Gateway Selection in Hybrid Wireless Networks through Cooperative Probing

Bong Jun Ko*, Sisi Liu§, Murtaza Zafer*, Ho Yin Starsky Wong*, and Kang-Won Lee*

*IBM T. J. Watson Research Center, {bongjun_ko, mzafer, hwong, kangwon}@us.ibm.com
§Broadcom, sisimm@email.arizona.edu

***Abstract*—In hybrid wireless networks (HWNs), where the fixed cellular network infrastructure is utilized to provide enhanced network coverage and communication performance for nodes in mobile ad-hoc networks, the selection of the gateway for each node towards the external network needs to be based on accurate and timely network performance perceived by each mobile node. Continuously monitoring these performance metrics by each individual node, however, would incur prohibitively high communication and processing overhead. In this paper, we propose a distributed network probing mechanism, called *CoPing*, that utilizes the cooperation among the nodes in the measurement process of path performance in MANETs towards the gateways of HWNs. In our approach, each node makes use of the end-to-end performance probing results measured by other nodes to estimate its own performance to the gateways in a fully distributed manner. Furthermore, the distributed process does not require explicit structure and coordination between the nodes, making it ideal for highly dynamic networking environments. Through the combination of analysis and experimental evaluation, we show our cooperative probing mechanism can achieve accurate and efficient path performance results for gateway selection in HWNs.**

I. Introduction

We consider a hybrid wireless network (HWN), where a fixed cellular network is federated with a mobile ad-hoc network (MANET) to provide better network coverage/performance and the connectivity towards external networks for mobile nodes. Often the majority of the network traffic in HWNs flows through the base stations, also called *gateways* herein, of the cellular network, and therefore the decision of which base station (BS) shall be utilized for each mobile node at a given time instance has a significant impact on the quality of the network performance perceived by individual mobile users. Given the dynamic nature and scarce bandwidth resources of mobile wireless networks, the following properties are desirable in the gateway selection mechanism:

- The selection is based on the accurate, up-to-date network performance metrics of the paths between each mobile node and each base station.
- The process of obtaining the path performance metrics does not incur prohibitively large networking and processing overhead.
- The mechanism can be used in conjunction with *any* underlying routing mechanism suited to given environments.

In this paper, we propose a novel mechanism called *CoPing* (Cooperative Pinging), with which the mobile nodes can cooperatively probe the network performance, such as latency, reliability, and throughput, to the gateways in HWNs. A main design goal of CoPing is to save the bandwidth consumed by the probe packets, yet yielding sufficiently accurate probing results, and we achieve this by letting each probing node cooperatively utilize the probing results of other nodes to estimate its own metrics. Specifically, a node's probe packets toward the gateways need not travel all the way to the gateways in CoPing; instead, some other nodes return the responses to the probing node on behalf of the gateways based on their own 'perception' of the path performance to the gateways. In other words, the cooperating nodes act as *proxies* for network probing to the probing nodes, while this probe-proxy relationship is fully distributed and can be mutual. Since pinging (i.e., ICMP echo request-response) is a basic yet fundamental building block for many probing technologies, we focus on the mechanism that provides pinging results cooperatively, hence the name *CoPing*.

Enacting the cooperation between network nodes usually requires coordination between them, and the coordination typically translates into additional communication cost. This can be problematic in HWNs since, not only it costs additional network resources for coordination, but the coordination should occur frequently to cope with network dynamics. In our context, the best results of the cooperative probing occur when a node A's probing result to a gateway S is estimated from the results of another node B placed between A and S. However, obtaining the knowledge of nodes' mutual position w.r.t. a particular gateway along the network path is quite challenging in HWNs because the network path changes frequently by the underlying routing protocol (many routing protocols for MANETs discover the path in an on-demand fashion [11], with some of them selecting the path even on a per-packet basis [2]). Our CoPing mechanism overcomes this challenge by enabling the cooperation between the nodes *implicitly* in a manner that requires *no coordination between the cooperating nodes* and also is *independent of the underlying routing protocol*.

We present the design of the CoPing mechanism and how this process can be realized in a fully distributed fashion using the standard ICMP echo response-reply protocols. Through experiments performed on a wireless network emulation plat-

978-3-901882-50-0 ©2013 IFIP

form, we evaluate the performance of CoPing in terms of path performance accuracy and its impacts on the gateway selection. In particular, we show the gateway selection accuracy can be improved by several orders of magnitude compared to a broadcast-based selection mechanism.

II. Related Work

Most existing approaches to gateway selection in wireless multi-hop networks adopt the principle (or its variations) in cellular networks [1], [3], [4]. In single-hop wireless cellular networks, the selection of the base station (or 'cell' to be precise) is based on the quality of wireless links (e.g., received signal strength, signal-to-noise ratio), which is measured continuously by each mobile device on wireless signals broadcast by multiple candidate base station.However, when applied to and extended for HWNs, such a process has several drawbacks because the network performance of individual base stations with respect to each mobile node should be measured over multi-hop wireless network paths. First, continuously broadcasting packets to be measured by mobile nodes throughout the network incurs a prohibitively large amount of traffic; it is reported that even a seemingly small amount of broadcast control traffic has multiplicative overhead effects [5]. Furthermore, it is well known that the path performance estimated via the measurement on broadcast packets is inaccurate [7], [8], even failing to correctly indicate the reachability over multi-hop paths due to the difference in the data rate and link coverage between the broadcast and point-to-point transmissions.

Alternatives to broadcasting mechanisms exist, which estimate the multi-hop path performance via some function of individual links as a means to routing decision, such as Expected Transmission Time (ETT) [9] and Expected Transmission Count (ETX) [6]. These approaches, however, are designed for static multi-hop wireless networks, still rely on propagating throughout the network the individual links' metrics measured by local broadcast, and, for the purpose of gateway selection, can be used only with the specific routing protocol and link measurement mechanism. In contrast, CoPing enables 'direct' measurements of the path performance from each mobile node's perspective and can be used in conjunction with *any* routing protocol of choice suitable for given environments.

Approaches utilizing the cooperation for network measurements exist in the literature. Tian et al. [14] presents a tracerouting method that exploits hierarchical structures of ISPs and leverages the collaboration of the probing vantage points. Stemm et al. [12] proposes a network measurement architecture in which network clients collaboratively share the measurement results to the Internet hosts through use of a local central repository. Though sharing a common philosophy with these approaches, CoPing addresses a unique problem of measuring real-time network path performance in highly dynamic environments, which existing work cannot adequately cope with.

III. Cooperative Probing

In this section, we describe the details of the cooperative probing process, CoPing, beginning by the high-level idea and motivation behind our design. Here we assume the performance metrics of interests are latency and packet loss rate; the extension for other metrics will be presented in Section IV.

A. CoPing Overview

The basic idea behind our scheme is to have the mobile nodes obtain the performance metrics to the gateways from the intermediate nodes in the paths to the gateways, and combine them with their own measurement results to those intermediate nodes. For example, suppose a node A is attempting to measure the latency to a gateway S, and a node B is in the routing path from A to S. We call B an *upstream node* with respect to A's path to S, and A a *downstream node* of B.

In a normal probing case, to measure the latency to the gateway, A would send probe packets (e.g., ping requests) to the gateway S and measure the RTT upon receiving the probe response from S (A can also measure the loss rate by counting the number of lost ping packets out of the total number of ping packets it sent).

In CoPing, however, instead of sending the probe packets all the way to the gateway S, a downstream node A sends the ping requests to an upstream node B, which responds to the ping request from A after "adding" the performance measure (i.e., delay in this example) from itself to the gateway S. When receiving the ping response from node B, A then measure the latency to B, and "adds" it with the latency from B to S to obtain the estimated latency to the gateway S. Note that this can be a recursive process, i.e., B's own performance metric to S itself can also be deduced from those of other nodes between B and S. This additive combining can be also applied to measuring the loss rate; the loss probability, p_{ABS} along a path from node A via B to the gateway S can be additively obtained (in its logarithm) from p_{AB} and p_{BS} (the loss probabilities in the path A to B, and from B to S, respectively) by $\log(1-p_{ABS}) = \log(1-p_{AB}) + \log(1-p_{BS})$, under the assumption that the loss is independent in each segment of the path.

Though conceptually simple, achieving such coordination in MANETs is a difficult prospect because of the dynamics of network topology and routing paths; The upstream-downstream relation between nodes is not fixed any more in MANETs, and even becomes unclear when highly dynamic routing mechanisms (e.g., opportunistic routing [2]) are employed. Furthermore, even if one could somehow find all intermediate nodes, these paths can change frequently due to node mobility, fading, interference, etc., rendering the attempts to find and organize the cooperative structure impractical.

Another issue is related to the part of the upstream node B's "addition" of the performance metric into its response to downstream node's probe measurement. While one could invent a new protocol for this cooperative behavior such that the additional measurement information can be included in the response by an intermediate node, this would require the changes in the standard protocol and the behavior of the ping client process upon receiving the response. Rather, we would like to use the standard ICMP protocol and its message format in the cooperative probing process.

Our solution to these issues is simple yet elegant. The first issue of finding the nodes for cooperation is addressed by simply exploiting the underlying routing mechanism when sending the probe requests, so that the upstream nodes are *implicitly* selected along the path the routing protocol delivers the packets toward the gateway; the downstream nodes do not even need to know or select who the upstream nodes are. In other words, we *piggyback* on the underlying routing protocol, whatever it is.

The second issue of realizing the cooperative process using the standard ping protocol is addressed by having the upstream nodes "simulate" the performance of the remaining path to the target gateway before responding to the ping requests from downstream nodes. This enables the downstream nodes to follow the standard measurement process, without having to explicitly combine the measurement results from the upstream nodes (it does not require the changes in the packet format or the behavior by the gateways, either).

In what follows, we describe in details the processes by the pinging clients (downstream nodes) and intermediate nodes (proxies); the behavior of the target ping servers (i.e., gateways) requires no change.

B. CoPing process

In CoPing, the cooperation among the nodes, i.e., the upstream nodes helping the downstream nodes estimate the path performance (latency and/or loss rate) to the gateways, is realized implicitly by the participating nodes without requiring any explicit effort for coordination between the nodes or additional information exchanged between the participating nodes. We assume for now that every node in the network participates in the cooperative probing process and that they periodically probe the path performance to a set of gateways. Later in this section, we will address the cases when not all nodes participate in the process.

Suppose a node A in a MANET wants to measure the latency and loss in the path from A to a gateway S in the candidate gateway set. In CoPing, the probing node A's behavior is simply the same as in the case of normal, non-cooperative probing. In other words, A simply sends a ping packet (i.e., ICMP echo request) to each gateway S in the candidate set, and monitors the response from each gateway to measure the path performance (latency and loss). The implication is that the ping request to the gateway S is delivered toward S by the underlying routing mechanism via multi-hop forwarding of other nodes in the network. Note that, due to dynamic routing mechanism in MANETs, it may be possible that the ping packets sent by A to S are delivered along different paths for different packets. Each probing nodes then stores the measurement results of the latest probing packets, for example, the recent N probe results, to derive the path performance metrics, such as the average delay and packet loss probability.

Then other (upstream) nodes along the paths from A to S perform the following process:

- First, when a ping request packet from A to S passes through B (i.e., B is an intermediate node forwarding the packet to S), B intercepts the packet.
- B then looks up for the latest probe result of its own to the gateway S, and simulates the performance for the remaining portion of the path from A to S based on the latest result. More specifically, if the latest probe result was the ping loss, B simply discards the ping request from A. Otherwise, B sends the ping response back to A after holding the response for the duration equal to the latency measured by its latest ping packet to S.

In other words, an upstream node B intercepts the ping request from other node A, and masquerades what the ping response by the original gateway at further upstream location would have been based on its own latest measurement for the gateway. Note that this process is inherently recursive: B's own probe requests to the gateways are also intercepted and responded by other nodes further upstream toward the gateways, hence the cooperative probing can take place in a multi-hop fashion.

Note also that this mechanism can naturally addresses dynamic routing issues because the intermediate nodes that respond to the probing node's ping request are determined implicitly by the underlying packet-forwarding mechanism. This observation is valid even for opportunistic routing protocols, which determines the packet forwarding paths on-the-fly in a per-packet basis, or multi-path routing protocols that carry multiple copies of the packets along multiple paths (the probing node would simply need to take the best or average performance if it receives multiple responses for the same request).

The benefit of bandwidth saving, however, comes with small price: The response from an intermediate node, B, is delayed (or discarded) based on B's own measurement made at a recent time in the past w.r.t. to the time that the original ping request from the source node passes through B. If the network condition changes frequently, this difference in measurement time can result in an error term in the measurement result of the probing node, compared to the original, non-cooperative probing.

We will investigate the impact of this potential error in the cooperative probing process later in the paper analytically and experimentally. A good news is that the probing results in most cases of the gateway selection are taken after statistical averaging of multiple of them to eliminate the measurement noise, after which the error term in individual probe result is reduced quite significantly. Moreover, in the context of gateway selection, these results are only used to *comparatively* assess the performance for a set of gateways, hence the small errors in the measurements do not impact the accuracy of the gateway selection much.

C. Controlling the Overhead of CoPing

Suppose every single node in the network participates in the above CoPing process. Then a node's ping request to any gateway would be intercepted and responded by the very next-hop node towards the gateway. Clearly, this is a best case scenario in terms of bandwidth saving aspects of the cooperative probing; However, it is also a worst case in terms of the measurement error caused at each hop of cooperative response which takes place in every single hop along the path to the gateway recursively.

Therefore, it is desirable to have a mechanism embedded in CoPing process that can control the balance between these two conflicting features. One of such mechanism, and a conventional one, is to adjust the frequency at which the periodic probing is performed by each node, such that, by increasing the frequency of sending the probes, the time difference between the arrival time of the ping request at an intermediate node and the time the last measurement made can be reduced, hence potentially reducing the error term in the probing result, at the expense of increased traffic volume of probing packets (or vice versa). Besides this method which can be applied in most probing techniques, another mechanism specific to our cooperative probing process is to control the distance that each probe packet travels in the network before it gets intercepted by an intermediate node (the farther the probes travel, the smaller the measurement error is, yet the smaller the bandwidth saving is).

We present here two approaches to controlling the hop distance of probe packets–both are again fully distributed mechanism that requires no explicit coordination between the nodes. In the first approach, the hop distance in the network that a ping request travels is controlled by the probing source. Specifically, suppose a node, A, wants its probe packet toward a particular gateway S to reach an intermediate node at least K hops away from it. The mechanism uses the TTL (Time-To-Live) field of IP packets as follows:

- Node A sets the TTL field of the outgoing ping request to K, and sends it to the gateway S.
- All nodes intercept the ping requests only when their TTL values have reached to zero.

The significance of this approach is that the CoPing processes of the nodes within K hops from the probing source A are oblivious of the ping request sent by A, since the underlying IP stack would decrement the value in the TTL at each hop until the packet reaches the nodes at the desired number of hops.[1] This mechanism leverages the inherent functionality in IP forwarding mechanism, and hence is immediately available for MANET routing protocols operating in IP layer (See Section IV for how to realize this with layer-2-based forwarding mechanisms).

The second approach is based on the decision by the intermediate nodes. An effect identical to the first method could be achieved by letting the non-source nodes intercept the ping request packets of their TTL=$K_{max} - K$ where K_{max} is the default TTL value set by the TCP/IP stack of the source node. Unfortunately however, different operating systems use different default TTL values (e.g., 64 in Linux, 128 in MS Windows NT 4.0), rendering this approach impractical unless the OS of all nodes are known in advance. Therefore, we instead utilize a probabilistic method, where nodes intercept the ping requests from others with some probability p, to indirectly control the hop distances of the probe packets; the probe packets tend to reach farther nodes when the value of p is smaller (and vice versa).

Note that both approaches have their pros and cons. The source-based approach can have fine control on the hop distances of the packets (it is also possible to control the distance in a per-packet basis). However, limiting the TTL to a certain low value has a risk of losing the probe packets all together when not all the nodes are participating in the CoPing process (imagine the TTL has reached zero but the corresponding node does not capture the request packet). On the other hand, the second approach based on the decision of intermediate nodes does not suffer from this risk, because, even if a node does not intercept the probe packet, the packet would still travel toward the gateway, leaving the possibility of other cooperating nodes intercepting it. But it is difficult to accurately adjust the hop distances as the interception takes place probabilistically at each node independently.

IV. Implementation and Extension

A. Implementation of CoPing

As described in Section III, the CoPing uses the standard ICMP echo requests and responses, and the behaviors of the ping clients and servers do not need any changes. The part specific for CoPing is the processes that involve (i) intercepting a client node's ICMP echo request, (ii) simulating the response to it, and (iii) responding back to the client on behalf of the ping servers. One additional requirement is that, since CoPing's primary objective is to provide the real-time path performance to a few gateways in the gateway selection mechanism, each CoPing node periodically shall probe the performance to those specified gateways periodically and remember the latest response from each of them, which is used later when it responds to other node's probe request.

The CoPing process is implemented as a process in Linux-based systems, where the ICMP echo request and response is processed in a user-space program, while the packet interception and masquerading is performed in the network kernel space. We assume that each mobile node is able to obtain a candidate gateway set of given time using existing gateway discovery mechanisms, for example, scoped broadcast by the gateways [10], which takes place at much lower frequency than the probing.

1) Start-up and periodic behavior: When the CoPing process starts up, it first (i) stores the target gateways' addresses to periodically ping, (ii) starts a periodic timer for sending

[1] In practice, A can set TTL equal to $K+1$ and the intercepting nodes can capture the packets when TTL=1, because the IP packets of TTL=0 can be silently discarded by the IP stack.

ping request to each gateway in a specified interval, and (iii) installs the ping packet interception rule.

Then periodically, it sends the ICMP echo request to each gateway in the current candidate set, and monitors the ICMP echo responses. It also records the result (latency or packet loss) of the latest ICMP echo request from each gateway, so that it can later use the result in the event it responds to other nodes' requests.

2) Intercepting ICMP echo request: The ICMP echo request packets are intercepted and passed to CoPing's user-space processing mechanism via netfilter.[2] The netfilter provides "hooks" that in various points of packet traversal within the kernel's IP stack, such that customized processing of the IP packets can be installed in the netfilter hooks. One of such hooks is located at the point called "PREROUTING", which enables us to install customized rules for processing the incoming IP packets before they enter the decision of whether to forward the packets to other nodes or pass it to the local process. When a CoPing process starts up, it installs a rule that sends the incoming IP packets to the specified local process if the packet type is ICMP and the ICMP type is echo request. If the TTL value of the ICMP packets are used by the probing source to control the probing hop distance, the incoming packets are also matched based on the value in the TTL field so only those with TTL expired will be intercepted. If the underlying MANET routing protocol uses packet forwarding in layer 2, the CoPing node installs an equivalent rule in Linux's layer-2 packet filter, br-netfilter.[3]

3) Processing ICMP echo request: When an ICMP echo request from a source A to a target C is intercepted, the CoPing process either (i) discards the packet if the latest result to the same target C is the ping loss, or (ii) sends ICMP echo response to A after delaying the same amount of time that is indicated in the latest ICMP echo response from the target C. Recall that, when sending the ICMP echo response to A, the CoPing process needs to set the source address of the response packet as the target address C, so that A would take the response as if it came from the original target C. This address masquerading is achieved by a connection tracking mechanism in Linux network kernel, called conntrack.[4]

B. Measurement of other path performances

Though the CoPing process is presented in previous sections in the context of measuring simple metrics like the latency and loss rate, the same principle can be also applied to measure more sophisticated methods in a cooperative manner. Take for example `traceroute`, which discovers the network routes from the probing source to the destination. Traceroute "traces" the path by repeatedly issuing ICMP packets with the TTL value of the packets incremented at each step, and then recording the nodes sending the ICMP "Time Exceeded" responses as the routers along the path to the destination. One of the major issues in using this tool in MANETs is that many probing packets are successively injected into the network to discover the path to a single destination. The cooperative mechanism for traceroute can be easily designed following the design principle of CoPing, such that an intermediate node responds to the traceroute probe requests, on behalf of other nodes further upstream along the path toward the target node.

The cooperative method can be also applied in the estimation of the available bandwidth of a path by the use of packet pair, or packet 'trains' [13], where the arrival time differences of the responses to the ping packets sent back-to-back are used to estimate in a statistical analysis based on queueing principle. In principle, since the queuing delay along the multi-hop path is an additive measure at each hop, the responses to the back-to-back ping requests can be cooperatively handled by an intermediate node, which simulates the response of the target host based on the measurement result of the time differences of the responses to its own packet trains to the target.

V. Error Analysis

In this section, we analytically investigate the errors in the latency and loss rate measures by CoPing versus what would be obtained by a native, non-cooperative probing method.

Let $x_{i,j}(t)$ denote the path performance from a node i to another node j at time t, representing either the latency between i and j or the logarithm of the packet delivery success ratio, i.e., $\log(1 - p_{i,j}(t))$, where $p_{i,j}(t)$ is the packet loss probability between i and j at time t. We assume that $x_{i,j}(t)$ is an independent, wide-sense stationary random process, with mean $\mu_{i,j}$, standard deviation $\sigma_{i,j}$, and autocorrelation $R_{i,j}(\tau)$.

Suppose that the path from i to s at time t includes an intermediate node k. Note that the measure $x_{i,s}(t)$ is additive along the packet path, i.e., $x_{i,s}(t) = x_{i,k}(t) + x_{k,s}(t)$. For the purpose of analysis we will assume a 2-hop case where k is the node that responds to i's ping request sent at time t, using its own measurement to s made at some time τ before the actual measurement time t of i. Then i's path measurement based on the simulated response of k will be $\hat{x}_{i,s}(t) = x_{i,k}(t) + x_{k,s}(t-\tau)$. Therefore, $\hat{x}_{i,s}(t) - x_{i,s}(t) = x_{k,s}(t-\tau) - x_{k,s}(t)$.

Therefore, it follows that $E[\hat{x}_{i,s}(t) - x_{i,s}(t)]$=0, and

$$E[(\hat{x}_{i,s}(t) - x_{i,s}(t))^2] = E[(x_{k,s}(t-\tau) - x_{k,s}(t))^2] \quad (1)$$

$$= 2\sigma_{k,s}^2 \left(1 - R_{k,s}(\tau)\right). \quad (2)$$

Suppose each node performs the probing periodically at every h time unit. Then, since $\tau \leq h$, assuming $R_{k,s}(\tau)$ is a positive non-increasing function of $\tau > 0$,

$$E[(\hat{x}_{i,s}(t) - x_{i,s}(t))^2] \leq 2\sigma_{k,s}^2 \left(1 - R_{k,s}(h)\right) \quad (3)$$

Now let us denote by $y_{i,s}(t)$ and $\hat{y}_{i,s}(t)$ the running averages of N recent $x_{i,s}(t)$'s and $\hat{x}_{i,s}(t)$'s, respectively,

$$y_{i,s}(t) = \frac{1}{N} \sum_{n=0}^{N-1} x_{i,s}(t - nh) \quad (4)$$

[2]http://www.netfilter.org.
[3]http://ebtables.sourceforge.net.
[4]http://conntrack-tools.netfilter.org.

$$\hat{y}_{i,s}(t) = \frac{1}{N}\sum_{n=0}^{N-1}\hat{x}_{i,s}(t-nh) \quad (5)$$

Then, the mean error is,

$$E[\hat{y}_{i,s}(t) - y_{i,s}(t)] = \frac{1}{N}\sum_{n=0}^{N-1} E[\hat{x}_{i,s}(t-nh) - x_{i,s}(t-nh)] = 0, \quad (6)$$

and the mean-square error is,

$$\begin{aligned} &E[(\hat{y}_{i,s}(t) - y_{i,s}(t))^2] \\ &= \frac{1}{N^2}E\left[\left(\sum_{n=0}^{N-1}(x_{k,s}(t-nh-\tau_n) - x_{k,s}(t-nh))\right)^2\right] \end{aligned} \quad (7)$$

where τ_n denotes the difference between the time k receives the ping request sent by i at time $t-nh$ and the time k has obtained the last result before $t-nh$.

Lemma 1: The mean square error given in (7) goes to zero for large N if the auto-correlation function, $R_{k,s}(\tau)$, satisfies, $R_{k,s}(\tau) = 0, \tau \geq lh$ for some l that satisfies $l = o(N)$.

Proof: Consider the expression given in equation (7); expanding it gives,

$$\begin{aligned} &E[(\hat{y}_{i,s}(t) - y_{i,s}(t))^2] \\ &= \frac{1}{N^2}\sum_{n=0}^{N-1} E(x_{k,s}(t-nh-\tau_n) - x_{k,s}(t-nh))^2 + \\ &\frac{2}{N^2}\sum_{i=0}^{N-2}\sum_{j=i+1}^{N-1} E\Big\{(x_{k,s}(t-ih-\tau_i) - x_{k,s}(t-ih)) * \\ &\qquad (x_{k,s}(t-jh-\tau_j) - x_{k,s}(t-jh))\Big\} \end{aligned}$$

Expanding and re-writing in terms of the auto-correlation function gives,

$$\begin{aligned} &E[(\hat{y}_{i,s}(t) - y_{i,s}(t))^2] \\ &= \frac{2\sigma_{k,s}^2}{N^2}\sum_{n=0}^{N-1}(1 - R_{k,s}(\tau_n)) + \\ &\frac{2}{N^2}\sum_{i=0}^{N-2}\sum_{j=i+1}^{N-1}\sigma_{k,s}^2\Big\{R_{k,s}((j-i)h + \tau_j - \tau_i) \\ &\qquad + R_{k,s}((j-i)h) - R_{k,s}((j-i)h - \tau_i) \\ &\qquad - R_{k,s}((j-i)h + \tau_j)\Big\} \end{aligned}$$

Since $R_{k,s}(\tau) = 0, \tau \geq lh$, and $-1 \leq R_{k,s}(\tau) \leq 1$, for all τ, we get,

$$\begin{aligned} &E[(\hat{y}_{i,s}(t) - y_{i,s}(t))^2] \\ &\leq \frac{2\sigma_{k,s}^2}{N^2}\sum_{n=0}^{N-1}(1 - R_{k,s}(\tau_n)) + \frac{2}{N^2}\sum_{i=0}^{N-2}\sum_{j=i+1}^{l} 4\sigma_{k,s}^2 \\ &\leq \frac{4\sigma_{k,s}^2}{N}(1 + 2l) \end{aligned}$$

From above, since $l = o(N)$, the result follows. ∎

Thus, as intuitively expected, the measurement error between the running averages decreases with the averaging window size.

VI. Performance Evaluation

A. Experimental setup

We evaluate the performance of the CoPing on top of CORE (Common Open Research Emulator).[5] CORE is a network emulation platform in which each network node runs the real process and networking stacks in real time using a container-based virtualization, where the network connectivity, wireless signal propagation, and node mobility are simulated based on the models.

We run a few experiments, each consisting of 30 mobile nodes and 5 gateways. The mobile nodes are initially placed at random locations in a 1000x1000 m^2 area, and the gateways are placed evenly at the top row of the area. To see the effects of the nodes' mobility, and hence the impact of network partitioning and dynamic route changes, we move the mobile nodes according to random walk model of low and high speeds (2m/s and 15m/s, respectively). The wireless connectivity and packet transmission is driven by the 802.11b MAC/PHY model provided in EMANE (Extendable Mobile Ad-hoc Network Emulator).[6] MANET extension of OSPFv3[7] is used as the routing protocol.

Each mobile node runs CoPing process, probing each gateway at every 5 seconds, and measures the average of N recent ping results as the metrics of the real-time path performance to each gateway, i.e., the running average latency and the packet loss ratio. Then each node uses these metrics to compare the gateways and select the best one (i.e., lowest average latency or lowest packet loss ratio). The experiments are run for 10 minutes for each scenario in the steady states.

The wireless environments and the density of the nodes in our experimental scenario is such that the connectivity between the nodes is quite low, resulting in frequent disruption of the packet delivery between the nodes along multi-hop paths. In such circumstances, the primary measure of the gateway selection becomes rather the connection reliability, represented by the packet loss probability, than the latency performance. Hence, in our evaluation, we focus our attention on the loss probability measured by the CoPing process. Specifically, we evaluate the error in the estimated loss probability, p, along a path, compared to the ground-truth loss probability, p^* obtained from non-cooperative pinging process, where p and p^* are obtained from N recent pinging responses.

For the gateway selection results, we denote by S^* the set of best gateways selected by the native ping method (this is the ground truth), and by S_{coop} the set of the gateways selected by CoPing at the same time (each of these two sets typically contains only up to 1 element due to the continuous

[5] http://cs.itd.nrl.navy.mil/work/core/.
[6] http://labs.cengen.com/emane/.
[7] http://www.ietf/rfc/rfc5614.txt.

metric value, and is empty if all gateways are not reachable). For comparison, we also include the results of the set of the gateways selected based on the hop count indicated in the routing table populated by broadcast routing protocol, denoted by S_{RT}. The accuracy of a gateway selection S (S is either S_{coop} or S_{RT} is measured by the following two metrics

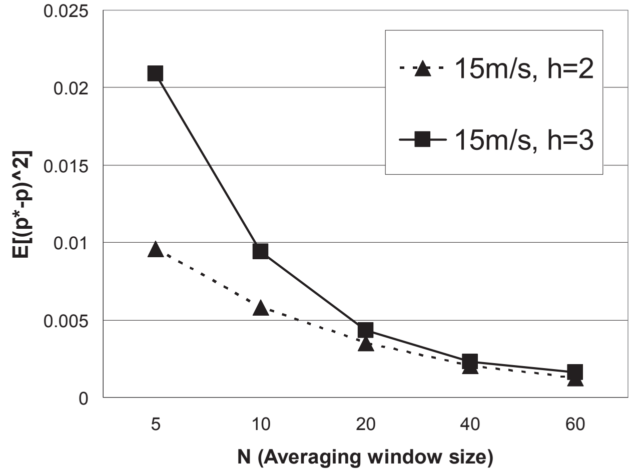
- False positive ratio, FPR_S, of a gateway selection S is the average of the number of gateways that S erroneously includes (i.e., $S - S^*$ to the number of elements in S, i.e., $FPR_S = \frac{|S-S^*|}{|S|}$, out of the times when S^* is not empty.
- False negative ratio, FNR_S, of a selection S is the average fraction of the number of gateways that S fails to include out of those in S^*, i.e., $FPR_S = \frac{|S^*-S|}{|S^*|}$

Again, we assume a gateway to which the packet loss probability is the lowest is selected by the client based on the either measurement of p and p^*.

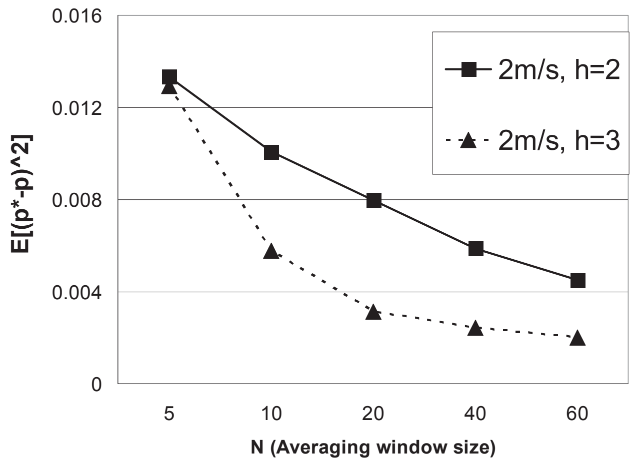
B. Experimental results

1) Measurement error: Figure 1 shows the variance of the error in the loss probability, i.e., $E[(p^* - p)^2]$, under two different scenarios: the high-mobility one (Figure 1(a)) and the low-mobility case (Figure 1(b)). Each curve in the plots represents the errors separately for different number of hops between the CoPing clients and the gateways, showing the impact of the number of intermediate nodes that are involved in the cooperative probing process from a client to the gateway. The x-axis is varied by the size of the averaging window, N. We expect the error will generally increase as the number of hops from the client to the gateway increases, because each intermediate node's response to the client's request based on its own recent measurement introduces error terms at each hop of cooperation. In the high mobility case in Figure 1(a), this can be indeed observed because, in general, as the network changes frequently, the connectivity measure between a client and a gateway is also highly impacted, making the response based on some time in the past vary more widely from the actual performance at the time the request is issued. In the low-mobility scenario (Figure 1(b)), the situation is reversed rather surprisingly. This can be accounted for by the fact that the network is quite stable, so the estimation by cooperative behavior does not impact the resulting performance measure much. In both cases, we can observe the error decreases as the averaging window size N increases, as expected from our analysis in Section V.

2) Gateway selection accuracy: We compare the gateway selection result by CoPing mechanism against that by the original ping method, the latter constituting the ground truth. The false positive ratio and the false negative ratios are measured. For a comparison, we include the performance of gateway selection results based on the hop count between the client and gateways, such that one of the gateways of the lowest hop count is the selected. The hop count is a measure that is readily available from the routing table without any probing process. The problem is however, it represents only a low-granularity metric, is typically obtained through broadcast packets (which



(a) Mobility speed=15m/s

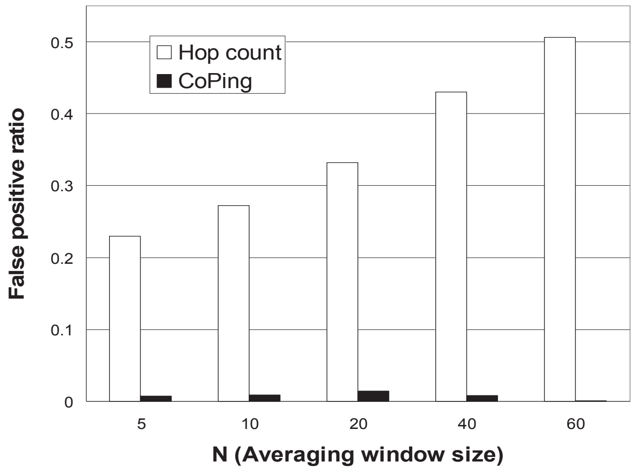


(b) Mobility speed=2m/s

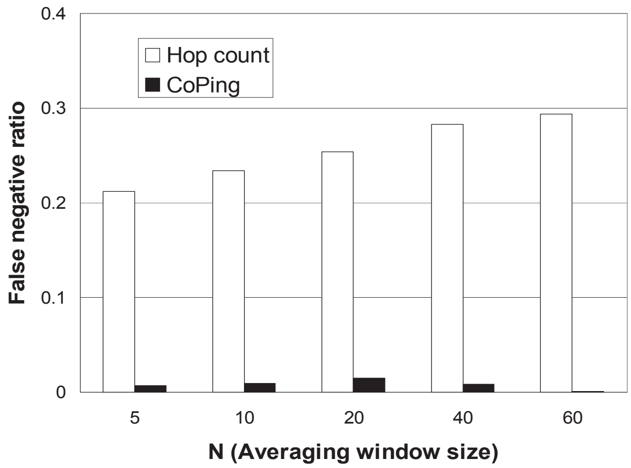
Fig. 1. Packet loss ratio measurement error due to cooperative probing vs. non-cooperative probing $E[(p^* - p)^2]$

is inaccurate as a measure for unicast QoS), and does not reflect the actual path performance in many cases. Also, many routing protocols for MANET discover the routes on demand, and do not maintain the routing tables. Nonetheless, we evaluate the gateway selection via the hop counts because our CoPing's probing overhead is at a level comparable with the broadcast-based route discovery, particularly when all nodes run CoPing process and probe packets are responded by a one hop neighbors toward the gateways.

Figures 2 and 3 show the false positive and false negative ratios of the gateway selection due to our CoPing mechanism and by hop count. The benefit of using CoPing process in the gateway selection is fairly dramatic: While the CoPing process happens at every hop of the routing path in our experiment, limiting the distance the probe packets travel only to the immediate next hop (similarly to the routing protocols' periodic neighbor discovery messages), the resulting accuracy of the gateway selection by CoPing mechanism is better by several orders of magnitude than that by the hop count

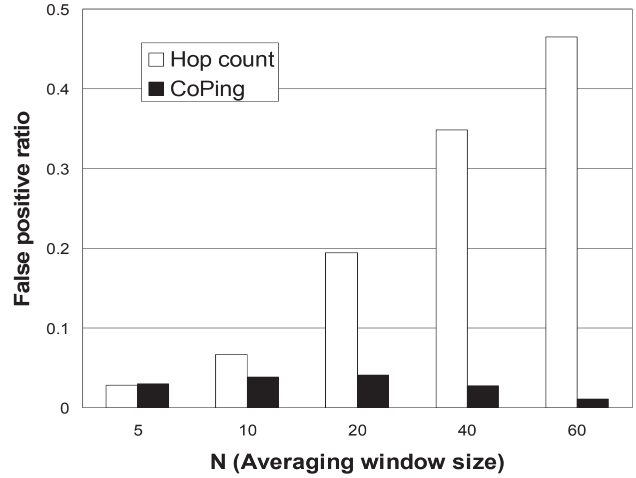


(a) False positive ratios

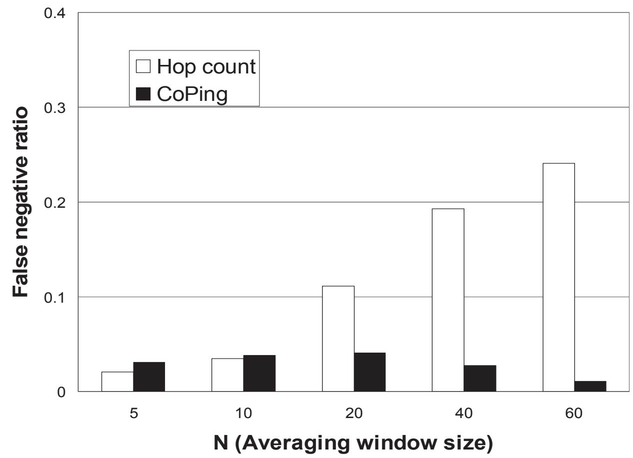


(b) False negative ratios

Fig. 2. False positive ratios and false negative ratios of gateway selection: mobility speed=2m/s, number of hops from clients to gateways=2



(a) False positive ratios



(b) False negative ratios

Fig. 3. False positive ratios and false negative ratios of gateway selection: mobility speed=15m/s, number of hops from clients to gateways=3

measures, in both cases of high and low mobility. This shows the effectiveness of the CoPing process in saving the network bandwidth consumption while achieving the accurate results in the gateway selection (the small errors introduced by the cooperative measurement does not impact the gateway selection results much). Again, the accuracy improves as N increases, due to the decreased error in the measurement result.

VII. Conclusion

In this paper, we present a novel cooperative probing mechanism, called CoPing, for measuring the path performance to the gateways in HWNs. The CoPing mechanism leverages a fully distributed process of the mobile nodes acting as network measurement proxies for each other. The significance of this principle is that the network bandwidth consumption can be saved significantly while the nodes can still obtain fairly accurate network performance. Through the analysis and experiments, we show this process is quite effective in achieving its goal of bandwidth saving and providing up-to-date real-time network performance metrics to be used in the gateway selection process. In the future, we plan to further investigate the problem of optimal and adaptive selection of CoPing parameters, such as probing frequency and probing hop distances.

Acknowledgements This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] Usman Ashraf, Slim Abdellatif, and Guy Juanole. Gateway selection in backbone wireless mesh networks. In *WCNC*, 2009.

[2] Sanjit Biswas and Robert Morris. ExOR: Opportunistic multihop routing for wireless networks. In *Proc. of ACM SIGCOMM'05 Conference*, Pennsylvania, USA, Aug. 2005.

[3] S. H. Bouk and I. Sasase. Multiple end-to-end qos metrics gateway selection scheme in mobile ad hoc networks. In *International Conference on Emerging Technologies*, 2009.

[4] M. Boushaba and A. Hafid. Best path to best gateway scheme for multichannel multi-interface wireless mesh networks. In *WCNC*, 2011.

[5] J. Camp, Nancuso V, O. Gurewitz, and E. W. Knightly. A measurement study of multiplicative overhead effects in wireless networks. In *INFOCOM 2008*.

[6] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03*.

[7] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, and Robert Morris. Performance of multihop wireless networks: shortest path is not enough. *SIGCOMM Comput. Commun. Rev.*, 33(1):83–88, January 2003.

[8] Richard Draves, Jitendra Padhye, and Brian Zill. Comparison of routing metrics for static multi-hop wireless networks. In *SIGCOMM '04*.

[9] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom '04*.

[10] U. Javaid, F. Rasheed, D. Meddour, and T. Ahmed. Adaptive distributed gateway discovery in hybrid wireless networks. In *WCNC*, 2008.

[11] C. E. Perkins and E. M Royer. Ad-hoc on-demand distance vector routing. In *WMCA '99*.

[12] Mark Stemm, Srinivasan Seshan, and Randy H. Katz. A network measurement architecture for adaptive applications. In *INFOCOM 2000*.

[13] Jacob strauss, Dina Katabi, and Fran Kaashoek. A measurement study of available bandwidth estimation tools. In *IMC '03*.

[14] Y. Tian, R. Dey, Y. Liu, and K. W. Ross. China's internet: Topology mapping and geolocating. In *INFOCOM 2012*.