, one counter holds the total size of all group of unmarked ACKs and a third counter keeps track of the number of groups. Additionally, if groups of size zero are used, for each flow, the sequence number of the last byte in the CWR packets needs to be stored. In the common case, only one CWR packet per RTT is expected for a given flow and it can be discarded after it is matched to the corresponding ACK.

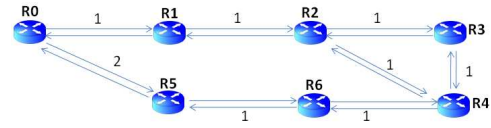


Fig. 3. A sample network with link weights

IV. INFERRING LINK LEVEL CONGESTION INFORMATION

We next describe the building blocks for computing link level estimates. Recall that a router's congestion map is the set of all inferred MPs. To support our explanations, we use the sample network in Figure 3 where routing is shortest-path. Unless otherwise stated, the route through R_5 is not used. We use S_{ij} to denote the path from router R_i to R_j . P_{ij} is the estimated MP over S_{ij} . If R_i and R_j are neighbors then $P_{ij} = L_{ij}$, where L_{ij} is the LMP.

A. Building Blocks

The first building block is the set of PMPs that a router can infer. These PMPs can then be further broken down using the second building block, a set of equations that directly result from leveraging link state routing information. We next characterize the set of PMPs that a router can infer when using our congestion inference solution.

Scenario 1 - Only data markings: A router R_i that observes only the data packets from the traffic sent by some source R_s can infer P_{si} . In other words, R_i can infer the MP on all paths that carry data traffic to it. In Figure 3, if R_0 and R_1 send traffic to R_4 , then, R_2 can infer P_{02} and $P_{12} = L_{12}$.

Scenario 2 - Only ACK markings - Groups of size zero not used: In this case, when groups of size zero are not used, a router R_i that observes only the ACKs from the traffic between a receiver R_d and a source R_s can infer P_{sd} as in §III-C. In other words, R_i can infer the MP over the entire forward path from R_s to R_d . In Figure 3, consider that R_0 sends traffic to R_4 using R_1 as a next hop but the reverse ACK traffic goes through R_5 . In this case, from ACK analysis, R_5 infers P_{04} .

Scenario 3 - ACK and data markings - Groups of size zero can be used: A router R_i that observes both the ACK and data packets sent between some sender R_s and some receiver R_d can infer both P_{si} and P_{sd} . P_{si} can be computed as in Scenario 1. P_{sd} can be computed as described in Scenario 2. In our example, suppose there is traffic from R_0 to R_4 and it is taking the route through R_1 . R_1 can infer P_{01} by analyzing data packet markings and P_{04} by analyzing ACK markings.

All scenarios described above regularly appear in practice. Let Fwd_{sd} and Rev_{sd} be the set of routers on the forward path and on the ACK path for traffic generated by R_s for R_d . If $R_i \in Fwd_{sd} \cap Rev_{sd}$ then Scenario 3 applies. If routing is asymmetric, then, a router can possibly find itself on either a forward path or an ACK path for which it is neither source nor destination. If $R_i \in Fwd_{sd} \setminus Rev_{sd}$ Scenario 1 applies. If $R_i \in Rev_{sd} \setminus Fwd_{sd}$ Scenario 2 applies.

The second building block is the method used for breaking down the inferred aggregate PMPs. Link-state routing information plays a vital role as it allows routers to compute the set

of links comprising paths with known PMPs. With this, PMPs can be further broken down into PMPs for shorter portions of the initial path. We can formally represent these ideas as follows.

A router R_i can compute the path level estimate P_{jk} if it knows P_{tj} and P_{tk} ($t = i$ usually, but not necessarily), and S_{tj} is a strict subset of S_{tk} :

$$P_{tj} + (1 - P_{tj})P_{jk} = P_{tk} \rightarrow P_{jk} = \frac{P_{tk} - P_{tj}}{1 - P_{tj}} \quad (2)$$

A similar equation can be derived if P_{jt} and P_{kt} are known instead. To exemplify the use of the equations, consider Scenario 3, where P_{si} and P_{sd} are inferred. Using equation (2) a router can compute P_{id} . For example, from P_{01} and P_{04} a router can derive P_{14} . In other words, in Scenario 3, R_i can now infer the MP on both the paths carrying traffic to it and away from it.

We use the model described above to approximate the effect of congestion marking in reality. In practice, the probabilistic marking and the inference errors could cause $P_{tj} > P_{tk}$. In those cases 0 should be used as the numerator in equation (2). The following evaluation section appraises the accuracy of our model in practice.

V. SIMULATION

A. Methodology

We conduct ns-2 simulations for evaluation because they provide us with fine granularity, router level congestion information to compare against. We next describe our default experimental setup. Exceptions are discussed with the corresponding experiments. The groups of size zero are used in all experiments.

Our solution considers each network path in isolation, infers path level congestion estimates and then breaks these down into link level estimates. In this section, we look at several scenarios that can appear on a network path. We then obtain link level estimates and use them to quantify the accuracy of our approach. Figure 4 describes our default 10 hop network path. Link bandwidth is limited to 500Mbps in order to keep the simulation times tractable. However, we also discuss the effects of changes in bandwidth. The propagation delay on each link in our network is 5ms. All the inferences presented are from the point of view of node R_0 .

AQM configuration: The function used by AQM algorithms to map congestion measures to MPs influences the inference accuracy. We evaluate the effect on inference of both the linear functions (PI [9], RED [12]) and exponential functions (REM [5]) present in ns-2 AQMs. By default, we use a linear marking function. We use RED to represent this group of functions since it is standardized and widely implemented [8][16]. For RED we disable the waiting between marked packets (the ns-2 `wait_` parameter) in order to be compliant with the RFC. We set the MP to linearly increase to 1 (`max_p_` in ns-2) as the average queue size grows to `max_thresh`. The queue measuring function and the marking probability are defined per-byte. The `min_thresh` is 25% and `max_thresh` is

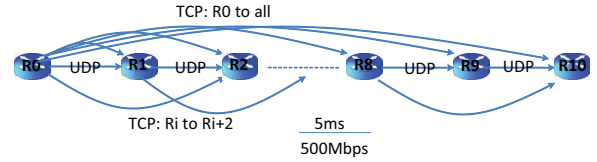


Fig. 4. A particular network path inside a network

75% of the router buffer size. As it is common today [4], the buffer size is the product of the link capacity and average network RTT.

Traffic: We use the ns-2 TCP Reno and UDP sources to generate traffic. Packet size is 500 bytes. The TCP flows used are FTP flows that once started last for the entire duration of the simulation. However, in §V-B4 we also evaluate the influence of small flows. There is TCP traffic from node R_0 to each of the other nodes:

$$N_{0 \rightarrow i} = Nr. \text{ Flows from } R_0 \text{ to } R_i = N * i^2 \quad (3)$$

N is tunable and by default is 250. $N_{0 \rightarrow i}$ increases with the distance from R_0 . We made this choice to minimize the bias that TCP has against flows with larger RTTs. Background traffic is simulated by 100 TCP flows started from any node R_i to node R_{i+2} . We also wish to understand the effect of congestion variation on the inference. We use UDP sources to induce variation in the congestion level. Compared to TCP sources, UDP sources offer more flexibility in creating congestion level variations because they do not adapt to the network conditions. We devised a custom UDP source that changes its sending rate by a percentage of the link bandwidth every second while continuously cycling between 0 and 500Mbps. The default value is 2% (10Mbps). Such sources are started between every consecutive pair of routers. Note that the use of UDP sources reduces the number of TCP markings available for inference. In reality, some networks may see a significantly smaller percentage of UDP traffic. In those cases the accuracy of our solutions can improve. All the TCP and UDP flows described are set up on the forward path. The ten links comprising our network path show average MPs for each EI ranging from a high of roughly 0.3 for (R_0, R_1) to 0.12 for (R_8, R_9) . The link (R_9, R_{10}) marks packets with roughly 0.05 probability.

Parameter values: Unless otherwise stated an SI of 0.5s (roughly our network's average RTT) and an EI of 3s are used. Each simulation runs for 500s. The results include the initial phase in which flows are started are congestion suddenly ramps up. R_0 monitors at most 1000 flows per EI for each R_i . Note that this is the maximum allowed; some EIs will observe less flows. As described in §III-C the first 2 SIs are not used for inference but rather for selecting an initial set of flows to monitor.

Metric: To quantify the inference accuracy we use the 50th and the 90th percentiles of the following metric:

$$\text{Accuracy Metric} = |\text{Inferred MP} - \text{Real MP}| \quad (4)$$

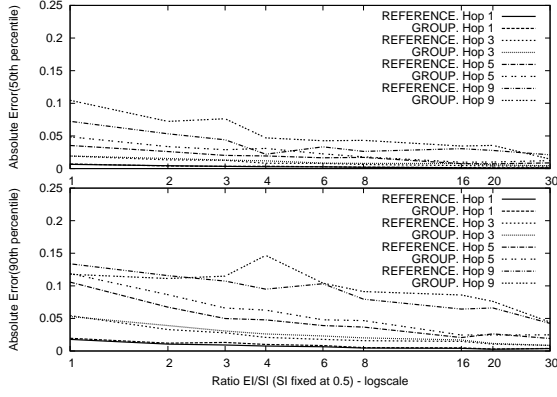


Fig. 5. Error vs ratio of EI and SI

Let EI_s and EI_f be the start and finish time of an EI used by R_0 for inferring the LMP on some link (R_i, R_j) using ACK markings. The estimate cannot be directly compared to the real MP on the link from time EI_s to EI_f because of the delay from (R_i, R_j) to R_0 . Therefore, we consider EI_a and EI_z be the times during some EI at which R_0 receives the first and last ACKs corresponding to data packets forwarded by R_j . $EI_s \leq EI_a \leq EI_z \leq EI_f$. Also let t_a and t_z be the times at which those data packets were forwarded by R_i . The real MP on (R_i, R_j) against which we compare is the time weighted average over all MPs at R_i over the interval that starts at $t_a - (EI_a - EI_s)$ and ends at $t_z + (EI_f - EI_z)$. If no packets are received from R_j during an EI the real MP is computed from time EI_s to EI_f .

We perform numerical comparisons between the inferred and real MPs. While in practice a discrete representation of congestion (e.g. low, medium) should suffice for most applications, performing direct numerical comparisons allow us to better present the strengths and limitations of our approach.

A reference solution: To assess any potential impact that altered markings have on the inference accuracy, we compare against a reference solution that does not suffer from alteration. This solution uses an alternative echoing scheme in which TCP receivers mark an ACK if and only if the corresponding data packet was marked. This basically reduces the problem of ACK packet inference to that of data packet inference where the sample can be calculated based on the ratio of marked packets. Therefore, even though in this paper we focus on ACK packet inference, the accuracy of inference using data packets can still be discerned from the accuracy of the reference solution.

B. Experimental Results

1) **Sensitivity to the length of the EI:** We analyze the effect that the EI length has on inference accuracy. The results are shown in Figure 5 for an SI of 0.5s. The x-axis is in logarithmic scale. Our solution's accuracy is comparable to that of the reference solution. We observe that the inference for small EI/SI ratios is more error-prone than for larger ratios. For small ratios fewer packets and groups are available for inference. The inference error also increases with the

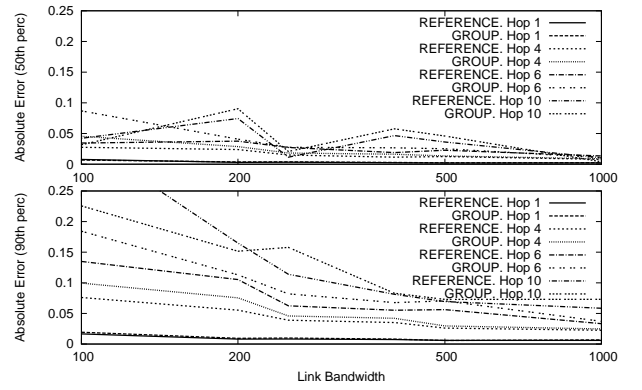


Fig. 6. High bandwidth environments

number of congested links. This is because each congested link introduces small variations in the aggregate MP as flows cannot all be marked with precisely the same probability. Uncongested links do not introduce such variations since their marking rate is 0. Nevertheless, even in such a network where every link is congested, for an EI of 3s (ratio 6) the 90th percentile of the error for the 5th hop is within 0.05. If an EI of 15s is used, the error for all hops is within 0.05. More importantly, even for very small EI values (1s), only the most distant hops (9th hop in this experiment) have an inference error of more than 0.1.

2) **Performance in high bandwidth environments:** We next analyze the sensitivity of the inference to an increase in bandwidth. We start with a link bandwidth of 100Mbps and go up to 1Gbps. Intuitively, an increase in bandwidth provides more groups for the inference. Results are shown in Figure 6. As bandwidth increases there are far more packets exchanged between R_0 and the other routers and therefore more groups of unmarked packets. Since the inference process benefits from more data points the accuracy increases with the bandwidth. Note that the improvement diminishes when bandwidth is scaled up to 1Gbps because the inference process is already provided with many data points. Note also the increased variability in the inference for the last hop which is caused by the relatively smaller number of packets and groups.

3) **Sensitivity to sudden changes:** The default behavior of RED is to smooth out variations in the congestion severity and therefore the MP. To test our solution in scenarios where the MP varies suddenly and substantially we instruct the UDP sources to change their sending rate by 50Mbps (10% of link bandwidth) every second while cycling between 250Mbps and 750Mbps (50%, 150% of link bandwidth). Every 10s we also stop the UDP sources for a random duration between 0s and 10s. Every 10s we start 3000 TCP flows between a random pair of nodes in the network. Each of these 3000 flows finishes after sending for a random time between 0s and 10s. The resulting MP variation at the EI scale for the second hop is presented in Figure 7. The other hops present a comparable pattern of variation. Packet drops are often encountered in this experiment.

The results are presented in Figure 8. Even with these

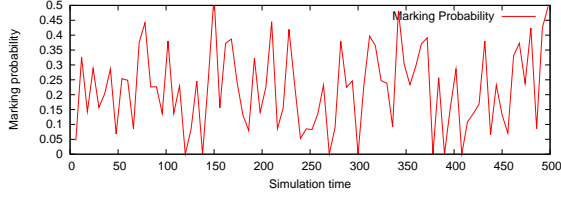


Fig. 7. MP created on the second hop by the scenario

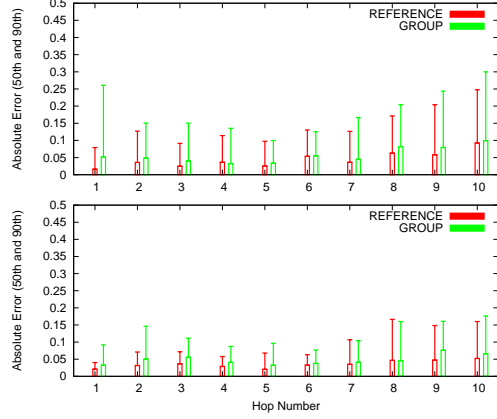


Fig. 8. Error for EI of 3 and 10 for each hop

sudden changes in the MP and with an EI of just 3s the 50th percentile of the error is still under 0.05 for the first seven hops. The inference for the 1st hop is affected by the small number of flows started to R_1 . As a result, very few flows are monitored from R_1 . As the EI is increased to 10s the accuracy improves to the point where the 90th percentile of the error is under 0.1 for most hops.

4) **Sensitivity to flow size:** In practice, a significant number of flows are small flows. To simulate such scenarios we consider a percentage of the total number of flows to be small flows. We use 10%, 50% and 90% as the values. We then limit the number of packets counted from these flows. If the limit is 2, then only the first two counted packets of a flow are considered for inference; exactly as if the flow had only two packets. The results are presented in Figure 9 for the 4th and 7th hop. Other hops show similar trends.

If 90% of the monitored flows are small flows, the effect on the accuracy become visible. However, even when 50% of the flows are small the inference result is good. This is because over all flows there are still enough groups for inference. We also evaluated the reference solution. As expected, the reference solution is less affected by the size of the flows because it uses packets instead of groups of packets. This experiment shows that accurate results can be obtained even when a significant number of flows are small flows.

5) **Robustness against false positives:** A specific case of measurement inaccuracy is inferring an uncongested link as congested. This overestimation can occur because of the probabilistic nature of our algorithm that can yield minute differences in the MP of flows on the same path. In this experiment we congest only the first network link and analyze the overestimation on the other links. To congest only (R_0, R_1)

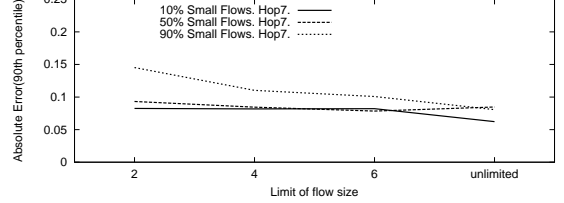
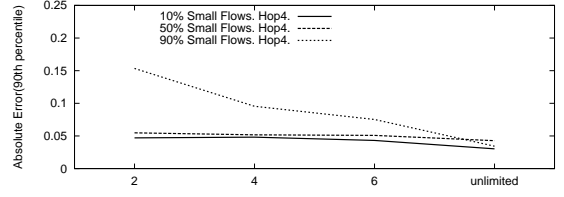


Fig. 9. Error vs % of small flows vs limit on flow size for our proposed, group-based solution

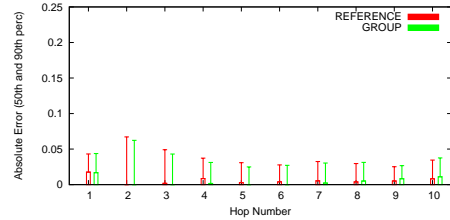


Fig. 10. Error on uncongested links vs hop count. Box is 50th percentile, error-bar is 90th percentile

we remove all traffic unless started from R_0 . We perform several runs with each run varying the MP on (R_0, R_1) by starting a variable number of flows from R_0 to R_1 and by varying the rate at which UDP sources send traffic from 50Mbps to 200Mbps in 25Mbps increments. In Figure 10 we present the inference errors for the uncongested links for a representative run. The overestimates are very small. If coarse congestion estimates (e.g low, high) are used, the severity of the congestion will be correctly inferred in most cases.

6) **Sensitivity to different AQM algorithms:** Alongside RED, we also evaluate REM and PI. The default ns-2 parameters are used for REM and PI. The inference accuracy depends on the function that the AQM algorithm uses for mapping congestion measures to MPs. For REM this is an exponential function. It creates abrupt variations in the MP for small changes in the congestion measure. RED and PI use a linear function. The results for our proposed solution are shown in Table I. The inference for the reference solution yields very similar results. PI exhibits good performance, similar to RED. REM, however, does not perform well. REM's exponential marking function is far more likely to suffer from an effect we call limited visibility, than the linear marking functions of RED and PI. For REM, in our experiment, the limited visibility is a factor as soon as the second hop. Limited visibility is discussed in §VI.

7) **Additional experiments:** Additional experiments are available in our technical report [10]. In one experiment we show that the variation in the delay from the inferring router to end-hosts has little effect on the inference accuracy in the common case when the delay variation is smaller than the EI

Hop	RED 50th	RED 90th	PI 50th	PI 90th	REM 50th	REM 90th
1	0.002	0.006	0.026	0.035	0.001	0.014
3	0.009	0.023	0.028	0.068	0.999	0.999
5	0.014	0.034	0.035	0.103	0.999	0.999
7	0.030	0.068	0.047	0.132	0.999	0.999
10	0.055	0.086	0.079	0.175	0.999	0.999

TABLE I
ABSOLUTE ERROR FOR DIFFERENT AQMS

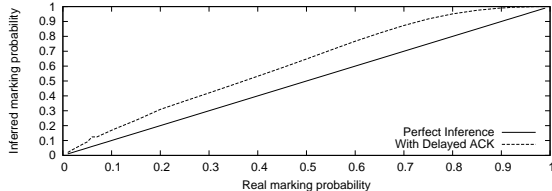


Fig. 11. Numerical analysis on effect of delayed ACK

length. In [10] we also tested the accuracy of our solution when groups of size zero are not used. We observed that, in a scenario with a bottleneck link with an MP under 0.5, the loss in accuracy is minimal. The accuracy decreases if the MP continues to increase because the proportion of groups of size zero becomes predominant.

VI. NUMERICAL ANALYSIS

Next, we perform a numerical analysis to understand the effect of two factors that can affect our inference solution by distorting the sequences of congestion markings in ACK packets: the TCP delayed ACK algorithm and packet loss. Moreover, we also analyze the limited visibility effect and using real topologies, we quantify the extent of the coverage of a router's congestion map.

A. Effects of Delayed ACK

The delayed ACK algorithm [7] allows a TCP receiver to acknowledge up to two data packets at once. In this case, the congestion marking echoed in the delayed ACK packet becomes the logical disjunction of the markings in the two data packets acknowledged [20]. This process alters the number and sizes of the sequences of unmarked ACKs compared to a TCP that does not use delayed ACK. For our numerical analysis we consider a large number of data packet markings set with some MP. An ACK marking is created for every two data packets according to the delayed ACK algorithm. Figure 11 shows the results. We compare against a perfect inference to point out the *additional* inaccuracy caused by using delayed ACK. Even though the delayed ACKs cause an overestimation in the inference, the severity of the inferred MPs is comparable to that of a perfect inference.

B. Sensitivity to Dropped Packets

With AQM/ECN networks packet loss should be rare because packet markings function as early congestion warnings for TCP. Nevertheless, we next analyze the effect of dropped data and ACK packets on the ACK based inference. In the

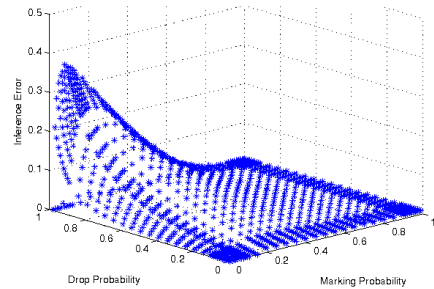


Fig. 12. Numerical analysis - additional inaccuracy caused by dropped ACKs

ideal case, the distribution and sizes of the sequences of unmarked ACKs that a router observes should be consistent with the aggregate MP on the corresponding forward path. Dropped ACK and data packets could affect these conditions. Drops may also limit the number of packets available to the inference algorithms.

Dropped data packets can appear when congested non ECN routers coexist with AQM/ECN enabled routers. Dropped data packets have negligible effect on the accuracy of our solution as long as enough groups are left for inference. The reason is that the probabilistic drops will not change the MP. Therefore, the sequences of unmarked ACKs will still be consistent with the MP. See [10] for a more detailed discussion and an experimental validation.

Dropped ACK packets: Pure ACKs are not ECN capable and can be dropped during congestion. Since our solution uses the sequences of ACKs for computation, pure ACK losses can impact the inference accuracy. In this experiment we analyze the degree of inaccuracy that ACK loss *adds* to a perfect inference. To this end, we use a numerical model where a large number of markings are created with some probability p and dropped with a probability d for flows with a window size of w . This model allows us to discuss results for all values of d , p and w whereas in a realistic simulation such parameters cannot be precisely controlled.

The worst case is for large w values. This case is pictured in Figure 12 for a w of 50 packets. The inaccuracy appears because the reduction in the number of unmarked ACKs due to the drops is not coupled with a comparable reduction in the number of groups and this skews the average group size calculation. If w decreases, the error also decreases because there is a greater chance of dropping all the markings between groups of unmarked packets. This will cause groups to unite which leads to fewer groups and therefore smaller errors compared to a larger w . Note that d must be in excess of 0.5 for the inference error to top 0.1. For comparison, for w of 4 packets d must be in excess of 0.65 for the error to top 0.1. Such high drop probabilities are never desired as they significantly degrade end-user performance.

C. Path Level Visibility

A very high aggregate MP on a path can make the inference of downstream links more challenging. To provide more insight into this effect, which we term *limited visibility*,

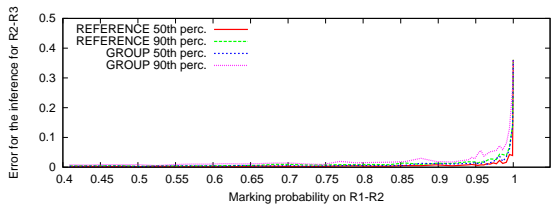


Fig. 13. The effect of limited visibility

we perform a numerical analysis. We use a simple network consisting of three links, from router R_0 to R_3 . We consider that flows are started from R_0 to all other routers, from R_1 to R_2 and from R_2 to R_3 . We fix the LMP on (R_2, R_3) at 0.36 (any value is equally good this illustration) and vary the LMP on (R_1, R_2) . The purpose of the analysis is to see the degradation in the inference for the link (R_2, R_3) when the LMP on (R_1, R_2) increases to 1.

The results are shown in Figure 13. The effect of limited visibility can be seen at high MPs. As the MP increases, and fewer and fewer packets leave router R_1 unmarked, it is increasingly more difficult to encode the LMP on (R_2, R_3) . In the extreme case (LMP of 1 on (R_1, R_2)) there are no more unmarked packets to carry useful information about (R_2, R_3) so R_0 considers the MP for (R_2, R_3) to be 0. The resulting error is 0.36, exactly the value of the LMP on (R_2, R_3) . A threshold at which the limited visibility commences depends on multiple factors including number of packets received, the aggregate MP on the paths and the variation of MP on the measured path. However, a significant level of congestion is necessary. In our simple experiment, the inference accuracy is impacted only when more than 95% of the packets are marked.

D. Extent of Congestion Map Coverage

If all-to-all traffic exists between the routers, the coverage of the congestion maps is high. Such a traffic pattern ensures that router R_i 's congestion map contains estimates for all the links that carry data traffic to and away from R_i . In this type of traffic pattern, R_i is a first hop for flows and can therefore analyze both their data and ACK packets. Therefore, as described in Scenario 3 from §IV-A, R_i can compute an estimate for the paths to and from all other routers in the network. Equation (2) can then be used to calculate the LMPs. The all-to-all traffic pattern is very common today. A large number of networks deploy routers to aggregate traffic from entire cities, and there is usually traffic flowing between any two cities.

To understand what percentage of the LMPs can be inferred in practice under an all-to-all traffic pattern, we analyze six real network topologies: Internet2, TEIN2 (Trans-Eurasia), iLight (Indiana), GEANT (Europe), SUNET (Sweden) and NLR (National LambdaRail) assuming shortest-path routing. We find that on average, the congestion map of a router from NLR, Internet2 and GEANT contains around 60% of LMPs. Under the assumed routing configuration the remaining LMPs are less important because the corresponding routers cannot carry traffic to the inferring routers. When those links

begin to carry traffic, their congestion level can be quickly inferred. Moreover, the maps of routers from TEIN2, iLight and SUNET contain on average as much as 91%, 94% and 95% of the LMPs.

As it can be seen from the results above, an inferred congestion map may not always contain all LMPs. One case is when markings from a link never reach an inferring router. For the example in Figure 3, the link (R_3, R_4) never carries packets from flows that reach R_0 , since it is not on the shortest path from R_0 to any router. Therefore, R_0 will not be able to infer $P_{34} = L_{34}$. Maps may also not contain all LMPs when the traffic pattern is not all-to-all. In these cases some PMPs cannot be completely broken down into LMPs. This can happen, when one of the routers on a path is not the first hop for traffic that reaches the inferring router. In Figure 3, suppose that the only traffic is from R_0 to R_5 and from R_0 to R_4 using R_5 as the first hop. While (R_6, R_4) does carry traffic to and from R_0 , R_0 will not be able to infer the LMP P_{64} because R_0 does not receive ACKs from R_6 . Nevertheless, R_0 can infer the PMP P_{54} . PMPs also provide useful information to applications.

We do not expect the use of Equal-Cost Multipath (ECMP) to limit the coverage of the maps as long as the rule used by a router to choose between the equal cost paths can be discerned by other routers. One such algorithm called hash-threshold has been proposed [14]. It hashes the packet header fields that identify a flow using a well known hash function (e.g. a CRC code).

VII. DISCUSSION

Incremental deployment: Universal deployment of AQM and ECN is not necessary for our approach to be effective. Our solution can be incrementally deployed. It can be deployed on specific AQM/ECN enabled paths in the network. Moreover, it can be deployed around non AQM/ECN enabled routers. The only condition is that the position of the non AQM/ECN enabled routers be known by all other routers in the network. Their presence can be factored out by the algorithm because their marking rate is effectively zero.

Deployment in heterogeneous environments: Using the MP as a congestion measure makes our solution applicable to a broad range of environments. One example are heterogeneous environments with multiple AQMs types or configurations. We do not require routers to use the same AQM algorithm nor the same parameters for the AQM algorithm. All that is needed is that congestion marking is used alongside an AQM algorithm that marks packets probabilistically as a function of a congestion measure.

Robustness under re-routing: After a re-routing decision, a link state routing protocol reliably disseminates new link state information into the network. When routers that use our solution obtain the new link state information they reset all the counters for the paths affected by the change and immediately start computing congestion estimates for the new paths using the markings from the re-routed flows. This ensures there is only a minimal interruption in the inference process.

VIII. RELATED WORK

The Re-ECN protocol [6] is a method for holding flows accountable for the congestion they create. It requires a non-standard use of header bits for TCP receivers to convey path level congestion information upstream. The TCP sources use an extra header bit to mark the data packets according to the information received. Policers placed on forward paths of flows can then use the source markings along with the ratio of ECN CE markings to obtain upstream and downstream path level congestion information. Policers can then detect misbehaving flows that create more congestion than permitted under current conditions. In contrast, our solution allows congestion information to be sent upstream without requiring any changes to either protocols or end-hosts. Moreover, the use of routing information allows us to obtain fine-grained link level congestion information whereas Re-ECN routers only obtain aggregate path level information.

DCTCP [2] is a TCP-like transport protocol for data centers that leverages ECN to provide feedback to end-hosts. In order to facilitate the transmission of congestion information from receivers to senders, DCTCP proposes changes to the ECN algorithm that allow the sender to recover all the markings seen by the receiver in the data packets. Our solution can provide the same functionality in transmitting congestion information without requiring changes to ECN.

Most methods developed for inferring the congestion severity are designed for TCP end-hosts and help them change their data sending rate according to current network conditions. TCP end-hosts are concerned only with the overall quality of the end-to-end paths. As a result, most of today's algorithms convey path level congestion information [23], [1]. Other approaches [3] report only the most congested link on a particular path. Our solution goes beyond these methods and uses the path level measurements to allow the routers to obtain fine grained link level congestion information.

Network tomography [11] deals with the inference of link properties from end-to-end path measurements. This usually requires solving a system of equations where the link characteristics are the unknowns. For the end-to-end measurements, active probing packets (multicast or unicast) are used. In contrast, our solution allows the measurement of path congestion in a completely passive fashion. Oftentimes, the problem that network tomography tries to solve is ill-posed. There are multiple possible solutions that can explain a set of end-to-end measurements. In contrast, the hop-by-hop nature of our solution allows in many scenarios the inference of unequivocal link level congestion information.

IX. CONCLUSIONS

We presented a novel approach that routers can use locally to passively infer a network-wide congestion map from markings already present in existing traffic. As a result, the maps are continuously updated with fine-grained real-time congestion information. Our solution can be leveraged by several distributed router level algorithms to allow more intelligent decisions to be made. Our solution is incrementally deployable

and can also be used in heterogeneous environments. We showed that the inference accuracy is good using numerical analysis and simulations on environments with multiple congestion points and sudden changes in the congestion pattern.

X. ACKNOWLEDGEMENTS

This research was sponsored by NSF CAREER Award CNS-0448546, NeTS FIND CNS-0721990, NeTS CNS-1018807, by an Alfred P. Sloan Research Fellowship, an IBM Faculty Award, and by Microsoft Corp. Views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of NSF, the Alfred P. Sloan Foundation, IBM Corp., Microsoft Corp., or the U.S. government.

REFERENCES

- [1] M. Adler, J. Cai, J.K. Shapiro, and D. Towsley. Estimation of congestion price using probabilistic packet marking. In *INFOCOM*, 2003.
- [2] M. Alizadeh, A. Greenberg, D. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. Dctcp: Efficient packet transport for the commoditized data center. In *SIGCOMM*, 2010.
- [3] L. L. H. Andrew, S. V. Hanly, S. Chan, and T. Cui. Adaptive deterministic packet marking. *IEEE Communication Letters*, 10(11):790–792, 2006.
- [4] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *SIGCOMM*, 2004.
- [5] S. Athuraliya, V. H. Li, S.H. Low, and Q. Yin. Rem: Active queue management. *IEEE Network*, May 2001.
- [6] B. Birsoe, A. Jacquet, C.C Gilfedder, A. Salvatori, A. Soppera, and M. Koyabe. Policing congestion response in an internetwork using re-feedback. In *SIGCOMM*, 2005.
- [7] R. Braden. *RFC 1122 - Requirements for Internet Hosts*, 1989.
- [8] Cisco. IOS Software Releases 11.1 Distributed WRED.
- [9] C.V.Hollot, V. Misra, D. Towsley, and W. Gong. On designing improved controllers for aqm routers supporting tcp flows. In *INFOCOM*, 2001.
- [10] F. Dinu and T.S.E. Ng. Gleaning network-wide congestion information from packet markings. In *Technical Report TR 10-08, Rice University*, July 2010. http://compsci.rice.edu/TR/TR_Download.cfm?SDID=277.
- [11] N. Duffield. Network tomography of binary network performance characteristics. *Information Theory, IEEE Transactions on*, 52(12):5373–5388, dec. 2006.
- [12] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.
- [13] J. He, M. Chiang, and J. Rexford. Towards robust multi-layer traffic engineering: Optimization of congestion control and routing. *IEEE J. on Selected Areas in Comm.*, 25, 2007.
- [14] C. Hopps. *RFC 2992 - Analysis of an Equal-Cost Multi-Path Algorithm*, 2000.
- [15] J. Hughes, T. Aura, and M. Bishop. Using conservation of flow as a security mechanism in network protocols. In *IEEE SSP*, 2000.
- [16] Juniper. JUNOS 9.1.x Quality of Service Configuration Guide.
- [17] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne. Fast ip network recovery using multiple routing configurations. In *INFOCOM*, 2006.
- [18] K. Lakshminarayanan, M. Caesar, M. Rangan, T. Anderson, S. Shenker, and I. Stoica. Achieving convergence-free routing using failure-carrying packets. In *SIGCOMM*, 2007.
- [19] A. Mizrak, Y. Cheng, K. Marzullo, and S. Savage. Fatih: Detecting and Isolating Malicious Routers. In *DSN*, 2005.
- [20] K. Ramakrishnan, S. Floyd, and D. Black. *RFC 3168 - The Addition of Explicit Congestion Notification to IP*, 2001.
- [21] A. Shaikh, L. Kalampoukas, A. Varma, and R. Dube. Routing stability in congested networks: Experimentation and analysis. In *SIGCOMM*, 2000.
- [22] M. Shand and S. Bryant. IP fast reroute framework. Internet Draft draft-ietf-rtwg-ipfr-framework-13.txt, October 2009.
- [23] R. Thommes and M. Coates. Deterministic packet marking for congestion price estimation. In *INFOCOM*, 2004.
- [24] Bo Zhang. *Efficient Traffic Trajectory Error Detection*. PhD dissertation, Rice University, Houston, TX, 2010.