Coding-Aware Multi-path Routing in Multi-Hop Wireless Networks

Song Han*, Zifei Zhong*, Hongxing Li†, Guihai Chen†, Edward Chan‡, Aloysius K. Mok*

* Department of Computer Sciences, The University of Texas at Austin
{*shan, zfz, mok*}@*cs.utexas.edu*
† State Key Laboratory of Novel Software Technology, Nanjing University
{*hxli, gchen*}@*dislab.nju.edu.cn*
‡ Department of Computer Science, City University of Hong Kong
{*csedchan*}@*cityu.edu.hk*

Abstract— **To overcome the inherent lossy property of wireless links and increase network throughput, many multi-path routing protocols have been proposed to improve the reliability and latency of packet delivery in wireless networks. Multi-path routing protocols, however, do not take advantage of existing coding opportunities to maximize network throughput. In this paper, we propose a novel coding-aware multi-path routing protocol (CAMP), which forwards packets over multiple paths dynamically based on path reliability and coding opportunity. CAMP employs a route discovery mechanism which returns to the source multiple paths along with ETX (Expected Transmission Count) of all links on each path. Using a novel forwarding mechanism, CAMP splits the traffic among multiple paths and actively creates instead of passively waiting for coding opportunity by switching its path to maximize the switching gain. Experimental results demonstrate that CAMP can achieve much higher throughput than comparable schemes for delivering packets in wireless networks.**

I. INTRODUCTION

Wireless networks are characterized by unreliable links, dynamic topology, limited channel bandwidth and constrained battery power of the nodes. Many routing protocols have been proposed to overcome these constraints and improve the reliability, throughput, and routing overhead in wireless networks. However, most of these protocols route data packets to the destination over a predetermined single path based on a certain metric, for example, the shortest path.

Much recent research has focused on multi-path routing to improve the reliability of data delivery, to balance the traffic load and power consumption among nodes, to reduce the end-to-end delay and frequency of route discovery, to cope with the frequent topological changes and consequently unreliable communication services and to improve the network security. However, multi-path routing protocols face the problem of exacerbating the network congestion when forwarding packets over multiple paths simultaneously. It becomes worse when multiple non-disjoint paths are selected for packet transmission and the intersection node becomes the bottleneck.

A common drawback of existing single-path and multi-path routing protocols is that their paths are predetermined in the route discovery phase and they do not actively seek out coding opportunity even though network coding has been demonstrated to play an important role in improving network throughput.

In this paper, we propose a novel coding-aware multi-path routing protocol (CAMP), which dynamically switches its paths based on path reliability and coding opportunity. A new concept of path switching gain is also proposed to quantitatively decide which path should be switched to thus utilize the coding opportunity judiciously.

CAMP consists of two major phases: route discovery and packet forwarding. The route discovery mechanism employed in CAMP returns to the source multiple good paths along with ETX [1] of all links on each path. The intermediate nodes of each path cache other route candidates for further switching decision. Using a novel forward packet forwarding mechanism, CAMP dynamically switches the default path to maximize network coding and path switching gain. Our experimental results demonstrate that CAMP can achieve much higher throughput than traditional best-path routing, multi-path routing, and single-path based network coding schemes for delivering packets in wireless networks.

The rest of the paper is organized as follows. Section II briefly discusses related works in multi-path routing and network coding. In Section III we describe our Coding-Aware Multi-path routing Protocol (CAMP) in detail. Section IV presents our performance studies. Finally, we conclude our work in Section V and present the future works.

II. RELATED WORK

There have been several proposals for designing new routing metrics, improving the reliability, network throughput and reducing the end-to-end delay of packet delivery in wireless networks. In this section we discuss those that leverage either of multi-path routing and network coding, or both.

A. Routing Metric

The Expected Transmission Count metric (ETX) was first proposed in [1] and implemented for DSR and DSDV routing protocols. ETX finds high-throughput paths on multi-hop wireless networks. It minimizes the expected total number

978-1-4244-3367-4/08/$25.00 ©2008 IEEE

of packet transmissions (including retransmissions) required to successfully deliver a packet to the ultimate destination. The ETX metric incorporates the effects of link loss ratios, asymmetry in the loss ratio between the two directions of each link, and interference among the successive links of a path.

Incorporated into the MR-LQSR protocol (Multi-Radio Link Quality Source Routing), the Expected Transmission Time (ETT) and Weighted Cumulative ETT (WCETT) [2] are designed especially for routing in multi-radio, multi-hop scenarios. ETT, the link metric, enhances ETX by taking both the loss rate and the bandwidth of a link into consideration while in WCETT, the path metric allows the users to trade off channel diversity and path length by changing the control parameter.

As CAMP focuses on improving network throughput in the single-radio, multi-hop scenarios, it uses ETX in its route discovery mechanism for the routing measurement.

B. Multi-path Routing

Meshed Multi-path Routing (M-MPR) [3] is a meshed multi-path routing protocol, which firstly constructs a path mesh between a source and a destination, and then in transmission each node selectively forwards packets to one of its downstream nodes in the mesh. The routing decision is taken dynamically in a hop-by-hop manner, according to the conditions of downstream nodes.

In [4], the authors analyze the performance of disjoint multi-path and meshed multi-path approaches, and argue that using disjoint paths limits route reliability in mobile ad hoc networks, while using meshed multiple paths could achieve better connectivity and performance.

Braided multi-path [5] is an approach proposed to improve the resilience of wireless sensor networks to node failures. In this approach, the braided paths are constructed using a path discovery mechanism called "localized path reinforcement". After sessions of localized path reinforcement, a braided path mesh is constructed around a primary path. Making path reinforcement localized is energy-efficient while discovering multiple alternative paths for failure recovery.

AOMDV, the Ad hoc On-demand Multi-path Distance Vector protocol [6], is a protocol based on AODV [7]. AOMDV finds multiple loop-free link-disjoint paths to achieve fast and efficient recovery from route failures.

Multi-path Dynamic Source Routing (MDSR) [8] extended DSR [9] to compute multiple disjoint paths for overhead reduction in mobile networks. Split Multi-path Routing (SMR) [10] is similar to MDSR except it uses a modified flooding mechanism and the data traffic is split among the multiple paths to avoid congestion.

Opportunistic Multi-path Scheduling (OMS) [11] opportunistically splits traffic over multiple paths based on the measured path conditions, favoring those paths with low-latency and high-throughput. Many other multi-path routing schemes are summarized in [12].

CAMP is different from these protocols in path selection, because CAMP selects paths which could result in high reliability and failure resilience in addition to maximizing network coding opportunity during packet forwarding, regardless of the path disjointness.

C. Network Coding

Network coding is an emerging and powerful technique which could be used to improve resilience, reduce end-to-end delay, and improve network throughput of wireless networks.

COPE [13] is a coding architecture which exploits coding opportunity in single path routing in wireless networks. The design and implementation of COPE have demonstrated that network coding techniques can be applied to real wireless networks. However, COPE is mainly designed for single path routing and it adopts a passive approach, essentially just waiting for coding opportunity.

Growth Codes [14] makes wireless sensor networks resilient to data loss by compactly replicating data packets and disseminating them among neighbors. The Growth Codes method assumes that a node is unaware of the network topology, and a node does not have any path pre-determined for packet transmissions. The same coded packet may travel multiple paths, which however are not pre-determined.

CAMP differs from COPE in an important aspect. It could dynamically change routes to create coding opportunity while COPE only can wait for the opportunity on a prefixed route. CAMP is also different from Growth Codes since packets do not grow under CAMP.

D. Multi-path Routing with Network Coding

To achieve better throughput as well as resilience over failures, many recent approaches combine multi-path routing with network coding.

In [15], [16], a multi-path scheme is proposed based on diversity coding [17]. This scheme adds some overhead to the original packet when encoding, and then splits the coded packet into several equal-size non-overlapping blocks, which are distributed to multiple node-disjoint paths. As long as a destination can collect enough data blocks such that the total size of those blocks is no less than the original packet, it can reconstruct the original packet.

To optimize the probability of successful data delivery in Delay Tolerant Networks (DTNs), an approach which combines erasure coding with multi-path routing is proposed in [18]. This approach specifically considers the modeling path failure and volume constraints in DTNs, and solves the optimal allocation problem of erasure code blocks over multiple paths. Another similar approach is presented in [19], which emphasizes data delivery latency in DTNs.

While the above approaches split coded packets among multiple paths, CAMP acts in a way similar to that of COPE - it simply XORs packets and broadcasts the coded packets out.

III. THE CAMP APPROACH

In this section, we present details of our CAMP approach. CAMP consists of two phases: route[1] discovery and coding-

---

[1] In the following of this paper, we use "path" and "route" interchangeably.

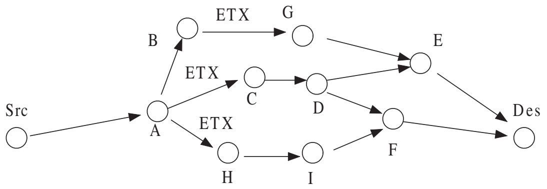


Fig. 1. An example of EDSR.

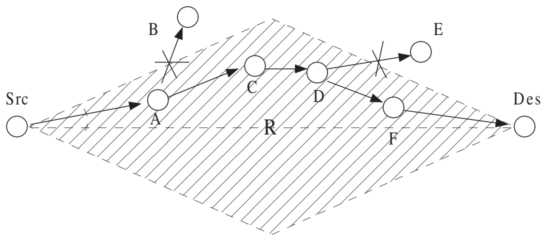


Fig. 2. An example of EDSR-OPT.

aware packet forwarding. It is clearly advantageous if the coding opportunity could be somehow predicted at the route discovery stage, and thus path selection could favor those paths which create rich coding opportunity. However, this is difficult to achieve, because coding opportunity is dependent of traffic flows, and at the initial stage there is no way to predict traffic pattern in most wireless networks. CAMP judiciously solves this problem by dynamically selecting the path in a hop-by-hop manner during the packet forwarding phase to maximize the coding opportunity and thus improve network throughput.

A. Route Discovery

In this section, we describe our route discovery approaches in detail. We first present our assumptions in Section III-A.1 and then we give the description of the basic approach named EDSR in Section III-A.2, which is an extended variant of DSR. Finally, EDSR-OPT, an optimization of EDSR, is presented in Section III-A.3 to restrict the route request packet's flooding region thus improve the latency of the route discovery process.

1) Assumptions: We focus our work on the scenario of wireless mesh networks where the wireless nodes are mostly static. We assume that in the wireless network, each node periodically broadcasts probing packets within the neighborhood. This usage of probing packets is inspired by a similar mechanism of measuring link ETX [1]. By sending periodical probing packets, a node can detect its neighbors, and compute the link ETX between the node and any of the neighbors. For simplicity, we also assume that the link in the wireless network is bi-directional; thus route reply is sent on the exact route obtained by reversing the route discovered.

2) Basic Approach: Our basic route discovery mechanism, EDSR, is derived from DSR [9]. It differs from DSR in adding link information into the route request before re-broadcasting it out. In EDSR, after receiving the route request from a neighbor, a node attaches the ETX of the previous link with the route request, and then rebroadcasts it out. If a node gets the same route request twice, it discards the second one.

Each time a destination receives a route request, it replies to the source with a route reply which includes the discovered route and the ETX of each link on that route. A route request from a source usually could traverse through multiple routes to the destination, and thus after the route discovery phase a source could get multiple routes to the destination.

Shall a destination reply to every route request it receives? Our answer is ”no”. Ideally we want the destination to send only the good routes back to the source. To achieve this, the destination should reply only to a route request arriving within a time bound τ, because late route requests usually indicate that they have traversed links with very low delivery ratio, or have gone through too many hops. For this purpose, the source includes a timer τ in the route request, and any node should not forward or reply to a route request when τ expires.

Like DSR, EDSR caches new routes with ETX of each link it learns by any means. This mechanism is particularly important for EDSR not only because it can speed up route discovery and reduce propagation of route requests, but also it can provide the candidates for path switching in the packet forwarding phase to maximize the utilization of existing coding opportunity and thus improve network throughput.

3) Optimization: To search for possible routes to the destination, EDSR floods the network with discovery packets which will increase the latency of the discovery phase and network overhead especially when the network size is large. Although DSR will gradually increase the route request's TTL field to limit the extent of propagation, the lack of an explicit propagation region still results in larger latency in the route discovery phase.

In this subsection, we present the EDSR-OPT mechanism which limits route request flooding to a restricted diamond region between the source and destination. Different from EDSR, here we strengthen our assumption that each node is aware of the geographic location of itself and those of the source and the destination. Since the nodes in a wireless mesh network are mostly static, the nodes' location information can be imparted during their initial deployment phase via standard trilateration approach using other GPS-capable nodes.

The intuition of EDSR-OPT is demonstrated in Fig. 2. With a source specified parameter θ, a diamond region R is defined between the source and the destination to restrict the route request propagation. When a node receives the route request, it will first check whether it is inside R before re-broadcasting the route request to its neighbors. For example, in Fig. 2, node B and E will discard the route request from A and D respectively while C and F will identify themselves as proper candidates. With the introduce of the restricted region R, the path pruning is not only happened at the destination, but also in-network. More importantly, as the size of the restricted region is independent of the network size, the route discovery becomes scalable.

Assuming the geographical positions of the node itself, the source and the destination are (x, y), (x_S, y_S) and (x_D, y_D)

respectively, the condition for the node to reside inside the restricted region R with a specified θ is presented below.

$$\begin{cases} x > (y - y_S) \times \tan(\arctan(\dfrac{y_D - y_S}{x_D - x_S}) + \theta) + x_S; \\ x < (y - y_D) \times \tan(\arctan(\dfrac{y_D - y_S}{x_D - x_S}) + \theta) + x_D; \\ y > (x - x_S) \times \tan(\arctan(\dfrac{y_D - y_S}{x_D - x_S}) - \theta) + y_S; \\ y < (x - x_D) \times \tan(\arctan(\dfrac{y_D - y_S}{x_D - x_S}) - \theta) + y_D; \end{cases} \tag{1}$$

B. Coding-aware Packet Forwarding

In this section, we describe our coding-aware packet forwarding mechanism in detail. In the packet forwarding phase, the source's traffic is split among several good paths which have been found in the route discovery phase in Section III-A. More importantly, CAMP dynamically switches its path during the packet forwarding to create coding opportunity and consequently increase network throughput.

1) Intuition: Traditional single path and multi-path routing protocols have determined routes after the route discovery phase and packets will be transmitted along fixed paths from the source to the destination. Although the selected path(s) are the best path(s) according to a certain metric, e.g., shortest path, they ignore the possible rich coding opportunities existing in other paths and make the overall performance suboptimal.

To overcome this drawback, CAMP uses a coding-aware packet forwarding approach to leverage the coding opportunity. Instead of waiting for coding opportunity, CAMP can change the forwarding path to *actively* create coding opportunity. Fig. 3 shows a scenario where node A wants to deliver packet a to D, and at the same time B wants to deliver packet b to C. Assume that the best path from A to D is A-E-G-D, and the best path from B to C is B-E-F-C, and further assume G can not overhear B, and F can not overhear A. In this scenario, there is no way for COPE [13] to discover any coding opportunity, since packet exchanging does not exist, COPE's opportunistic listening does not help. However, as long as paths E-A-C and E-B-D are also cached in node E, under CAMP node E will XOR a and b, and broadcast $a \oplus b$ to A and B. Then A and B could decode $a \oplus b$ to get b and a respectively. A then forwards b to C, and B can forward a to D. Note that this coding opportunity cannot be exploited by any single path routing scheme, because in single path routing node E is not aware of any alternative path to a destination. Traditional multi-path routing schemes which forward packets over multiple path simultaneously also cannot benefit from the coding opportunity here, since their paths are fixed and cannot change dynamically.

2) Traffic Splitting: After the route discovery phase, the source has got multiple path candidates to the destination with ETX of all links on each path. The source evaluates the ETX for each path and selects the top K of them. Here K is a user-specified parameter to define the granularity of the traffic splitting; it is always a small number, for example, 3. Assuming e_i is the ETX of link i on a path P, then the ETX for path P, ETX^P, can be evaluated as:

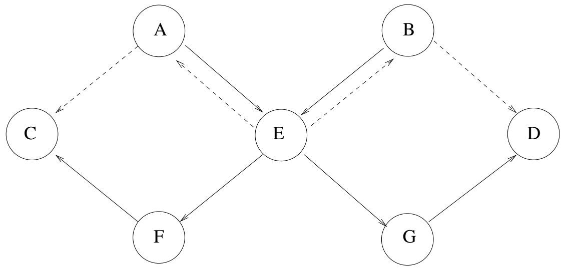
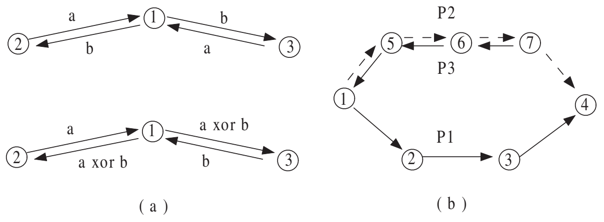


Fig. 3. Illustration shows that packet forwarding under CAMP could actively exploit coding opportunity. To take coding opportunity, node E drops the best paths.



(a) (b)

Fig. 4. Evaluation of the exchange gain and path switching gain

$$\text{ETX}^P = \sum e_i \quad (i \in P) \tag{2}$$

Assuming the path set $\{P_i\}_{i=1}^K$ is chosen by the source to forward a traffic flow T and their relative path ETX is $\{\text{ETX}_i^P\}_{i=1}^K$. The portion of T which will be split to path P_i is:

$$T_i = \frac{\text{ETX}_i^P}{\sum_{j=1}^K \text{ETX}_j^P} \cdot T \tag{3}$$

The reasons why we do not split the traffic on all the available paths to increase the latency of the packet transmission are: first, the paths with low ETX values are not reliable and will cause a large amount of packet retransmissions. Second, the attempt to transmit packets simultaneously on as many as possible paths will cause severe network congestion which hurts overall throughput. Last, the TCP packet reordering problem will make the situation worse.

3) Theoretical Analysis of Path Switching Gain: In coding-aware path switching, switching the path immediately to an alternative path which has superior coding opportunity is not always advantageous. There is a tradeoff between the achieved coding gain and the loss from changing the best path to the suboptimal one. Only when the coding gain can overcome loss in the path ETX should the path switching be conducted.

We first define the concept of exchange gain and deduce its theoretical estimation through a simple example. Then based on the exchange gain, we define and evaluate the path

TABLE I

SYMBOLS AND DEFINITIONS

P_i	path i
$\hat{s}^i$	predecessor of node s on path P_i
$\check{s}^i$	successor of node s on path P_i
$m_{i,j}$	number of codable nodes on P_i and P_j
$r_{i,j}$	two-way successful delivery probability between i and j
$s_{i,j}$	two-way failed delivery probability between i and j
$g_{i,j,k}$	exchange gain at j when packets exchange between i and k
$g^P_{i,j,k}$	sum of exchange gain along path P
$G_{i,j,k}$	gain of switching path from P_i to P_j, for packet exchanging with P_k

switching gain which quantitatively decides whether a path switching would be carried out.

Definition 1: Exchange gain $g_{i,j,k}$ is the number of reduced transmissions from a successful exchange of packets between node i and node k via the intermediate node j using network coding.

First consider the simple scenario in Fig. 4(a) and assume that the one-way delivery probability from node i to node j is given by $r_{i\to j}$. We use $r_{i,j}$ to denote the probability of a successful two-way delivery including the acknowledgment, i.e., $r_{i,j} = r_{i\to j} r_{j\to i}$. According to the ETX metric [1], the expected number of transmissions for a successful exchange without coding is $\frac{2}{r_{1,2}} + \frac{2}{r_{1,3}}$. Even with coding, since only node 1 performs coding, we still need $\frac{1}{r_{1,2}} + \frac{1}{r_{1,3}}$ transmissions for nodes 2 and 3 to deliver both packets to node 1. To complete the exchange, node 1 needs to successfully deliver the coded packet to both nodes 2 and 3, which requires the coded transmissions:

$$\begin{aligned} T \;=\; & \sum_{k=1}^{\infty}\Big(s_{1,2}^{k-1} r_{1,2} \sum_{i=1}^{k} s_{1,3}^{i-1} r_{1,3} \\ & + s_{1,3}^{k-1} r_{1,3} \sum_{i=1}^{k} s_{1,2}^{i-1} r_{1,2} - s_{1,2}^{k-1} s_{1,3}^{k-1} r_{1,2} r_{1,3}\Big)k \end{aligned}$$

where k is the number of retries conducted before the transmission is successful and $s_{i,j} = 1 - r_{i,j}$, which is just $\frac{1}{r_{1,2}} + \frac{1}{r_{1,3}} - \frac{1}{r_{1,2}+r_{1,3}-r_{1,2}r_{1,3}}$. Therefore, the expected number of transmissions for a successful exchange with coding, is $\frac{2}{r_{1,2}} + \frac{2}{r_{1,3}} - \frac{1}{r_{1,2}+r_{1,3}-r_{1,2}r_{1,3}}$. Effectively, $g_{2,1,3}$, the exchange gain at node 1,through the successful exchange of packets between node 2 and node 3 is $\frac{1}{r_{1,2}+r_{1,3}-r_{1,2}r_{1,3}}$. Formally, assume node i and node k exchange packets successfully through node j, the exchange gain at node j is defined as:

$$g_{i,j,k} = \frac{1}{r_{i,j} + r_{j,k} - r_{i,j} r_{j,k}} \tag{4}$$

Using the concept of exchange gain, we define path coding gain and path switching gain as follows:

Definition 2: Path coding gain $g^P_{i,j,k}$ is the number of transmissions reduced by switching the path from P_i to P_j to embrace the coding opportunity with P_k.

Definition 3: When we switch from P_i to P_j for exchanging packets with P_k, the *Path switching gain* $G_{i,j,k}$ is the difference between the path coding gain $g^P_{i,j,k}$ and the path ETX loss, $\mathrm{ETX}^P_j - \mathrm{ETX}^P_i$.

In Fig. 4(b), node 1 attempts to switch from P_1 to P_2 thus it can exchange the packets with path P_3 and enhance coding opportunity. This switching is only successful when the path coding gain achieved from node 5 and node 6 can overcome the increased path ETX caused by the path switching, that is $G_{1,2,3} > 0$.

$$g^P_{1,2,3} = g_{1,5,6} + g_{5,6,7} \tag{5}$$

$$G_{1,2,3} = \mathrm{ETX}^P_1 - \mathrm{ETX}^P_2 + g^P_{1,2,3} \tag{6}$$

Formally, assume path P_i is to be switched to path P_j to exchange packets with P_k. $\{N^C_s\}^{m_{j,k}}_{s=1}$ denotes the set of nodes shared by P_j and P_k which can achieve exchange gains. The path coding gain, $g^P_{i,j,k}$ and path switching gain, $G_{i,j,k}$, are evaluated as below:

Alg 1 Path Switching Algorithm

```
Input: A set of paths $P = \{P_i\}_{i=1}^{m}$ $(m \geq 1)$; the default path $P_k$ and the set of candidates $\{\tilde{P}_j^k\}_{j=1}^{n}$.
Output: $l$, the index of the alternative path $\tilde{P}_l^k$ which maximizes the path switching gain.
 1: {lines 2-3: initialization}
 2: $G_{MAX} \leftarrow 0$
 3: $l \leftarrow 0$
 4: {lines 5-19: calculate the maximum gain}
 5: for candidate $j = 1$ to $n$ do
 6:     $G \leftarrow 0$
 7:     for path $i = 1$ to $m$ do
 8:         $G_{old} \leftarrow G$ {backup previous value}
 9:         if $i \neq k$ and $P_i \cap P_k \neq \emptyset$ then
10:             {calculate the gain}
11:             $G \leftarrow$ CalcGain$(P_k, \tilde{P}_j^k, P_i)$
12:             if $G \leq G_{old}$ then
13:                 $G \leftarrow G_{old}$
14:         else
15:             continue
16:     if $G_{MAX} < G$ then
17:         $G_{MAX} \leftarrow G$ {update with higher gain}
18:         $l \leftarrow j$
19: return $l$ {return the index of the selected path}
```

Alg 2 CalcGain(i, j, k)

```
Input: Path $P_i$, $P_j$ and $P_k$.
Output: $G_{i,j,k}$, the path switching gain by switching $P_i$ to $P_j$ to exchange packets with $P_k$.
 1: {account for the loss when choosing longer path}
 2: $G_{i,j,k} \leftarrow \mathrm{ETX}_i^P - \mathrm{ETX}_j^P$
 3: {lines 4-7: calculate the path switching gain}
 4: for all node $v \in P_j$ do
 5:     {add together the exchange gain on each node}
 6:     $G_{i,j,k} \leftarrow G_{i,j,k} + g_{\hat{v}^j, v, \check{v}^j}$
 7: return $G_{i,j,k}$
```

$$g^P_{i,j,k} = \sum_{s=1}^{m_{j,k}} g_{\hat{s}^j,s,\breve{s}^j} \tag{7}$$

$$G_{i,j,k} = \mathrm{ETX}^P_i - \mathrm{ETX}^P_j + g^P_{i,j,k} \tag{8}$$

where $\hat{s}^j$ and $\breve{s}^j$ are the predecessor and successor nodes of node N^C_s on path P_j respectively. The path switching between P_i and P_j will only be conducted when $G_{i,j,k} > 0$.

4) Path Selection Algorithm: Based on the approach for path switching gain in Section III-B.3, at intermediate nodes, CAMP chooses an alternative path for each path P_k to enhance coding opportunity and maximize switching gain. The path selection algorithm examines each candidate and maximizes path switching gain by exchanging it with all the possible paths, except the ones disjoint with itself. The candidate with the maximum gain will be selected to forward the packets. Assuming there are m existing paths and n candidates, the complexity of the algorithm is $O(m \cdot n)$. The details of the path selection algorithm are presented in Alg. 1.

5) Discussion: In the algorithm presented in Section III-B.4, we evaluate the maximum path switching gain in a pair-wise manner and do not consider the combined effect from multiple paths on path switching gain. If chosen this more complex metric will greatly increase the calculation overhead and transmission delay in the intermediate node. As the future work, we will explore a simple method to take the combined path switching gain into consideration and do the path switching more accurately.

IV. PERFORMANCE EVALUATION

This section presents the performance evaluation results of CAMP. We use the NS2 [20] simulation platform for our evaluation. We have conducted two experiments to compare CAMP with peer schemes in terms of throughput. In one experiment we use a topology with 6 nodes. In the other experiment we use a 4x4 grid topology, which has 16 nodes in total. In all our experiments, we set the packet size to 1500 bytes. We turn off RTS/CTS exchange, since the RTS/CTS exchange just incurs routing overhead and decreases network throughput [21], [22].

A. Evaluation with a 6-node Topology

On the 6-node topology, where every link is assumed perfect with a delivery probability of 1.0, we have two simulation scenarios. In both scenarios, we set the transmitting rate to 1Mbit/s, and set the transmitting interval to 0.2s, which corresponds to a very light traffic load. We compare CAMP with three other schemes: Multi-path routing, COPE, and Traditional routing. The simulation time in each case is 50 seconds.

In the first simulation scenario, there is only one data flow in the network. This flow traverses 3 hops from the source to the destination. Figure 5 shows the per flow throughput of applying different schemes in this simulation scenario. We have several observations: (i) COPE and traditional best path routing have the same performance. This is because COPE does not have any coding opportunity given that there is only a single flow in the network. (ii) CAMP and Multi-path routing have equally good performance. This is because CAMP does not have coding gain with a single flow, and thus it just forwards packets over multiple paths in the same way with Multi-path routing without coding any packets. (iii) CAMP and Multi-path routing outperforms COPE and traditional routing. The reason is that the first two approaches, unlike the later two which use only the best path, forward packets over multiple paths.

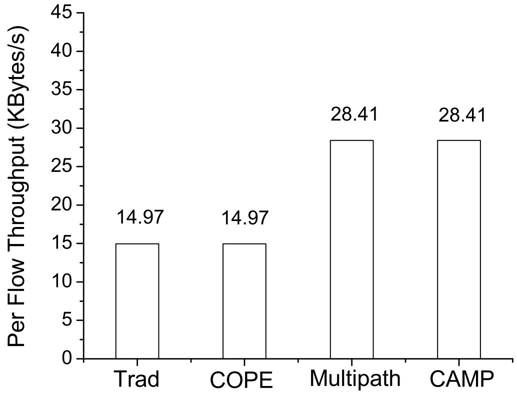


Fig. 5. No coding opportunity exist when only single traffic flow exists. CAMP and Multi-path outperform COPE and traditional routing.

In the second simulation scenario, we include an additional data flow to the network; it goes from the destination of the first flow to the source of that flow. Now there are two traffic sources in the network and they send packets to each other. In this scenario, coding opportunities exist due to the many packet exchange. The per-flow throughput of the four schemes studied in the paper is shown in Figure 6. The key observation is: CAMP has the highest per flow throughput, while Multi-path routing is better than COPE, which has higher throughput than traditional routing. CAMP performs best because it takes advantage of both multi-path rouging and network coding. Another observation is that COPE does not have significant throughput improvement over traditional routing in this light traffic load scenario, which means that the packet queue at each node must be short, which reduces coding opportunity.

B. Evaluation with a 4x4 Grid Topology

In this experiment, we have a 4x4 grid topology with total 16 nodes. The topology is shown in Figure 8. Links with dashed lines in the topology all have 10% loss ratio, while links with solid lines are lossless.

We select two flows transmitting in the network. One flow goes from the top left of the grid down to the furthest corner at the right, while the other flow goes in the reverse direction. The two flows traverse different number of hops, and they have only two-hop overlap along their best routes.

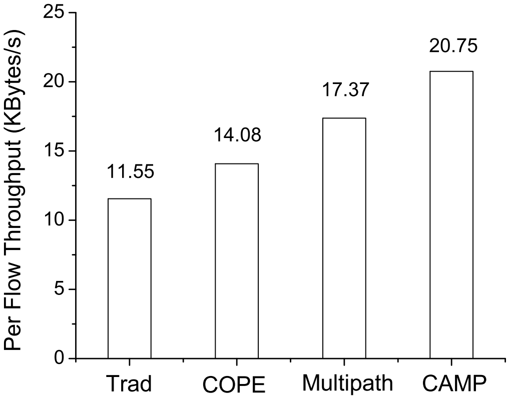


Fig. 6. With two exchanging traffic flows, CAMP has highest throughput. COPE does not have significant gain because of light traffic load.

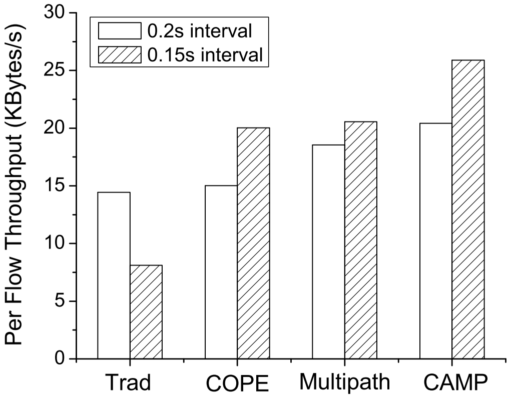


Fig. 7. With different traffic loads, CAMP outperforms peers schemes in terms of throughput. Traditional routing suffers from congestion when traffic load becomes heavy.

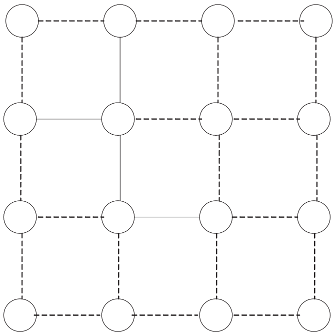


Fig. 8. 4x4 grid topology used in simulation. Dashed links are with 10% loss ratio, solid links are perfect with loss ratio of 0.

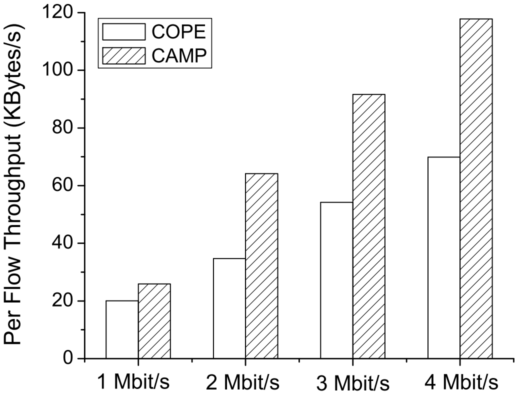


Fig. 9. CAMP vs. COPE: with transmitting rates varying from 1Mbit/s to 4Mbit/s, on average CAMP achieves 70% throughput improvement over COPE.

Figure 7 shows the per-flow throughput of CAMP and other three schemes, with two different traffic loads. In both cases, CAMP outperforms the other schemes in terms of per flow throughput. When the transmitting interval is 0.15s, the traffic load could be very high at the overlapped links of the best routes of the two transmitting flows. In this case, traditional routing suffers heavily from collision and congestion and performs worse than in the light traffic setting (with 0.2s interval). COPE performs better because it employs network coding to significantly reduce the traffic load. CAMP performs best because it could take more advantage of network coding by actively creating coding opportunities. CAMP's multi-path forwarding is another important reason for better throughput.

For a more thorough comparison between CAMP and COPE, we vary the transmitting rates in our simulation. The results are shown in Figure 9. For all the 4 transmitting rates from 1Mbit/s to 4Mbit/s, we observe significant per flow throughput improvement of CAMP over COPE. On the average CAMP achieves about 70% throughput improvement over COPE.

V. CONCLUSION

In this paper we present CAMP, a novel coding-aware multi-path routing protocol which forwards packets over multiple paths dynamically based on path reliability and coding opportunity. CAMP actively creates coding opportunity by switching its path to the one which brings the maximum path switching gain and thus improving network throughput. CAMP is demonstrated through simulation that it can achieve high throughput for delivering packets in wireless networks.

Future work on CAMP includes evaluating the impact of interference on CAMP, taking the packet re-ordering issue into consideration, examining CAMP with larger and more complex network topologies, and implementing CAMP in a test bed. We are also exploring various timing control issues with network coding, particularly how to maximize coding opportunities while still maintaining the end-to-end delay in multi-hop wireless networks within a certain constraint.

ACKNOWLEDGMENT

Guihai Chen's work is partly supported by China NSF grants (60573131, 60673154, 60721002), and Jiangsu High-Tech Research Project of China (BG2007039).

REFERENCES

[1] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," in *Proc. ACM MobiCom*, 2003.

[2] Richard Draves, Jitendra Padhye, and Brian Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, 2004.

[3] S. De, C. Qiao, and H. Wu, "Meshed multipath routing with selective forwarding: An efficient strategy in wireless sensor networks," *Elsevier Computer Communications Journal*, vol. 26, no. 4, 2003.

[4] J. J. Garcia-Luna-Aceves and Marc Mosko, "Multipath routing in wireless mesh networks," in *First IEEE Workshop on Wireless Mesh Networks (WiMesh 2005)*, 2005.

[5] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *Mobile Computing and Communications Review*, vol. 4, no. 5, Oct. 2001.

[6] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *IEEE International Conference on Network Protocols (ICNP)*, 2001.

[7] C. Perkins and E. Royer, "Ad Hoc On-Demand Distance Vector Routing," in *The 2nd IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999.

[8] A. Nasipuri, R. Castaneda, and S. R. Das, "Performance of multipath routing for on-demand protocols in ad hoc networks," *ACM/Kluwer Mobile Networks and Applications (MONET) Journal*, vol. 6, no. 4, 2001.

[9] D. Johnson, D. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," in *Adhoc Networking*, Charles E. Perkins, Ed. 2001, pp. 139–172, Addison-Wesley.

[10] S. J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *IEEE ICC*, 2001, pp. 3201–3205.

[11] C. Cetinkaya and E. Knightly, "Opportunistic Traffic Scheduling Over Multiple Network Paths," in *Proc. IEEE Infocom*, 2004.

[12] W. Lou, W. Liu, and Y. Zhang, "Performance optimization using multipath routing in mobile ad hoc and wireless sensor networks," *Combinatorial Optimization in Communication Networks*, 2005.

[13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," in *Proc. ACM Sigcomm*, 2006.

[14] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein, "Growth codes: Maximizing sensor network data persistence," in *Proc. ACM Sigcomm*, 2006.

[15] A. Tsirigos and Z. J. Hass, "Analysis of multipath routing, part 1: The effect on the packet delivery ratio," *Transactions on Wireless Communications*, vol. 3, no. 1, 2004.

[16] A. Tsirigos and Z. J. Hass, "Analysis of multipath routing, part 2: Mitigation of the effect of frequently changing network topologies," *Transactions on Wireless Communications*, vol. 3, no. 2, 2004.

[17] E. Ayanoglu, C.-L. I, R. D. Gitlin, and J. E. Mazo, "Diversity coding for transparent self-healing and fault-tolerant communication networks," *IEEE Trans. Commun.*, vol. 41, pp. 1677–1686, Nov. 1993.

[18] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," in *Proc. ACM Sigcomm*, 2005.

[19] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *SIGCOMM Workshop on delay tolerant networking and related networks*, 2005.

[20] NS2, "http://www.isi.edu/nsnam/ns/," .

[21] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and Evaluation of an Unplanned 802.11b Mesh Network," in *Proc. ACM MobiCom*, 2005.

[22] K. Xu, M. Gerla, and S. Bae, "Effectiveness of rts/cts handshake in ieee 802.11 based ad hoc networks," *Ad Hoc Network Journal*, vol. 1, no. 1, July 2003.