Wireless Netw (2016) 22:839–854
DOI 10.1007/s11276-015-1003-3

EPTR: expected path throughput based routing protocol for wireless mesh network

Xiaoheng Deng[1] · Lifang He[1] · Qiang Liu[1] · Xu Li[1] · Lin Cai[2] · Zhigang Chen[3]

Published online: 1 July 2015

Abstract For effective routing in wireless mesh networks, we proposed a routing metric, expected path throughput (EPT), and a routing protocol, expected path throughput routing protocol (EPTR), to maximize the network throughput. The routing metric EPT is based on the estimated available bandwidth of the routing path, considering the link quality, the inter- and intra-flow interference and the path length. To calculate the EPT of a routing path, we first calculate the expected bandwidth of the link and the clique, and then consider the decay caused by the path length. Based on EPT, a distributed routing protocol EPTR is proposed, aiming to balance the network load and maximize the network throughput. Extensive simulations are conducted to evaluate the performance of the proposed solution. The results show that the proposed EPTR can effectively balance the network load, achieve high network throughput, and out-perform the existing routing protocols with the routing metrics previously proposed for wireless mesh networks.

Keywords Wireless mesh network · Signal-to-noise ratio · Expected path throughput · Cross layer design · Routing protocol

1 Introduction

Wireless systems continue to attract research and development effort [3, 4]. Especially, multi-hop secondary networks such as mobile ad hoc networks (MANETs, so-called opportunistic networks) [5–7], wireless sensor networks (WSNs) have gained attention as promising designs to leverage the full potential of Cognitive Radio Networks (CRNs) [8–10]. Besides, the new era of Internet of Things [11–14] is driving the development for Vehicular Ad hoc Networks and Vehicular Sensor Networks [15]. While wireless mesh networks (WMNs) receive extensive attention thanks to its adaptability and expandability, and it can provide high data-rate, low cost and reliable wireless transmission methods in various types of applications, such as disaster relief, military surveillance and smart grid [16]. WMNs often rely on the IEEE 802.11 standards to provide wireless broadband services. As shown in Fig. 1, a typical WMN includes mesh clients, mesh routers and mesh gateways [17]. Mesh clients can roam in the WMN and send and receive data through mesh routers. Mesh routers forward packets to/from mesh gateways which are connected to the Internet. The mesh routers form a mesh of

Preliminary results of this work have been presented in IEEE WCNC'13 [1] and CNCC'13 [2].

✉ Xiaoheng Deng
dxh@csu.edu.cn

Lifang He
helifang@csu.edu.cn

Qiang Liu
jasnykra@csu.edu.cn

Xu Li
lishlxg@csu.edu.cn

Lin Cai
cai@ece.uvic.ca

Zhigang Chen
czg@csu.edu.cn

[1] School of Information Science and Engineering, Central South University, Changsha 410083, China

[2] Department of Electrical and Computer Engineering, University of Victoria, Victoria V8P 5C2, Canada

[3] School of Software, Central South University, Changsha, Hunan, China

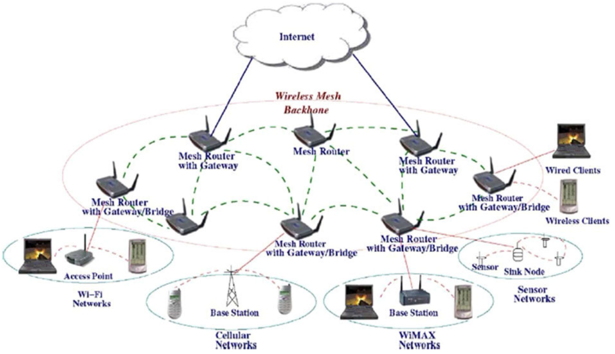


Fig. 1 Infrastructure-based WMNs

self-configurable and self-healable links to provide Internet access for the clients.

Traditional distributed routing mechanisms often use hop count or end-to-end delay as the routing metrics, as they are isotonic and easy to implement. In wireless mesh networks, however, they may fail to efficiently utilize the wireless resources because the network throughput is also largely dependent on the wireless link rate and the load nearby. And wireless link rate is determined by the signal-to-noise ratio (SNR) in the physical layer, affected by path-loss, fading, shadowing and interference. Further, as wireless medium is shared in nature, the available bandwidth of a link is determined by not only the traffic load of the tagged transmitting node, but also that of the neighboring nodes who compete for accessing the channel. So it is do need to design a cross-layer routing protocol for multi-hop wireless mesh networks which uses the lower layer information related to link quality and load to assist the network layer routing algorithm to select the optimal path.

In the past, extensive routing protocols have been proposed for mobile MANETs [18–24] and WSNs [25–31]. Even numerous efforts have been paid to the survey on routing protocols in MANETs and WSNs based Delay Tolerant Networks (DTNs) [32–34], Vehicular Networks [35–37] and Body Area Networks (BANs) [38]. But WMNs are different from both MANETs and WSNs as the relay nodes in the backbone of WMNs are static mesh routers without the stringent power limitation. In the literature, considering the lower layers' properties of WMNs, new routing metrics have been developed for WMNs, most notably the Expected Transmission Count (ETX) [39], Expected Transmission Time (ETT) [40], Weighted Cumulative ETT (WCETT) [41], Interfering Neighbors Count (INX) [42], AVAIL [43], resource aware routing for mesh (RARE) [44], channel quality and load sensitive [2]. Although these solutions can achieve substantial performance gains in various scenarios, the dynamic of traffic in WMNs are not fully considered which motivates this work.

In this paper, we design a cross-layer mechanism by combing the information from the PHY, MAC and network layers in order to maximize the network throughput. By considering the PHY layer adaptive modulation and coding, we adopt the link rate adaptation to handle the varying channel quality. According to the MAC layer status, the estimation of the network load and inter-flow interference can be calculated. Then, in the network layer, all information is synthesized to deal with issues such as the load balance and intra-flow interference in a distributed way.

Based on the above idea, our main contributions focus on a defined routing metric, the expected path throughput (EPT), and a distributed routing protocol, named expected path throughput routing protocol (EPTR). The metric, EPT, considers the varying link quality, the inter-interference and intra-interference and the path length, to estimate the available path bandwidth. Based on EPT, the EPTR protocol is proposed to balance the network load and maximize the network throughput. We have implemented the proposed EPTR in NS-2.31, and evaluated the performance in various scenarios. The results show that the proposed metric and protocol can achieve load balance very well and maximize the network throughput, and outperform ETXETT and CAB metrics which proposed in [45], and the existing AODV [19], DSDV [18], and DSR [21].

The rest of the paper is organized as following. Section 2 discusses the related work. In Sect. 3, we give the motivation and design of the proposed routing metric, EPT. The distributed on-demand source routing protocol based on EPT is proposed in Sect. 4. In Sect. 5, simulation settings and results are given, followed by the conclusions and future research issues in Sect. 6.

2 Related work

Despiting efforts to improve the current IEEE 802.11 standard to fully optimize the physical layer and MAC layer , the performance of wireless mesh networks still depends on the routing mechanism. Many cross-layer routing metrics and solutions have been proposed to improve WMN performance, and here we highlight the most relevant ones.

Based on successful transferring probability, the Expected Transmission Count (ETX) [39] is a basic metric, which is defined as the expected number of MAC layer transmissions that is needed to successfully deliver a packet, but it doesn't consider the link idle probability and clique bandwidth, so it cannot fully utilize the wireless resource and adapt to the dynamic traffic in the WMNs. The Expected Transmission Time (ETT) [40] improves

ETX by considering the link capacity, but the inter-flow and intra-flow interference are ignored. Similar problems appear in the Weighted Cumulative Expected Transmission Time (WCETT) [41] which has extended ETT to deal with multi-radio and intra-flow problem. While the Interfering Neighbors Count (INX) [42] has combined the link interference, it is still difficult to maximize the utilization of the link resources under the node co-interfering situation. By taking the fraction of busy time and loss probability into consideration, Available bandwidth (AVAIL) [43] is proposed but the transmission rate is ignored.

As for estimation of the link bandwidth which plays a significant role in wireless multi-hop routing protocol, many algorithms have been studied such as [46] and [47] where the idle status of the link has been used to assist the estimation. The idle probability is calculates by analyzing collisions and backoff in the MAC layer in [46], and the weighted average method is studied in [47] aiming to predict the idle probability accurately.

Besides, the clique-based path bandwidth computation which is widely used in the analysis of interference [43, 45, 48]. Authors in [43] developed a formula to approximately compute the available bandwidth of the path. Based on the clique mechanism, Leo Moser et al. proposed a isotonic metric and routing protocol which focuses on hop-by-hop QoS routing to guarantee bandwidth for flows [45], but their work is source routing which only uses reactive scheme. A QoS routing mechanism is proposed in [48], which computes in a distributed manner by recording the top k shortest path to find the close-to-optimal paths in order to match the user's bandwidth requirement. To improve the performance of Multi-Radio Multi-Channel (MRMC)WMNs, Duarte et al. [49] explored the possibility of exploiting Partially Overlapped Channels (POCs) by introducing a novel game theoretic distributed CA algorithm.

In this paper, based on cross-layer design, we aiming to obtain accurate available link capacity and find the path with highest bandwidth in a distributed manner. Firstly, we calculate the idle probability according to the network and MAC layer information in order to eliminate the affect from a certain flow. Then we proposed a metric called expected path throughput (EPT) which combines link quality and link load information together without modifying the MAC layer implementation and considers the intra- and inter-flow interference at the same time. Finally, Based on EPT, a distributed routing protocol named expected path throughput routing protocol (EPTR) is proposed which applies the clique-based path bandwidth computation to balance the network load and maximize the network throughput. And we apply the formula and reduce the routing overhead by designing the request packets to bring more required information.

3 EPT routing metric design

3.1 Link capacity

To calculate the EPT of a path, we first need to determine the link bandwidth, which is random due to channel dynamics. In order to utilize the dynamic wireless link efficiently, in the PHY layer, adaptive modulation and coding schemes have been proposed and widely deployed, so the link data rate is adjusted according to the channel quality, measured by SNR [50]. Specifically, given the SNR, the PHY layer will select the highest data rate transmission mode (which determines the modulation and coding schemes used) such that the bit-error-rate (BER) is below certain threshold. For instance, according to the IEEE 802.11b standard, different transmission modes can be selected which lead to the link data rate of 1, 2, 5.5 or 11 Mbps. The relationship between BER and SNR can be derived theoretically, or determined from the empirical measurement results. Given a BER threshold, the SNR range for different transmission mode can be determined.

For the path selection purposes, the expected SNR (or link capacity), rather than the instantaneous one, is concerned. Thus, we use an exponentially weighted moving average method to obtain the average of SNR to estimate the link quality as follows,

$$SNR = \alpha \times SNR + (1 - \alpha) \times SNR_{new}, \tag{1}$$

where SNR_{new} is the most recent measurement,[1] and we choose $\alpha = 0.75$ among several values from 0 to 1 according to the performance evaluation.

The average of SNR is then mapped to the expected link throughput according to a lookup table, e.g., using Table 1 for IEEE 802.11b, which is obtained from the empirical data [50] with the constraint of $BER \leq 10^{-5}$. Note that the above moving average of SNR is used to determine the expected link throughput for routing purpose only, which may deviate from the instantaneous link data rate. In the PHY layer, the transmission mode is adjusted according to the instantaneous channel quality, and different wireless vendors may use various proprietary technologies to select the mode, which is out of the scope of this paper. In Table 1, SNR below 1dB referring to the worst link status but not a broken one is mapped to the lowest rate, which is a low but non-zero value (0.01 in our simulation) for calculation which may lead to a low EPT for the route, on the other hand, every valid route can be found and the connectivity can be ensured.

[1] The SNR is measured from any received frames at the antenna of the receiver before decoding the spread signal [50].

Table 1 SNR and data rate mapping for IEEE802.11b

SNR threshold (dB)	Data rate (Mbps)	Transmission mode
>12	11	CCK11
8–12	5.5	CKK5.5
4–8	2	QPSK
1–4	1	BPSK
◁1	Lowest rate	

3.2 Expected link bandwidth

In addition to the estimated link capacity, to determine the available bottleneck bandwidth for a flow, the competition from other flows sharing the wireless channel needs to be considered. Although there exist extensive analytical work to model the throughput of single-hop wireless networks considering the contention medium access, such as Bianch's model [51] and Dhanasekaran's model [52], it is very difficult to apply these models to multi-hop WMNs because the factors of un-balanced load and the intra-interference among the nodes should be considered. Here, we let each node monitor the wireless activities to estimate the traffic load and the available bandwidth. Each node measures the percentage of the idle time of the wireless channel. The busy time and idle time are denoted by T_b and T_i respectively. The channel (around node A) idle probability is given by

$$P_{idle}(A) = \frac{T_i(A)}{T_b(A) + T_i(A)}. \tag{2}$$

If we consider the data transmission only, the receiver rather than the transmitter should measure the channel idle probability. In practical, the link layer often requires the receiver to transmit the ACK message back error-free (without collision). Thus, the idle time should be counted when the wireless channels around both the transmitter and the receiver are idle, which is difficult to measure. Here, since the transmitter and receiver are in close vicinity, a practical yet acceptable approach is to set the link idle probability to be the minimum of the channel idle probability monitored by the transmitter and that by the receiver, i.e., $P_{idle}(l_{AB}) = \min\{P_{idle}(A), P_{idle}(B)\}$, where l_{AB} represents the link between node A and node B. We can then estimate the expected available bandwidth of the link, ELB_l, by

$$ELB_l = C_l \times P_{idle}(l), \tag{3}$$

where C_l is the estimated average link data rate.

3.3 Expected clique bandwidth

For WMNs, the ideal routing protocol should find paths leading to the highest throughput and thus the wireless resources can be fully utilized. In a multi-hop wireless network, how to select the highest throughput path is a challenging issue due to the inter- and intra-flow interference. In other words, the multiple-hop transmissions for the same flow and the transmissions for other flows by interfering nodes need to be considered in determining the path throughput.[2]

To analyze the interference of the links, we should introduce the concepts of conflict graph and clique first. A conflict graph [53] is helpful to identify the cliques. To construct the conflict graph G_c of a network, each link is converted to a vertex in G_c, and two vertices are connected with an edge if they conflict with each other (i.e., those links with interference high enough to result in a collision). In an undirected graph $G_c = (V, E)$ where V is the vertex set and E is the edge set, a clique is a subset of the vertex and link set, $C \subseteq V$, such that for every two vertices in C, there exists an edge connecting the two , which is totally meets the definition of clique, namely, a subset is such a complete graph that the nodes together with any edges whose endpoints are both in this subset. A maximal clique is a clique that cannot be extended by including one more adjacent vertex. Figure 2(a) is an example network where the interference range of the node is twice of the transmission range and the links not adjacent may interfere with each other such as SA and CE, SG and FD. Its conflict graph is drawn and shown in Fig. 2(b) and according to it, we can find that there are 5 cliques $(c_1, c_2, c_3, c_4, c_5)$ in the network.

Then we use the example shown in Fig. 2(a) to illustrate the intra-flow interference problem in the WMNs. In the network, there are seven nodes and seven links with different available bandwidth (numbered next to the links). Consider a source and destination pair of nodes, S and D, and the uni-directional links from left to right only. Obviously, the maximum flow throughput over a link cannot exceed the link available bandwidth. In a multi-hop wireless network, the flow throughput can be much smaller than the link bandwidth, as multiple links may share the wireless channel.[3] At any time, a collision occurs and the transmissions fail if more than one links in a clique are transmitting simultaneously. Since all the links in a clique share the wireless resources, the expected available bandwidth (i.e.,the maximum throughput for a flow going through all the links) of clique c, namely ECB_c, is given by the following expression:

[2] In this paper, we use the terms bandwidth and throughput interchangeably.

[3] We use the term link to refer to the logic point-to-point connection between a transmitter and the receiver, and the term wireless channel to refer to the wireless medium that can carry information over certain space.

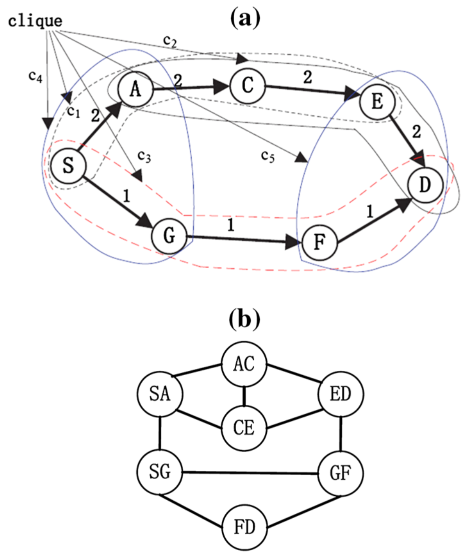


Fig. 2 A sample WMN with cliques $c_1 = (SA, AC, CE), c_2 = (AC, CE, ED), c_3 = (SG, GF, FD), c_4 = (SA, SG), c_5 = (ED, FD)$. **a** Topology of a sample network. **b** Conflict graph of the network

$$ECB_c = \left(\sum\nolimits_{l \in c} \frac{1}{ELB_l}\right)^{-1}, \tag{4}$$

where l is a link in clique c and ELB_l is the expected available bandwidth of link l, which has been discussed in the last subsection.

According to the conflict graph Fig. 2(b), the clique c_1 is formed by the links SA, AC and CE, and the expected available bandwidth of the clique is $ECB_{c_1} = (\frac{1}{2} + \frac{1}{2} + \frac{1}{2})^{-1} = \frac{2}{3}$ Mbps. With the same approach, we can get $ECB_{c_2} = \frac{2}{3}$ Mbps and $ECB_{c_3} = \frac{1}{3}$ Mbps.

3.4 Expected path bandwidth

Now we will introduce the estimation of the end-to-end expected path throughput of path P_i, namely EPT_{P_i}.

The bottleneck clique of path P_i will limit the maximum throughput for a flow through this path, so we can adopt the bandwidth of the bottleneck clique as EPT_{P_i}, i.e. EPT_{P_i} equals the minimum ECB_c in the path, given by

$$EPT_{P_i} = \min_{c \in C_{P_i}} (ECB_c), \tag{5}$$

where C_{P_i} is the set of cliques in path P_i. In the sample network illustrated in Fig. 2(a), two paths from S to D are P_1={S, G, F, D} and P_2={S, A, C, E, D} and the expected path bandwidth should be $EPT_{P_1} = \min(c_3) = \frac{1}{3}$ Mbps and $EPT_{P_2} = \min(c_1, c_2) = \frac{2}{3}$ Mbps. Based on this metric, P_2 is preferable to P_1 and the shortest path in terms of the hop count may not achieve the highest throughput.

But there is one shortcoming for (5). For example, in Fig. 2(a), if all links have the same capacity 2 Mbps, the bandwidth of the bottleneck cliques of the paths, P_1 and P_2, are both $\frac{2}{3}$ Mbps, so that we cannot find out the better path. But apparently, now path P_1 with three hops is better than path P_2 with four hops. Hence, (5) cannot reflect the influence of the hop count.

To study the influence of the hop count, a group of experiments are simulated in NS-2. All nodes have an interference range twice of the transmission range and 3 Mbps transmission rate, and other parameters are illustrated in Sect. 5. The nodes are distributed linearly with step equalling the transmission range and the number of nodes varies from 2 to 12. Figure 3(a), (b) are the topology and conflict graph respectively. Note that every node added to the path will introduce new cliques, which may change the bottleneck. The first node will sent traffic to the last node and the throughput results are shown in Fig. 4.

According to the conflict graph, we can find that the bottleneck clique bandwidth should be $(\frac{1}{l1})^{-1} = 3$ Mbps for 2 hops, $(\frac{1}{l1} + \frac{1}{l2})^{-1} = 1.5$ Mbps for 3 hops and $(\frac{1}{l1} + \frac{1}{l2} + \frac{1}{l3})^{-1} = 1$ Mbps for 4 and more hops, i.e. the expected throughput curve in Fig. 4. However, according to the experiments throughput curve in Fig. 4, for 5 or more hops, the throughput is lower than the expected throughput and

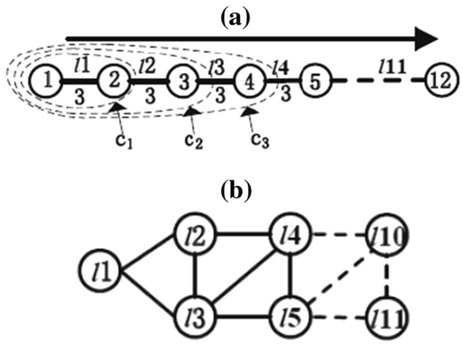


Fig. 3 Experiments to study the influence of the hop count. **a** Nodes distributed linearly. **b** Conflict graph of the linear topology

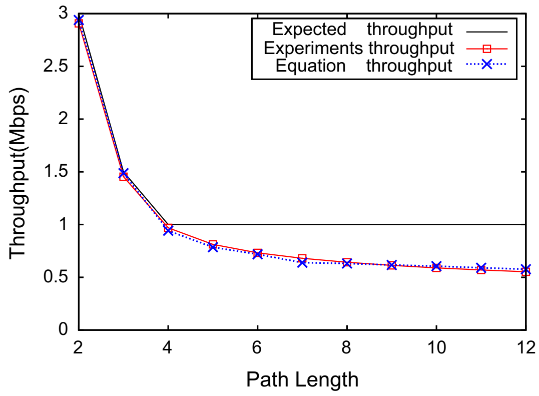


Fig. 4 Throughput of experiments on network Fig. 3

reduces as the hop count increasing. It is found that within 4 hops, the bottleneck cliques c_1 for 2 hops, c_2 for 3 hops and c_3 for 4 hops are introduced by node 2, 3 and 4 respectively, i.e. the bottleneck is always the clique introduced by the current node itself. But for 5 or more hops, the bottleneck is always the clique c_3 which is introduced by node 4. That means the bottleneck clique is attenuating as the path becomes longer. More hops will cause more underlying collisions and the collision recovering in each node will also decrease the throughput.

Further, for two paths with the same bottleneck link, the longer path will achieve less throughput. This is because that the dominant transport layer protocol, TCP, cannot guarantee the co-existing TCP flows with different round-trip time can converge to their fair shares, and flows with a larger round-trip time will open up their congestion window slower. To consider the influence of the path length, we use (6) and (7) to calculate EPT_{P_i} as follows.

$$EPT_{P_i} = \min\{f(h) \times ECB_{(P_i - D)}, \min_{c \in C_D} ECB_c\}, \tag{6}$$

and

$$f(h) = a \cdot (h+1)^b, \tag{7}$$

where $ECB_{P_i - D}$ is the bandwidth of the bottleneck clique formed by $(P_i - D)$, which represents the set of nodes except the destination node D itself, and h is the hop count between the current node itself and the node who have introduced the bottleneck clique. C_D is the set of cliques lately introduced by the destination D and a and b are constants to determine the decay. Equation (7) is used to estimate the influence of the increasing hop count. EPT_{P_i} is the minimum clique bandwidth of the path, considering the decay caused by the increasing hop count. It is defined recursively and can be calculated in a distributed way as described in Sect. 4.2. The values of a and b are obtained empirically from extensive simulations.

Then according to (6) and (7), we calculate the path bandwidth and present the results by the equation throughput curve in Fig. 4 where a and b are set 0.9692 and -0.2556 respectively. The equation matches the experimental results well. So the problem raised at the beginning of this subsection can be solved that P_1 (four hops) with $\min\{0.78 * 1.5, \frac{2}{3}\} = 0.67$ Mbps is better than P_2 (five hops) with $\min\{0.78 \cdot \frac{2}{3}, \frac{2}{3}\} = 0.52$ Mbps.

Based on the above, we obtain the expected path throughput (EPT) as the maximum bandwidth among all the paths from a source node S to its destination D,

$$EPT(S, D) = \max_{P_i}(EPT_{P_i}), \tag{8}$$

and the expected bandwidth of a path P_i has considered both the bottleneck clique in the path and the influence of the hop count, as given in (6).

The clique introduced by the shared wireless channel substantially increases the complexity of finding out a route with the maximum path bandwidth in a hop-by-hop manner. It is an NP-hard problem to find out all the cliques in a graph, so we tackle the problem in three steps: (a) calculating the link capacity (or average data rate) which is determined by the physical channel characteristics such as SNR; (b) considering the inter-flow interference to determine the available link bandwidth; and (c) considering the clique due to the intra-flow interference and the influence of the hop count to determine the EPT using (6). Then, we can design a heuristic, distributed routing algorithm using EPT as the metric, and the details will be given in Sect. 4.

4 Expected path throughput based routing protocol design

Based on the EPT routing metric discussed in Sect. 3, we design a distributed routing protocol, called EPTR, which combines proactive and reactive routing mechanisms. EPTR starts route discovery on user's demand, and also uses a periodic maintenance mechanism to consider the network dynamics. The main components of EPTR include route discovery, route calculation, route maintenance and route recovery, as discussed in the following subsections.

4.1 Route discovery

In EPTR, route discovery operates in an on-demand way to reduce the overhead. When a source node S needs to send data to a destination node D, it initiates the route discovery process. It first broadcasts a routing request message (RREQ). When a neighboring node receives the RREQ, it first measures the SNR, and use the smoothed SNR to estimate the link quality and calculate the ELB of the link according to the monitored link idle probability. Then it checks the cliques in the path from the source node to itself, and calculates the EPT of the path and broadcasts again until the destination receives the RREQ.

Each RREQ message, as shown in Table 2(a), contains the unique sequence ID of the RREQ, the minimum clique bandwidth ECB from the source to the node itself, which is currently processing the RREQ, the hop count h between the node itself and the node who have introduced the current bottleneck clique, and the path vector. The path vector includes the node address, the P_{idle} of the link to reach that node, and the list of neighbors who belong to the path, $N_P(i)$. The neighbor list is used to assist the following nodes to identify the intra-flow clique to calculate EPT. Thus, the neighbor list contains the tagged node's neighbors who are also part of the path from the source to the tagged node.

Table 2 Composition and procedure of RREQ

(a) Composition of RREQ

RREQ *id*		$\min ECB(u)$	h
Source node s	P_{idle_s}	$N_P(s)$	$ELB(s,\ s)$
Node 1	P_{idle_1}	$N_P(1)$	$ELB(s,\ 1)$
...	...	...	...
Node i	P_{idle_i}	$N_P(i)$	$ELB(i-1, i)$
...	...	...	...
Node u	P_{idle_u}	$N_P(u)$	$ELB(u-1, u)$

(b) RREQ produced in node D

k		4	1
S	80 %	G	∞
G	80 %	S, F	8
F	80 %	G, D	8
D	80 %	F	9.6

In case a node receives an RREQ with the same ID as the one it received before, it calculates and compares the EPT. If the current EPT is smaller than the previous one, the new RREQ is discarded; otherwise, it updates the information using the new path and then sends out an RREQ to its neighbor. The process is repeated until the RREQ message reaches the destination. Once the destination receives the RREQ messages from its neighbors, it first measures the SNR and calculates EPT of the path, and compares the EPT values in the RREQs with same ID and selects the path with the maximum EPT. It waits for a constant time T to get more RREQs, then chooses the best route to constructs the routing reply packet (RREP), and sends RREP which includes the path vector and EPT information back to the source node along the reversed path.

We use an example shown in Fig. 5 to illustrate the procedure. Assume $P_{idle} = 0.8$ for each link, and the *ELB* of each link is labeled on the link. At the beginning, node S broadcasts the RREQ message which includes the route discovery request from S to the destination D, and also contains a unique RREQ ID. When G receives the message, it calculates *ELB* of link *SG* and the EPT of path $\{S, G\}$ according to (3), (4) and (6). Then, it adds the information of its own address, idle probability, neighbor list, the *ELB* of link *SG* (8 Mbps), the minimum *ECB* of path $\{S, G\}$ (8 Mbps) and the hop count h (0 hops, itself) into the RREQ and broadcasts it after the timer expires. This procedure is repeated by F which identifies the clique (*SG*, *GF*) with $ECB = (\frac{1}{10*0.8} + \frac{1}{10*0.8})^{-1} = 4\,\text{Mbps}$, and calculates the EPT of path $\{S, G, F\}$ to be 4 Mbps. Finally, D receives the RREQ and identifies the clique (*GF*, *FD*) with $ECB = (\frac{1}{10*0.8} + \frac{1}{12*0.8})^{-1} = 4.36\,\text{Mbps}$ which is greater than the attenuated clique (*SG*, *GF*) equalling $f(1) * 4 = 3.12\,\text{Mbps}$. Then the EPT of the identified path $\{S, G, F, D\}$ is obtained 3.12 Mbps, and the path information is sent back to S using the reversed path $\{D, F, G, S\}$. The RREQ with id k produced at the destination is shown in Table 2(b).[4]

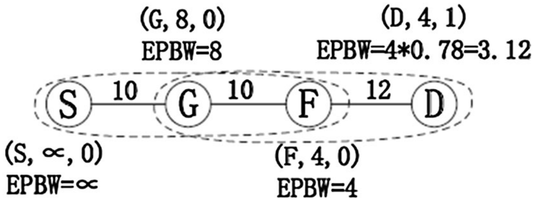


Fig. 5 Path vector in RREQ message

4.2 Route calculation

To find the maximum throughput path, some existing routing protocols used modified Dijkstra algorithm based on a centralized routing database to design the routing algorithm, which may not be scalable [54]. It is desirable to design a distributed algorithm so each node calculates the EPT from the source to itself and selects the best path in a distributed manner. Our proposed distributed route discovery algorithm is given in Algorithm 1.

Algorithm 1 Route discovery algorithm by node v

```
Require: $RREQ(n,u)$ message, including path vector $P_u$
    composed by triples of links $L_{ij}$ and expected band-
    width of links $ELB_{L_{ij}}$, neighbor set of node $i$ is $N_i$ where
    $i$ belongs to the path, $h$ is the hop count between the
    node itself and the node who introduced the bottleneck
    clique.
Ensure: path from $s$ to $v$ with maximum expected band-
    width $EPT$
Node $v$ receives $RREQ(n,u)$;
Convert path $P_v$ into a conflict graph $G_v$;
delete vertex $N_{kl}$ that is not connected to $N_{uv}$ in $G_v$;
$C_v = Bron\ Kerbosch(G_v)$;
$EPT_{new}(v) = \min(Cliquebandwidth(C_v))$;
$h = h + 1$;
$EPT_{att}(u) = fall(h) \cdot minECB(u)$;
if $EPT_{new}(v) < EPT_{att}(u)$ then
    $h = 0$;
    $minECB(v) = EPT_{new}(v)$;
else
    $EPT_{new}(v) = EPT_{att}(u)$;
    $minECB(v) = minECB(u)$;
end if
if $EPT(v) < EPT_{new}(v)$ then
    $EPT(v) = EPT_{new}(v)$;
    reconstruct $RREQ(n,v)$, broadcast $RREQ(n,v)$;
end if
```

[4] To decrease the storage space, the neighbor list excludes the predecessor node and successor node in the known part of the whole path.

As the EPT is mainly determined by the cliques along the path, we use the conflict graph to identify the cliques. We first construct the conflict graph G_v of the routing path P_v according to the rules introduced before. The well-known Bron and Kerbosch's algorithm can compute all cliques in a graph in linear time (w.r.t. the number of cliques) [55]. In line 4 of Algorithm 1, function $Bron\_Kerbosch(G_v)$ adopts this algorithm to compute all the new constructed cliques in the conflict graph G_v. Note that, when one more hop is added to the path, new cliques associated with the last-hop link are added in the conflict graph. In order to distribute the EPT computing load to all nodes along the path, each node only needs to find out the new cliques containing the vertex corresponding to the link where the tagged node is the receiver, so other vertexes not connected to the new vertex can be removed from the conflict graph, as shown in line 2. Then, according to (6), in lines 4–14, each node updates EPT by comparing the minimum *ECB* of new cliques (line 5) with the attenuated EPT of the path before current node (line 7) where $fall(h)$ is calculated by (7). At the same time, the minimum *ECB* is also updated (line 10 and 13). Noted that if a new clique becomes the bottleneck, h which is the hop count for the bottleneck clique should be updated in line 9. Finally, the node constructs the RREQ message and sends the message out once the timer expires.

Algorithm 2 $Cliquebandwidth(C_{L_{ij}})$

```
Require: $C_{L_{ij}}$ that includes $ELB$ of each link in the clique.
Ensure: calculate out the $ECB$ of clique $C_{L_{ij}}$
$T(C_{L_{ij}}) = 0$;
for $L_{ij} \in C_{L_{ij}}$ do
    $T(C_{L_{ij}}) = T(C_{L_{ij}}) + \frac{1}{ELB(L_{ij})}$;
    $i = j; j = succeed(j)$;
end for
$ECB(C_{L_{ij}}) = (T(C_{L_{ij}}))^{-1}$;
return $ECB(C_{L_{ij}})$;
```

Function $cliquebandwidth(C_{L_{ij}})$ is given in Algorithm 2, which calculates the expected bandwidth of the clique that includes L_{ij}, as shown in Fig. 6. The method is presented in Sect. 3.3. $EPT(v)$ *and* $EPT_{new}(v)$ present the end-to-end expected bandwidth of path p_v and the minimum ECB of the new cliques respectively. $RREQ(n, u)$ presents *RREQ* message with id n sent by node u.

The destination also does the same work and selects the path with the highest EPT and returns the path to the source. The source-routing information is carried in the RREP message which includes the ID, the path vector and EPT. The measurements of the ELB and idle probability of each link are also updated.

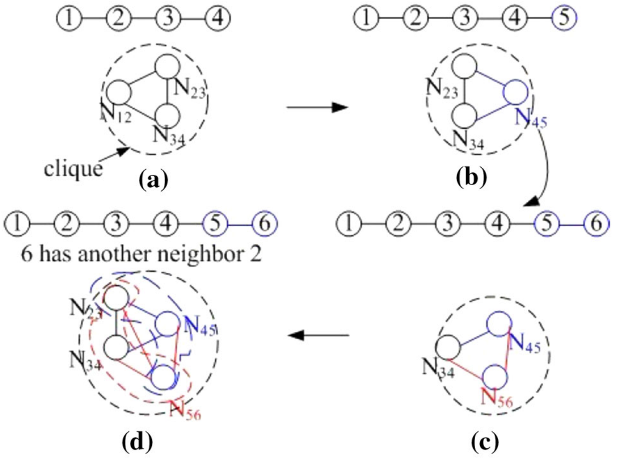


Fig. 6 Route transforms into the conflict graph, N_{ij} represents link L_{ij}, **a** route with 4 nodes and 3 links transforms into conflict graph, and all links are in a clique. **b** after adding node 5, only one new clique created. **c** after adding node 6 with node 2 is its neighbor, which creates 3 new cliques, and the big one includes other two

Although, in general, the complexity of clique computing in a graph increases exponentially with the number of nodes, the number of hops h in a path is typically small (e.g., $h < 16$), the number of cliques in a path is limited by $3^{(h/3)}$ according to Moon and Moser [56], which equals 243 for $h = 15$. Thus, the computational load of finding all maximal cliques in a path is tolerable for each router. Usually, the path with a higher throughput transfers RREQ faster so the number of rebroadcast for RREQ message is low. In addition, from the path vector, we can find out the bottleneck clique in the path to make a decision on whether to rebroadcast or just change the path vector, as the neighbor information has been carried in RREQ which decreases the rebroadcast message numbers. Thus, the routing protocol message exchange overhead is reduced.

4.3 Route maintenance

After route discovery, the source successfully obtains a path to deliver data. As the wireless network changes dynamically, the path throughput may become lower or even worse, the path may fail; or, a better path may occur. In these cases, a new suitable path should be found.

When the destination notices a substantial decrease of the throughput, it sends a triggering message (TREQ) to inform the source and then the source triggers a new route discovery process by sending out a routing update request message (UREQ). In the case of link broken, the source will detect it due to timeout, and then it will also sends a UREQ message to find a new path. Once a new path with a larger EPT is found, the source switches to the new path for the following data transmissions. However the above procedure cannot find a better route when the current path works well but other better routes occur. In this case, we also use a periodical route discovery mechanism with a low frequency to find a better route if possible, where the update period is determined based on the tradeoff of route optimality and message overhead.

During this procedure, if we do not carefully estimate the load introduced by the tagged flow, the oscillation may occur. That is, if there are two similar paths available for a flow, the idle probabilities of the links belonging to the current path are reduced due to the transmissions for the tagged flow, so the routing protocol may choose the alternative path during the path update procedure; then, the alternative path becomes crowded, and so on and so forth.

To solve the oscillation problem, it is important that during the route update process, the impact of the tagged flow's load on the EPT of a path should be considered. One effective way is to recalculate the idle probability of each node by adding the busy time in transmitting the tagged flow to the idle time monitored as follows,

$$P_{idle}(A) = \frac{T_i(A) + T_t(f)}{T_b(A) + T_i(A)}, \tag{9}$$

where $T_t(f)$ is the busy time of the channel around node A in transmitting packets for flow f, $T_b(A)$ and $T_i(A)$, which are the busy time and idle time around node A respectively, that can be monitored at the MAC layer. As each mesh node needs to make a route selection decision, the source IP and the destination IP of each packet are extracted to identify whether it belongs to flow f or not. In this period, if flow f transmits $N_f(A)$ packets and packet k is transmitted at data rate R_k, $T_f(f)$ can be presented as

$$T_t(f) = P_{size} \cdot \left(n_1 \frac{1}{R_1} + \cdots n_k \frac{1}{R_k} + \cdots n_{N_{f(A)}} \frac{1}{R_{N_f(A)}} \right) \tag{10}$$

where n_k is the number of transmissions of packet k till it is transmitted successfully. We assume that all the packets are with the same size P_{size}.

4.4 Route recovery

Routing failure most likely occurs in the following two processes. One is the routing information being lost when it is sent back from the destination to the source, and the other is due to disconnection during the data transmitting process.

When the RREQ reaches the destination, it will return RREP with the path vector and the EPT to the source. Although the links in WMNs are not symmetrical, the reverse path from the destination to the source is typically connected, although the reverse path throughput can be different.

During the data transferring process, as the MAC protocol typically implements an acknowledgement mechanism, and packets will be retransmitted until the number of retransmissions reaches the retry limit. When the path becomes disconnected, the node who fails to transmit the packet will return a route error message (RERR) to the source node. The source node then initiates a new RREQ message to find a new path.

5 Evaluation and analysis

To evaluate our cross-layer proposal, we implemented the EPT metric in NS-2.31 network simulator [57] and compared the performance with ETX, ETT and the metric (CAB) which proposed in [45]. The parameters used in the simulations are listed in Table 3. The transmission range of each node is 250 m and the interference range is 500 m. The data traffic is CBR flow with fixed packet size of 1024 bytes. Other parameters not mentioned are the default ones in the 802.11b module of NS-2.31. The modulation schemes are adaptively changed among DBPSK, DQPSK, CCK55, CCK11, according to the measured SNR, which takes data rates of $1, 2, 5.5, 11\,\text{Mbps}$ based on Table 1, respectively. The distributed routing protocol is different from the solution in [41] and [43] where routes are calculated based on global information captured previously. We noted that the difference of end-to-end delay between valid routes is less than 100 ms, so the constant time T to wait for more RREQs before choosing the best route to construct RREP is set to 100 ms in this paper. Once the path throughput decreases exceeding the threshold of $20\,\%$, or otherwise every 20 s an UREQ message is sent out. All the simulation results presented in this section are the average of 10 simulation runs. In each simulation run, both the random number generator (RNG) and random seed are unchanged for every metric, which means the network and flow settings are the same for the comparing metrics. But between each simulation run, the random seed is set to the current time to make a difference.

To evaluate the performance of different distance of source/destination (short for S/D hereinafter) node pairs, we use the hop count in ETX as benchmark to classify data flows into groups and then compare each metric in the same group in which the S/D node pairs are the same for every metric. Let G_i denotes the S/D node pairs group with i hops in ETX and S_a/D_b denotes the S/D node pairs with source node a and destination node b. For instance, $G_2 = \{S_1/D_4, S_5/D_2, S_{11}/D_7\}$, then we find the same S/D pairs which belong to G_2 in the simulation of ETT, CAB and EPT to calculate the per-flow throughput and path length. Noted that the throughput in this section is measured over the total simulation time. In Figs. 7(b), 8(b) and

Table 3 Part simulation parameters

Parameter	Value	Parameter	Value
Simulation period	500 s	PHY specification	IEEE802.11 b
Route discovery period	20 s	Transmit power	0.031622777 W
Route update threshold	80 %	CS threshold	$9.011872e^{-12} * \vartheta$
Propagation model	TwoRayGround	Data traffic	FTP over TCP
Antenna mode	OmniDirectional	Interference traffic	CBR over UDP
Propagation-path loss	1	Packet size	1024 bytes

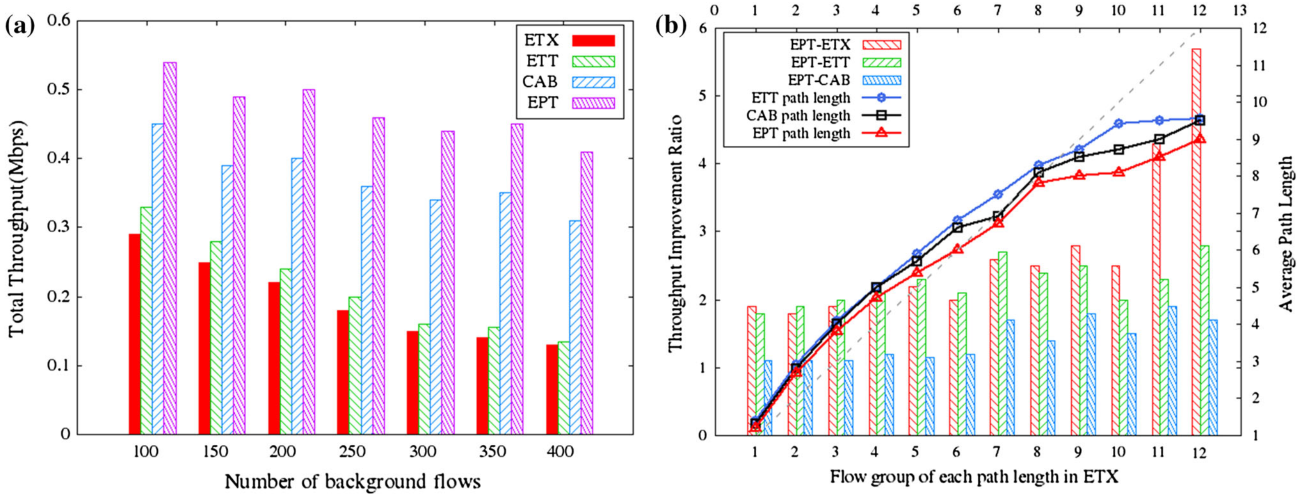


Fig. 7 Under 100 nodes grid network, EPT identifies the imbalance link with different available bandwidth and achieves more throughput. **a** Throughput of flows with different N_{Bg}. **b** Performance of each flow-group when $N_{Bg} = 200$

9, the x-axis represents flow group G_i which is classified by hop count in ETX, the number in the bracket after x-axis is the number of flows in each flow group, the first y-axis represents the per-flow throughput improvement ratio of EPT with ETX, ETT and CAB (denoted by EPT-ETX, EPT-ETT and EPT-CAB, respectively) while the second y-axis represents the per-flow average path length which is presented with lines.

5.1 Grid topology with imbalanced load

We deploy 100 nodes as a grid network with a fixed distance of 150 m between neighbors in a 1500 m × 1500 m square area. In order to set up imbalanced load in the network, we randomly generate background traffic similar to that in the experiments reported in [45]. The background traffic, which is 1-hop CBR flow with data rate following the uniform distribution $U(1, 20)$ Kbps, differ the available bandwidth of each link. The number of data flows is constant with a value of 200. We then increase the number of background flows (denoted by N_{Bg}) from 100 to 400 to have different degree of imbalanced network load.

In Fig. 7(a), the average total throughput of data flows with different background traffic N_{Bg} is depicted. With the N_{Bg} increasing, the throughput of ETX and ETT drops quickly as it do not consider the link load. But CAB performs better than ETX and ETT since it considers the available path bandwidth information. Meanwhile, EPT can identify the lightly loaded paths and it achieves highest throughput, which is higher about 25 % than that of CAB and close to two times higher than that of ETX and ETT when N_{Bg} is above 300. Hence, the EPT can balance the traffic load to utilize the link capacity efficiently to obtain higher throughput by monitoring the channel idle status. Figure 7(b) shows the performance of each metric when background traffic N_{Bg} is 200. The histograms indicate the average per-flow throughput improvement ratio of EPT with ETX, ETT and CAB while the lines indicate the average per-flow path length. For comparison, the dotted line which denotes the path length of ETX. We can observe that EPT performs the best with improvement ratio varying from 1 to 6, and it performs much better than CAB especially with the longer S/D node pairs. On the other hand, the average per-flow path length of EPT, CAB and ETT is longer than ETX when path length is within 6 hops. They

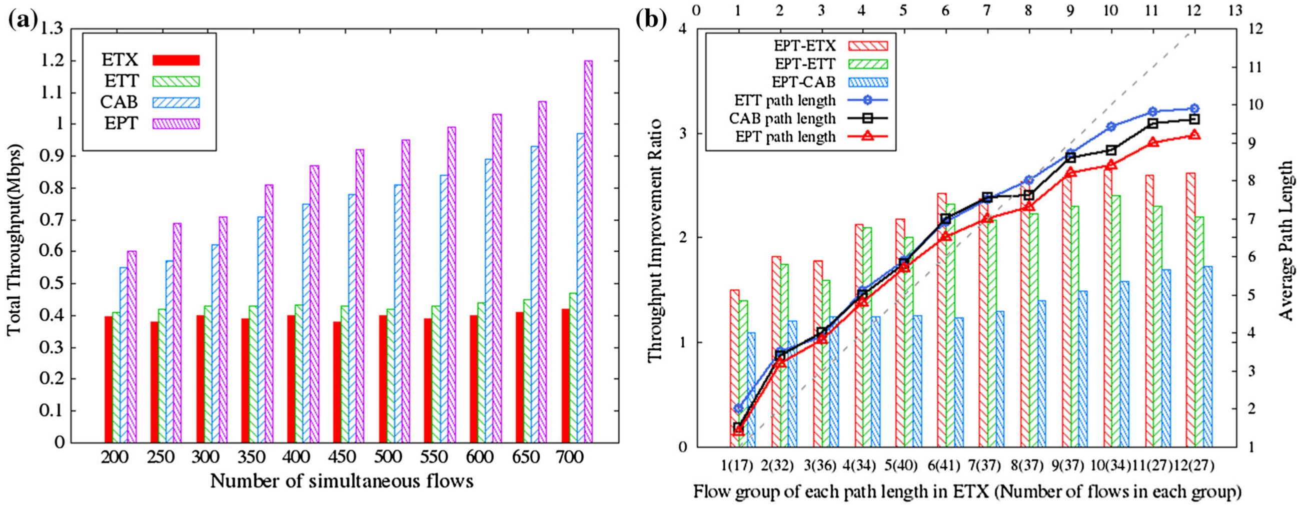


Fig. 8 100 nodes grid network in 1500 m × 1500 m with simultaneous data flows increasing from 200 to 700. **a** Total throughput of flows as the number of simultaneous data flows increasing. **b** Performance of each flow-group when $N_{Flow} = 400$

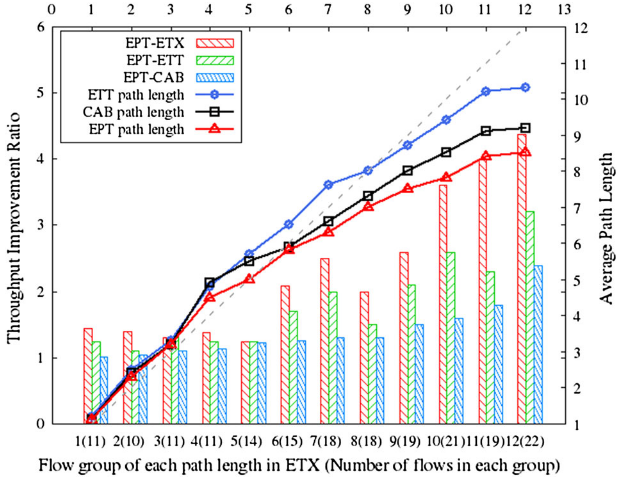


Fig. 9 Performance of each flow-group on random topology

consider the link capacity and adopt longer path to achieve higher throughput. When path length is longer than 7 hops, the path length of EPT, CAB and ETT is shorter than that of ETX and the path length of EPT is shorter than that of CAB and ETT. Simply chosen longer path will not lead to higher path capacity for intra-flow interference. EPT addresses this issue and chooses shorter path with higher capacity when the distance of S/D node pairs is reasonably long (7 hops in this simulation).

The average performance are show in Table 4, we can see that EPT outperforms CAB, ETT and ETX by 21,53 and 58 % in throughput respectively. Meanwhile, EPT decreases packets loss rate about 52, 50 and 25 % over ETX, ETT and CAB, and the end-to-end delay is also reduced about 2 ms than CAB. Since the EPT chooses a better path with higher available bandwidth.

5.2 Grid topology with varying number of flows

This subsection adopts the same topology as the previous simulation with a fixed N_{Bg} of 100. Next, we examine the performance of each metric by changing the number of data flows (randomly deployed and denoted by N_{Flow}) from 200 to 700.

The mechanism of cross-layer design in EPT improves the average throughput with the number of simultaneous data flows increasing. EPT considers not only the intra-flow but also the inter-flow interference, and combined with channel idle monitoring. As shown in Table 5, EPT could obtain the available bandwidth and achieve higher throughput with less packets loss rate and end-to-end delay. And we can see from the Fig. 8(a), with the N_{Flow} increasing, the total throughput of ETX and ETT increases a little bit while that of CAB and EPT improves greatly, and the improvement of EPT is greater than that of CAB. EPT results in higher network throughput thanks to its load-awareness. When some part of the network is congested, new flows can select paths avoiding the congested links, so the EPT can achieve better load balance and higher overall throughput. The average per-flow throughput improvement ratio and path length of each flow-group when $N_{Flow} = 400$ is shown in Fig. 8(b), EPT outperforms ETX, ETT and CAB in each flow-group and shows the superiority in longer S/D node pairs. Again, for EPT, some flows will choose longer paths to obtain higher throughput when the distance of S/D pairs is short (within 7 hops in this simulation) and the network load can be balanced. Thus, most of the flows with long paths (beyond 7 hops in this simulation) can select shorter paths with more throughput.

Table 4 Performance comparison of Grid topology with imbalanced load

Metric	Average throughput (Mbps)	Average delay (ms)	Average loss rate (%)
ETX	0.1928	70.86	4.91
ETT	0.2143	67.30	4.67
CAB	0.3671	62.84	2.66
EPT	0.4643	60.37	2.43

Table 5 Performance comparison of Grid topology with varying number of flows

Metric	Average throughput (Mbps)	Average delay (ms)	Average loss rate (%)
ETX	0.3959	42.60	2.96
ETT	0.4331	40.35	2.67
CAB	0.7363	37.78	1.91
EPT	0.8836	35.63	1.25

Table 6 Performance comparison of random topology

Metric	Average throughput (Mbps)	Average delay (ms)	Average loss rate (%)
ETX	0.2340	60.54	3.13
ETT	0.2892	58.15	2.86
CAB	0.4843	52.67	2.37
EPT	0.5812	50.24	1.75

5.3 Random topology

For the random topology, we use the *setdest* tool in NS-2 to generate a topology with 100 nodes randomly deployed in a 1500 m × 1500 m square area. The distance between each node is randomly selected, thus the path loss of each link is different. After setting up the scenario, we then start 200 data flows with a random duration of 5–25 s where the source and destination nodes are randomly selected and the data rates follow the pareto distribution with a mean of 10 Kbps. To be fair, we use the same network setting to compare the performance of different metrics. In order to compare them under varying channel condition, the data flows are classified according to the routing path length with ETX, and then we get the throughput and average routing path length of every set of data flows under ETX, ETT, CAB and EPT respectively.

As the results shown in Fig. 9 and Table 6, we can observe the similar tendency: all the improvement ratio is larger than 1 and the ratio is becoming even larger as the distance of S/D node pairs increasing, which means that our metric identifies the high throughput path on random topology and performs better especially with the longer S/D node pairs. On the other hand, the average per-flow path length of EPT, CAB and ETT is longer than ETX when path length is within 6 hops. They consider the link capacity and adopt longer path to achieve higher throughput. When path length is longer than 6 hops, EPT adopts shorter paths which have higher capacity to maximize the network overall throughput with the ability of load balance. And EPT still achieves much better performance than CAB, ETT and ETX: more than 0.5812 Mbps obtained of throughput while loss rate is as low as 1.75% (compared to that of CAB, ETT and ETX are 3.13, 2.86 and 2.37 %), and achieves about 6.5, 16 and 19 % reduction on end-to-end delay than CAB, ETT and ETX as shown in Table 6. Therefore, EPT shows good performance and adaptivity in a larger network with dynamical network load.

5.4 Protocols comparison

In the last experiment, we compare the throughput and routing overhead of different routing protocols. In the topology where 7 × 7 nodes are located in a 1200 m × 1200 m square area and the distance between each node is 150 m, three gateways and nine source nodes are selected randomly. During the 500 s simulation, the source nodes sent FTP traffic over TCP to one gateway which is randomly chosen and random background traffic occurred from 100 to 200 s and from 300 to 400 s. The average total throughput of EPTR, DSR, AODV and DSDV are shown in Fig. 10(a). It can be seen that EPTR outperforms the other protocols with more stable and higher throughput. The throughput of DSR and AODV decrease rapidly at the presence of interference traffic and DSDV performs poorly. The comparison shows that EPTR has better adaptivity

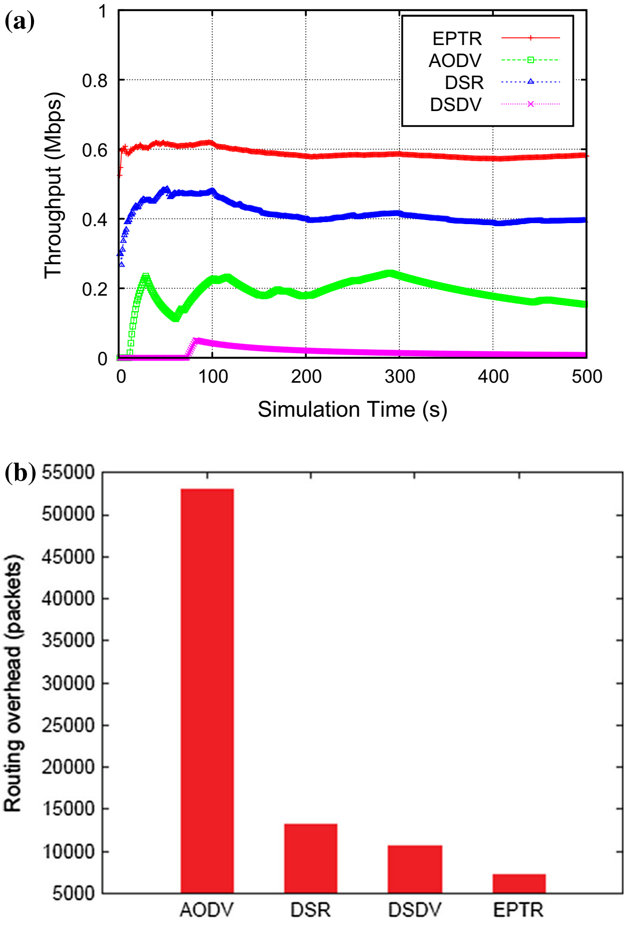


Fig. 10 EPTR vs. AODV, DSR, and DSDV. **a** Throughput comparison. **b** Message overhead comparison

than the other protocols. Figure 10(b) compares the routing overhead, and AODV has much higher overhead than others, because each node needs to broadcast HELLO message periodically to maintain the neighbor list and in the experiment all nodes sent over 24,000 HELLO packets which cause more conflicts leading to a lower efficiency. Although the method of DSR is similar to that of AODV, it only maintains the nodes on the active paths. Besides, it uses route cache mechanism to reduce the broadcast range and number which substantially decreases the routing overhead. The overhead of DSDV, which is a proactive routing protocol, increases rapidly as the network scale turns large even though nodes only broadcast the known routing information to their neighbors. By combining routing discovery triggering and long period route update mechanisms, the distributed EPTR greatly decreases the overhead up to 690, 89, 52 % over AODV, DSR, DSDV respectively, which is also one reason for the better performance achieved by EPTR.

In summary, we have the following main observations:

- EPT identifies the lightly-loaded links of the network. Hence it achieves better load balance.
- With the awareness of inter-flow interference and better load balance, EPT can achieves higher network throughput, shorter end-to-end delay and less packet loss ratio than that of ETX, ETT and CAB.
- EPT achieves higher per-flow performance than that of ETX, ETT and CAB when the channel condition of each link varies.
- By combining routing discovery triggering and long period route update mechanisms, EPTR outperforms other routing protocols with less routing overhead.

6 Conclusion

In this paper, we have proposed a routing metric, expected path bandwidth (EPT) and a routing protocol, EPTR. For EPT, besides the impact of the physical layer on the link quality, the inter-and intra-interference and the path length are considered. To cooperate with the proposed EPT and balance the network load, a distributed routing protocol, EPTR, is developed. Finally, we evaluate the performance of the proposed protocol EPTR in NS-2. The results show that EPT and EPTR can improve the network capacity and achieve better performance than other metrics, and other routing protocols including AODV, DSR and DSDV. In the future, study of the performance in the fading channel, the feasibility of the metric under multi-interface network, combination with end-to-end delay into the metric to meet the QoS requirements remain open issues worth further investigation.

Acknowledgments The authors acknowledge the support of National Natural Science Foundation of China projects of Grant Nos. 61379058, 61379057, 61350011.

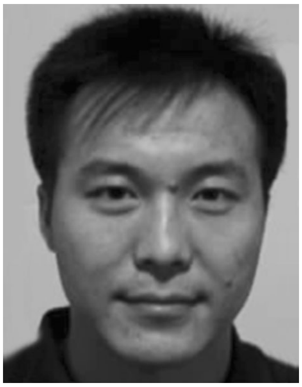
References

1. Deng, X., Liu, Q., Cai, L., & Chen, Z. (2013). Channel quality and load aware routing in wireless mesh network. In *2013 IEEE on wireless communications and networking conference (WCNC)* (pp. 2068–2073).
2. Deng, X., Liu, Q., Li, X., Cai, L., & Chen, Z. (2013). A channel quality and load sensitive routing in wireless mesh network. *Chinese Journal of Computers*, *36*(10), 1–11.
3. Vasilakos, A., Ricudis, C., Anagnostakis, K., Pedryca, W., & Pitsillides, A. (1998, May). Evolutionary-fuzzy prediction for strategic QoS routing in broadband networks. In *Fuzzy systems proceedings, 1998. The 1998 IEEE international conference on IEEE world congress on computational intelligence* (Vol. 2, pp. 1488–1493).
4. Busch, C., Kannan, R., & Vasilakos, A. V. (2012). Approximating congestion+ dilation in networks via "Quality of

Routing" games. *IEEE Transactions on Computers*, *61*(9), 1270–1283.

5. Woungang, I., Dhurandher, S. K., Anpalagan, A., & Vasilakos, A. V. (2013) . *Routing in opportunistic networks*. New York: Springer.
6. Li, P., Guo, S., Yu, S., & Vasilakos, A. V. (2012, March). CodePipe: An opportunistic feeding and routing protocol for reliable multicast with pipelined network coding. In *2012 Proceedings IEEE INFOCOM* (pp. 100–108).
7. Li, P., Guo, S., Yu, S., & Vasilakos, A. V. (2014). Reliable multicast with pipelined network coding using opportunistic feeding and routing. *IEEE Transactions on Parallel and Distributed Systems*, *25*(12), 3264–3273.
8. Demestichas, P. P., Stavroulaki, V. A. G., Papadopoulou, L. M., Vasilakos, A. V., & Theologou, M. E. (2004). Service configuration and traffic distribution in composite radio environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *34*(1), 69–81.
9. Attar, A., Tang, H., Vasilakos, A. V., Yu, F. R., & Leung, V. (2012). A survey of security challenges in cognitive radio networks: Solutions and future research directions. *Proceedings of the IEEE*, *100*(12), 3172–3186.
10. Youssef, M., Ibrahim, M., Abdelatif, M., Chen, L., & Vasilakos, A. V. (2014). Routing metrics of cognitive radio networks: A survey. *IEEE Communications Surveys & Tutorials*, *16*(1), 92–109.
11. Fadlullah, Z. M., Taleb, T., Vasilakos, A. V., Guizani, M., & Kato, N. (2010). DTRAB: Combating against attacks on encrypted protocols through traffic-feature analysis. *IEEE/ACM Transactions on Networking (TON)*, *18*(4), 1234–1247.
12. Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., & Qiu, D. (2014). Security of the Internet of Things: Perspectives and challenges. *Wireless Networks*, *20*(8), 2481–2501.
13. Sheng, Z., Yang, S., Yu, Y., Vasilakos, A., Mccann, J., & Leung, K. (2013). A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. *IEEE Wireless Communications*, *20*(6), 91–98.
14. Yan, Z., Zhang, P., & Vasilakos, A. V. (2014). A survey on trust management for Internet of Things. *Journal of Network and Computer Applications*, *42*, 120–134.
15. Liu, J., Wan, J., Wang, Q., Deng, P., Zhou, K., & Qiao, Y. (2015). A survey on position-based routing for vehicular ad hoc networks. *Telecommunication Systems*. doi:10.1007/s11235-015-9979-7.
16. Deng, X., He, L., Li, X., Liu, Q., Cai, L., & Chen, Z. (2015). A reliable QoS-aware routing scheme for neighbor area network in smart grid. *Peer-to-Peer Networking and Applications,* 1–12.
17. Akyildiz, I. F., Wang, X., & Wang, W. (2005). Wireless mesh networks: A survey. *Computer Networks*, *47*(4), 445–487.
18. Perkins, C., & Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers (Vol. 24, pp. 234–244). ACM.
19. Perkins, C., & Royer, E. (1999). Ad-hoc on-demand distance vector routing. In *Second IEEE workshop on mobile computing systems and applications, 1999, Proceedings, WMCSA99* (pp. 90–100).
20. Clausen, T., Dearlove, C., Jacquet, P., & Herberg, U. (2006). The optimized state routing protocol version 2. *draft-ietf-manet-olsrv2-00, Work in progress.*
21. Johnson, D. B., & Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, *353*, 153–181.
22. Yen, Y. S., Chao, H. C., Chang, R. S., & Vasilakos, A. (2011). Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs. *Mathematical and Computer Modelling*, *53*(11), 2238–2250.
23. Zhang, X., Zhang, Y., Yan, F., & Vasilakos, A. (2015). Interference-based topology control algorithm for delay-constrained mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, *14*(4), 742–754.
24. Meng, T., Wu, F., Yang, Z., Chen, G., & Vasilakos, A. (2015). Spatial reusability-aware routing in multi-hop wireless networks. *IEEE Transactions on Computers*. doi:10.1109/TC.2015.2417543.
25. Liu, A., Jin, X., Cui, G., & Chen, Z. (2013). Deployment guidelines for achieving maximum lifetime and avoiding energy holes in sensor network. *Information Sciences*, *230*, 197–226.
26. Han, K., Luo, J., Liu, Y., & Vasilakos, A. V. (2013). Algorithm design for data communications in duty-cycled wireless sensor networks: A survey. *IEEE Communications Magazine*, *51*(7), 107–113.
27. Yao, Y., Cao, Q., & Vasilakos, A. V. (2013, October). EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for wireless sensor networks. In *2013 IEEE 10th international conference on mobile ad-hoc and sensor systems (MASS)* (pp. 182–190).
28. Xiang, L., Luo, J., & Vasilakos, A. (2011, June). Compressed data aggregation for energy efficient wireless sensor networks. In *2011 8th annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks (SECON)* (pp. 46–54).
29. Liu, L., Song, Y., Zhang, H., Ma, H., & Vasilakos, A. V. (2015). Physarum optimization: A biology-inspired algorithm for the steiner tree problem in networks. *IEEE Transactions on Computers*, *64*(3), 819–832.
30. Jiang, L., Liu, A., Hu, Y., & Chen, Z. (2015). Lifetime maximization through dynamic ring-based routing scheme for correlated data collecting in WSNs. *Computers & Electrical Engineering*, *41*, 191–215.
31. Liu, Y., Liu, A., & Chen, Z. (2015). Analysis and improvement of send-and-wait automatic repeat-reQuest protocols for wireless sensor networks. *Wireless Personal Communications*, *81*(3), 923–959.
32. Dvir, A., & Vasilakos, A. V. (2011). Backpressure-based routing protocol for DTNs. *ACM SIGCOMM Computer Communication Review*, *41*(4), 405–406.
33. Vasilakos, A. V., Zhang, Y., & Spyropoulos, T. (Eds.). (2011). Delay tolerant networks: Protocols and applications. Boca Raton: CRC Press.
34. Spyropoulos, T., Rais, R. N., Turletti, T., Obraczka, K., & Vasilakos, A. (2010). Routing for disruption tolerant networks: Taxonomy and design. *Wireless Networks*, *16*(8), 2349–2370.
35. Liu, J., Wan, J., Wang, Q., Li, D., Qiao, Y., & Cai, H. (2015). A novel energy-saving one-sided synchronous two-way ranging algorithm for vehicular positioning. *Mobile Networks and Applications*. doi:10.1007/s11036-015-0604-5.
36. Liu, Y., Xiong, N., Zhao, Y., Vasilakos, A. V., Gao, J., & Jia, Y. (2010). Multi-layer clustering routing algorithm for wireless vehicular sensor networks. *IET Communications*, *4*(7), 810–816.
37. Zeng, Y., Xiang, K., Li, D., & Vasilakos, A. V. (2013). Directional routing and scheduling for green vehicular delay tolerant networks. *Wireless Networks*, *19*(2), 161–173.
38. Zhang, Z., Wang, H., Vasilakos, A. V., & Fang, H. (2012). ECG-cryptography and authentication in body area networks. *IEEE Transactions on Information Technology in Biomedicine*, *16*(6), 1070–1078.
39. De Couto, D., Aguayo, D., & Morris, R. (2003). A high-throughput path metric for multi-hop wireless routing. In *MobiCom (Ed.)* (pp. 134–146). ACM.
40. Bicket, J., Aguayo, D., & Morris, R. (2005). Architecture and evaluation of an unplanned 802.11 b mesh network. In *Proceedings of the 11th annual international conference on mobile computing and networking* (pp. 31–42).

41. Draves, R., Padhye, J., & Zill, B. (2004). Routing in multi-radio, multihop wirelessmesh networks. In *MobiCom (Ed.)* (pp. 114–128). ACM.
42. Langar, R., Bouabdallah, N., & Boutaba, R. (2009). Mobility-aware clustering algorithms with interference constraints in wireless mesh networks. *Computer Networks*, *53*(1), 25–44.
43. Salonidis, T., & Garetto, M. (2007). Identifying high throughput paths in 802.11 mesh networks: A model-based approach. In *IEEE international conference on network protocols, 2007, ICNP 2007* (pp. 21–30).
44. Kowalik, K., & Keegan, B. (2007). Rare C resource aware routing for mesh. In *IEEE international conference communications 2007, ICC07* (pp. 4931–4936).
45. Ronghui, H., King-Shan, L., & Jiandong, L. (2012). Hop-by-hop routing in wireless mesh networks with bandwidth guarantees. *IEEE Transactions on Mobile Computing*, *11*(2), 264–277.
46. Sarr, C., Chaudet, C., & Lassous, I. G. (2008). Bandwidth estimation for IEEE 802.11-based ad hoc networks. *IEEE Transactions on Mobile Computing*, *7*(10), 1228–1241.
47. Yaling, Y., & Kravets, R. (2005). Contention-aware admission control for ad hoc networks. *IEEE Transactions on Mobile Computing*, *4*(4), 363–377.
48. Jia, Z., Gupta, R., Walrand, J., & Varaiya, P. (2005). Bandwidth guaranteed routing for ad-hoc networks with interference consideration. In *Proceedings of 10th IEEE symposium on computers and communications, 2005, ISCC 2005* (pp. 3–9).
49. Duarte, P. B., Fadlullah, Z. M., Vasilakos, A. V., & Kato, N. (2012). On the partially overlapped channel assignment on wireless mesh network backbone: A game theoretic approach. *IEEE Journal on Selected Areas in Communications*, *30*(1), 119–127.
50. Del Prado Pavon, J., & Sunghyun, C. (2003. 2003-01-01). Link adaptation strategy for ieee 802.11 wlan via received signal strength measurement. In *IEEE international conference on communications, 2003, ICC03* (Vol. 2, pp. 1108–1113).
51. Fan, Y., Li, J., Xu, K., Chen, H., Lu, X., Dai, Y., et al. (2013). Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, *21*(18), 20529–20543.
52. Senthilkumar, D., & Krishnan, A. (2010). Throughput analysis of IEEE 802.11 multirate WLANs with collision aware rate adaptation algorithm. *International Journal of Automation and Computing*, *7*(4), 571–577.
53. Gupta, R., & Walrand, J. (2004). Approximating maximal cliques in ad-hoc networks. In *15th IEEE International symposium on personal, indoor and mobile radio communications, 2004, PIMRC 2004* (Vol. 1, pp. 365–369).
54. Jiang, W., Jiang, W., Liu, S., Liu, S., Zhu, Y., & Zhu, Y., et al. (2007). Optimizing routing metrics for large-scale multi-radio mesh networks (pp. 1550–1553). IEEE.
55. Bron, C., & Kerbosch, J. (1973). Algorithm 457: Finding all cliques of an undirected graph. *Communications of the ACM*, *16*(9), 575–577.
56. Moon, J. W., & Moser, L. (1965). On cliques in graphs. *Israel Journal of Mathematics*, *3*(1), 23–28.
57. The network simulator ns-2. (2009). http://www.isi.edu/nsnam/ns.

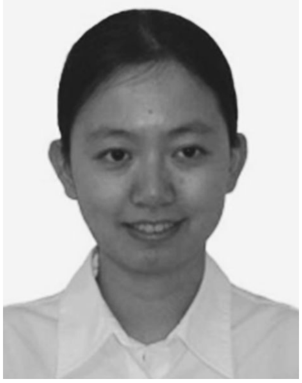


Xiaoheng Deng received the Ph.D. degrees in Computer Science Central South University, Changsha, Hunan, P.R. China, in 2005. Since 2006, he has been an Associate Professor and then a Full Professor with the Department of Electrical and Communication Engineering, Central South University. He is a senior member of CCF, a member of CCF Pervasive Computing Council, a member of IEEE and ACM. He has been a chair of CCF YOCSEF CHANGSHA from 2009 to 2010. His research interests include wireless communications and networking, congestion control for wired/wireless network, cross layer route design for wireless mesh network and ad hoc network, online social network analysis.

Lifang He is a M.Sc. student in School of Information Science and Engineering of Central South University, Changsha, China. She received the B.Sc. degrees in Electronic and Information Engineering from Kunming University of Science and Technology, Kunming, China, in 2013. Her major research interests are wireless mesh network and smart grid.

Qiang Liu CCF member, received the M.Sc. degrees in Information Science and Engineering of Central South University, Changsha, China, in 2014. His research interests include the routing protocol design and channel assignment of the wireless mesh network.

Xu Li CCF member, received his M.Sc. degrees in Information Science and Engineering of Central South University, Changsha, China, in 2014. His research interests include cross-layer design in routing metric and protocol of wireless mesh network and MAC layer.



Lin Cai (S'00–M'06–SM'10) received the M.A.Sc. and Ph.D. degrees in Electrical and Computer Engineering from the University of Waterloo, Waterloo, ON, Canada, in 2002 and 2005, respectively. Since 2005, she has been an Assistant Professor and then an Associate Professor with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada. She as been an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the EURASIP *Journal on Wireless Communications and Networking*, the *International Journal of Sensor Networks*, and the *Journal of Communications and Networks*. Her research interests include wireless communications and networking, with a focus on network protocol design and control strategy supporting emerging applications in ubiquitous networks.

Zhigang Chen He received B.Sc. the M.Sc. and Ph.D degrees from Central South University, China, 1984, 1987 and 1998, He is a Ph.D. Supervisor and his research interests are in network computing and distributed processing.