TÜBİTAK

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2013) 21: 470 – 492
© TÜBİTAK
doi:10.3906/elk-1104-25

End-to-end transmission time-based opportunistic routing protocols for bus networks

Luobei KUANG[1,*], **Zhijun WANG**[2], **Ming XU**[1], **Yingwen CHEN**[1]
[1]School of Computers, National University of Defense Technology, Changsha,
410073 Hunan, P.R. China
[2]Department of Computing, The Hong Kong Polytechnic University, Hong Kong

Received: 21.04.2011 • **Accepted:** 01.11.2011 • **Published Online:** 22.03.2013 • **Printed:** 22.04.2013

Abstract: Single path routing protocols for Internet access in bus networks (composed of traveling city public buses) may not achieve satisfactory performance because of frequent bus mobility. This paper studies the opportunistic routing protocols in bus networks to improve the system throughput and reduce the access delay. In this paper, we first propose an end-to-end transmission time (EET)-based opportunistic routing (OR) framework. We then derive 3 EET-based OR protocols, EETOR (EET-based OR), EETMcOR (EET-based OR with MAC contention consideration), and EETCcOR (EET-based OR with congestion control consideration), which consider 3 different EET metrics. EETOR considers the network layer transmission behavior to approximately estimate the EET without the knowledge of the MAC layer; EETMcOR calculates the EET by considering the MAC contention and builds a 3-dimensional Markov chain model to quantize the MAC behavior; and EETCcOR takes the congestion effect into account to evaluate the EET and hence decreases congestion by controlling the MAC layer transmission time. Simulations under a real city environment scenario with a bus mobility model are conducted to demonstrate the effectiveness of our OR protocols.

Key words: Bus networks, Internet access, opportunistic routing, end-to-end transmission time, congestion control

1. Introduction

A public bus is a type of a vehicle that people can use to get around a geographically compact and crowded city in a reasonable amount of time and at reasonable cost. Internet access in moving buses from openly available access points (APs) through Wi-Fi technology can provide Internet services at low cost and thus can greatly improve the quality of peoples' lives. It is considered to be the potential technology for vehicular communication systems [1–5]. Figure 1 shows a general bus network architecture. In the architecture, the cost-efficient high speed APs are placed in every bus station. The APs are connected to switching centers (SWCs), which are controlled by a public bus network controller (PBNC). A SWC controls the communication among different APs connected to it. The PBNC is a public bus network server in which the information of bus routes is placed. When a bus is ready to start a journey, it sends a message including the bus route number to the AP at the initial station, which in turn sends the message to the PBNC. The PBNC allocates an IP address for the bus, and then distributes the IP address, the bus route information, and the identities of APs along the route to the corresponding SWCs.

[*]Correspondence: silencejimmy@hotmail.com

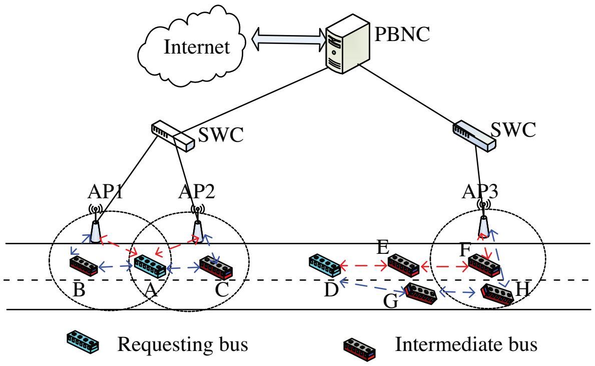


Figure 1. Internet access from moving buses.

In a bus network, a bus can retrieve data either directly from an AP or through relay from another bus. In the central areas of a city where bus density is high, the bus routes are seamlessly covered by APs. When a bus enters an intersection covered by 2 stations, it usually suffers from bad signal quality from both stations due to the long distance. In this case, a bus that directly connects to any AP may not have the best communication path to retrieve data. For example, as shown in Figure 1, suppose bus A is connecting to AP1 and moving to AP2. After bus A enters the area covered by both AP1 and AP2, it can keep the connection to AP1 or directly switch the connection to AP2. Both connections have low transmission rates due to the long distance. Bus A can also connect to AP1 through bus B or to AP2 through bus C. Due to better signal quality, the transmission rates between A and B (or C) and between B (or C) and the corresponding AP may be higher. Hence, the system throughput may be increased if bus A connects to an AP through a bus-to-bus path instead of using direct connections. In addition, when a bus (e.g., bus D in Figure 1) is moving in a suburban area that is not covered by any AP, the data from the bus to the AP have to be sent through a relay of other buses (e.g., bus$E$ and F or bus G and H). For this scenario, our previous work in [6] developed a single path routing protocol based on the bus lifetime (connection time between buses) to select the best communication path. However, the lifetime-based routing protocol may not be the best solution, because a longer lifetime path may have poor signal quality or high traffic load at the time. For example, bus D selects path $D-E-F$-AP3 as its data route due to its longer lifetime, but path $D-G-H$-AP3 may have higher end-to-end throughput. Hence, it is necessary to develop more efficient routing protocols.

To overcome the deficiency of the lifetime-based single path routing protocol, in this paper we propose an opportunistic routing (OR) protocol framework to improve the system throughput for bus networks. We first propose an end-to-end transmission time (EET)-based OR framework, which includes the metric calculation, the relay set determination, and the relay priority computation. We then propose 3 EET-based OR protocols: EET without media access control (MAC) consideration (denoted as EETOR), EET with MAC contention consideration (denoted as EETMcOR), and EET with congestion control consideration (denoted as EETCcOR). EETOR just considers the network layer transmission behaviors to calculate the EET rapidly without any knowledge of the MAC layer. In order to capture the end-to-end transmission time more accurately, EETMcOR is proposed, which takes MAC contention into account. By further considering the congestion control, EETCcOR is derived. EETCcOR can reduce the congestion by controlling the MAC layer transmission time. Finally, the simulation experiments conducted in a real city environment show that the proposed OR protocols can greatly improve the end-to-end communication performance. To the best of our knowledge, we are the first to study the EET-based OR protocol by considering the MAC contention and congestion control for bus networks.

The remainder of this paper is organized as follows. Section 2 presents the existing work on the routing protocols for providing Internet access in bus networks and OR in wireless networks. Section 3 introduces some preliminary work on rate adaptation in wireless networks. Section 4 proposes an EET-based OR framework, including metric calculation, relay set determination, relay priority computation, and implementation. In Section 5, 3 EET-based OR protocols are proposed based on 3 EET metrics. Section 6 evaluates the performance of the proposed protocols. Finally, Section 7 concludes the paper and outlines our future work.

2. Related work

Currently, dedicated routing protocols for bus networks are very few [6]. A vast amount of single path routing protocols suitable for ad hoc networks are used to support vehicular communications [7–11]. These protocols need either large communication control or other positioning systems to assist routing. Furthermore, they are also unsuitable for dynamic bus networks. Adaptive routing protocols [6,12–15] have also been proposed to consider the routing problem in which the data route lifetime is used for vehicles to select routes for communication. However, most of them are single path routing protocols and they do not consider bus mobility and transmission rate diversity. Moreover, they usually do not consider throughput guarantee. Routing metrics, like Hop Count [16] and ETX [17], guide routing ineffectively because of ignoring communication link reliability and path throughput. Although ETT-based routing [18] considers communication path throughput, the ETT-based single path routing protocol may still present low efficiency for bus networks in suburban environments with low bus density. It also does not consider the congestion control in high bus density environments, such as in city centers.

OR [19] utilizes the broadcast advantage of wireless communication to increase the reliability of a single transmission. ExOR [19] and GeRaF [20] are currently the most widely used OR protocols in wireless and vehicular networks. The former prioritizes a relay node for a node according to the expected number of transmission times between them. The latter selects nodes that are geographically closer to the destination to form a relay set. However, they cannot be viewed as efficient metrics to solve the problem for bus networks. In addition, neither OR protocol has considered the MAC contention and congestion control issues. Recently, ORCD [21] was proposed to prioritize relay nodes according to the number of queuing packets at each relay in order to address congestion. However, the ORCD solution ignores end-to-end throughput maximization. Hence, it may not be truly effective for bus networks. Although several OR protocols for vehicular networks have been proposed [22–25], they all mainly focus on packet forwarding. They suppose that all the vehicles along the source vehicle's movement direction are regarded as relays to forward packets, which may actually incur serious access collisions and low efficiency. Moreover, the protocols were developed for noninteractive services (e.g., traffic reports). They may not be suitable for Internet access, in which the communication requires longer transmission time.

3. Preliminaries

In wireless ad hoc networks, the transmission rate over a link can be adaptively selected according to the link signal quality. When the signal quality is high, a high channel rate can be used. Otherwise, a low rate should be used. This dynamic rate selection has been widely adopted by the IEEE 802.11 products. A higher channel rate often has a shorter maximum transmission distance because of its higher requirement for the signal-to-noise ratio (SNR) [26]. Therefore, when a wireless node receives a SNR that is larger than the SNR threshold for a certain channel rate, it can adopt this rate to communicate. Supposing that the noise level is a constant, a channel rate requests a minimum received signal power for the receiver to use the rate.

Based on the wireless radio propagation characteristics, the relationship between the signal strength and the transmission distance of 2 buses in a realistic scenario (which can be modeled as a shadowing model) can be expressed as [27]:

$$P_r = cd^{-\theta} \times 10^{\frac{\sigma_{dB} N^{-1}(\alpha)}{10}}, \tag{1}$$

where P_r is the received signal strength of a wireless node, c is a constant related to the antenna characteristics and the average channel attenuation, d is the distance between the sending and receiving nodes, $\theta$is the path loss exponent, σ_{dB} is a standard deviation of a shadowing model, β is a random variable that takes a value in [0, 1], and N^{-1} (•) is the reverse function of the normal distribution function.

A selected channel rate can uniquely determine the maximum distance between a sender and a receiver that can use the rate. If the distance is longer than the maximum distance, then they need to use a lower channel rate. In other words, a fixed distance can uniquely obtain the maximum channel rate to be used. For example, if channel rate R_k's maximum distance is d_k, then 2 wireless nodes whose distance is longer than d_k cannot use R_k (which must use a rate lower than R_k). Hence, for any distance d in the range $d_k < d \leq d_{k-1}$, the maximum available transmission rate is R_{k-1}. Supposing that the SNR threshold value for R_k is Th_{Rk} and the noise level is P_n, then the maximum distance for any given channel rate R_k can be evaluated as:

$$d_k = \log_{\theta} \frac{c \times 10^{\frac{\sigma_{dB} N^{-1}(\alpha)}{10}}}{P_n Th_{Rk}}. \tag{2}$$

From Eq. (2), we can determine the maximum distance for a transmission rate.

4. An EET-based OR framework

4.1. System model

We first consider a suburban environment where a bus may not be directly covered by any AP (e.g., bus D in Figure 1). When bus D needs to access the Internet, it can send a request to the neighboring buses (e.g., buses E and G, called the relay buses). The relay buses have their priorities to forward the request. After the relay buses receive the request, the one with the highest priority first transmits the request to the AP or the next hop bus. If it fails, the bus that has the second highest priority starts to relay the request. The process does not stop until the request is answered or no relay bus can forward the request.

To generalize the OR model in a bus network, we consider a network with N nodes (including several buses and an AP). Node n_i ($1 \leq i \leq N$) can transmit a packet at J different rates (R^1, R^2, ..., R^J). A link l_{ij}between n_i andn_j has reliability r_{ij}, which can be computed based on the link signal quality [6]. Link l_{ij} also has lifetime L_{ij} [6]. For simplicity, we ignore the high layer (above network layer) retransmission cost. The metric in our bus network OR model is an EET instead of a single-link expected transmission time. The EET is computed based on all the nodes in the relay set. EET denotes the delay time from a node to the destination node given all the relay nodes. We denote node n_i's EET by EET_i. Suppose that there are k relay nodes for node n_i (denoted as nr_{i1} to nr_{ik}), as shown in Figure 2a, and node nr_{ij} ($1 \leq j \leq k$) has probability r_{ij} for receiving a packet sent by node n_i.

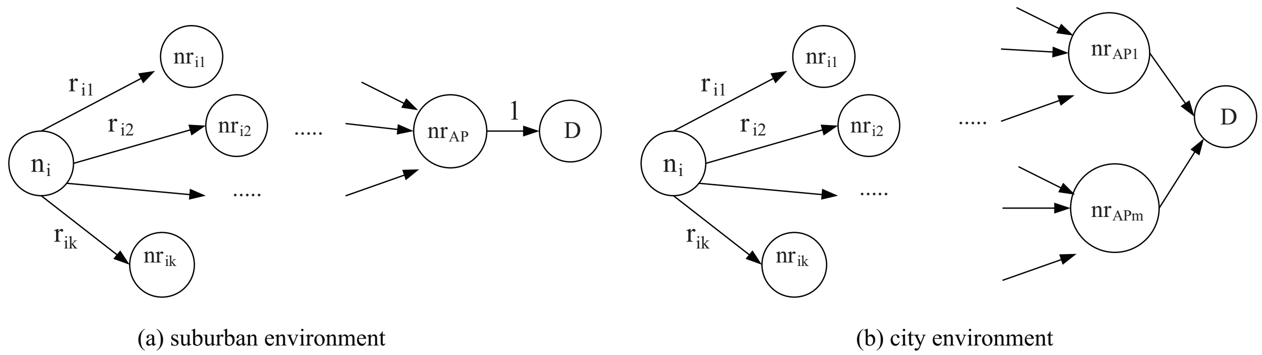


Figure 2. An example of the bus network OR model.

Supposing that the relay priority for the relay set of node n_i is arranged in a descending order of their priorities (e.g., $s =$ <1, 2, 3, ..., k>), then EET_i can be expressed as:

$$EET_i = \sum_{j=1}^{k} (N_P \times E[D_i] \times \frac{1}{r_{is_j}} + EET_{ij}) \times r_{is_j} \times \prod_{m=s_1}^{s_{j-1}} (1 - r_{im}), \tag{3}$$

where N_P is the number of the packets being transmitted at the time, $E[D_i]$ is the average transmission delay of node n_i, and s_i is the ith relay node in order s. For example, $s =$ <1, 2, 3, ..., k>, and then s_2 is the relay node nr_{i2}. $\frac{1}{r_{is_j}}$ denotes the expected number of transmissions. EET_{ij} is the EET of node nr_{ij} and can be computed in the same way, and it is regarded as 0 if nr_{ij} is an AP.

The OR model is constructed as follows. First, each node determines the relay nodes by computing the link lifetime and reliability based on its own moving speed, direction, and transmission time. The link lifetime and reliability can be calculated by using the methods proposed in our previous work [6]. Secondly, each node determines its relay nodes based on the link lifetimes and then computes its EET using its relay node's EETs and link reliabilities. Finally, each node obtains its relay priority. The model construction process completes until the source node (requesting bus) determines its EET and the relay priority of its relay nodes. Therefore, the main work of the EET-based OR model includes the relay set determination, $E[D]$ estimation, and relay priority computation.

In city environments, as in the scenario shown to the left of Figure 1, the OR model can be modeled as in the case shown in Figure 2b. The only difference from the suburban case is that there may be multiple APs to choose from and the number of relay hops is fewer. Hence, the OR model is the same as that of suburban environments.

4.2. Relay set determination

For a node n_i, the selection of relays should guarantee that no loops exist along any potential path through n_i. Moreover, the relay selection should consider the link lifetime, i.e. the link lifetime should be larger than the expected transmission time.

We further make additional 2 rules to decrease the relay set: 1) for the suburban environment, the relay nodes should be geographically located between the source node and the destination AP; 2) for the city environment, the relay nodes should be covered by a destination AP.

4.3. $E[D]$ estimation

$E[D]$ estimation is dependent on the transmission payload and rate and on network parameters such as access contention, traffic interference, and congestion effect. We have 3 heuristics for $E[D]$ estimation that are introduced in Section 5: 1) without MAC consideration; 2) with MAC contention consideration; and 3) with both MAC contention and congestion control considerations.

4.4. Relay priority computation

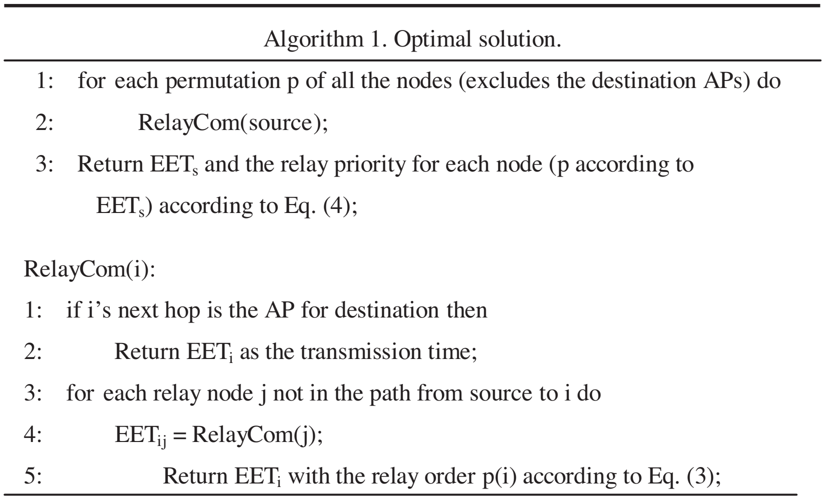
4.4.1. Optimal solution

The objective of the optimal solution is to find the relay priority at each node to make the EET of the source node minimal. Hence, we define the EET of the source node as:

$$EET_S = \min_{p\in P}[\sum_{j=1}^{k}(N_P \times E[D_S] \times \frac{1}{r_{ip(s)_j}} + EET_{ij}) \times r_{ip(s)_j} \times \prod_{m=p(s)_1}^{p(s)_{j-1}}(1-r_{im})], \tag{4}$$

where P is a permutation set and each element lists a permutation of all the relay nodes' priorities along the path. For example, p = (<1, 2, 3, ..., k_0>, <1, 2, 3, ..., k_1>, <2, 1, ..., k_2>, ..., <3, 1, 2, ..., k_n>). Each object at position i ($1 \le i \le n$) in the permutation p is a relay order of node n_i's relay nodes and is denoted as $p(i)$ (i.e. s in Eq. (3)). For example, $p(1)$ = <1, 2, 3, ..., k_1>denotes the relay order of node n_1's relay nodes, and $p(1)_2$ is the relay node nr_{12}. Hence, $p(0)$ is the relay order of the source's relay nodes and is also denoted as $p(s)$.

Based on the objective, we can construct an OR-tree rooted at the source node [28] and employ the optimal solution as Algorithm 1 to prioritize each relay at each relay node. The algorithm recursively guides each current node to compute its EET based on its downstream relay nodes. The process continues until the source obtains its EET. In this manner, all of the nodes' relay priorities can be computed.



Algorithm 1. Optimal solution.

```
for each permutation p of all the nodes (excludes the destination APs) do
    RelayCom(source);
Return $EET_s$ and the relay priority for each node (p according to
  $EET_s$) according to Eq. (4);

RelayCom(i):
if i's next hop is the AP for destination then
    Return $EET_i$ as the transmission time;
for each relay node j not in the path from source to i do
    $EET_{ij}$ = RelayCom(j);
        Return $EET_i$ with the relay order p(i) according to Eq. (3);
```

The complexity of the optimal solution is very high for its exhaustive search in the OR-tree at each node. For example, supposing there are n nodes and m layers in the OR-tree (excluding the destination APs), then the maximum number of computations for each node is about $(\lfloor \frac{n}{m} \rfloor !)^m$. It may be useful in a suburban scenario with a small number of possible relay nodes. However, it is not efficient when applied to a city environment.

4.4.2. Heuristic solutions

The potential complexity of the optimal solution is induced by the computation of the EET based on each permutation of the relay nodes of each node, which is different from a single metric for computing traditional OR relay priority. In this section, we propose 2 heuristic solutions for relay priority computation.

4.4.3. L-EET-based heuristic solution

The first heuristic EET solution computes the EET at each node locally, denoted as L-EET. When a node determines its EET and relay priority, it sends its EET to its upstream neighbors. The upstream neighbors then update their EETs and relay priorities and send them to their upstream neighbors. The process continues until the source node completes the computation. In the L-EET solution, the $L\text{-}EET_i$ of node n_i can be expressed as:

$$L - EET_i = \min_{s \in S} L - EET_i(s) = \min_{s \in S} [\sum_{j=1}^{k} (N_P \times E[D_i] \times \frac{1}{r_{is_j}} + L - EET_{ij}) \times r_{is_j} \times \prod_{m=s_1}^{s_{j-1}} (1 - r_{im})], \qquad (5)$$

where S is a permutation set, each element in S denotes a relay order among all the relay nodes of n_i, and s_j is the jth object in relay order s (i.e. the relay node nr_{ij}).

In the L-EET solution, for a relay node n_i, its relay priority depends on the relay priorities of its downstream relay nodes. The end-to-end relay priority from the source node to any destination AP is thus computed from the downstream relay nodes. Similarly, we can construct a sub-OR-tree rooted at each node (excluding the destination APs) and apply the heuristic solution (i.e. Algorithm 2) for each relay priority computation. In this way, each node can immediately obtain its EET and relay priority regardless of the computation of the source node.

Algorithm 2. L-EET-based heuristic solution.

```
Set L-EET of the AP for the destination as 0;
for each layer from bottom in the entire OR-tree do
    for each node $n_i$ at the layer do
    if $n_i$'s next hop is not the AP then
            Obtain L-EET of all its downstream neighbor nodes according to the sub-OR-tree;
        Construct an INT array RL[k] and initiate each element to 0;
        Set L-EET$_i$ to MAX_INTEGER;
        for each relay order s of the relay nodes do
            Compute L-EET$_i$(s) according to Eq. (5);
            if L-EET$_i$(s) <L- EET$_i$ then
                L-EET$_i$ = L-EET$_i$(s);
                RL[m] = $s_m$;
                Obtain L-EET$_i$ and its relay priority (i.e., RL);.
```

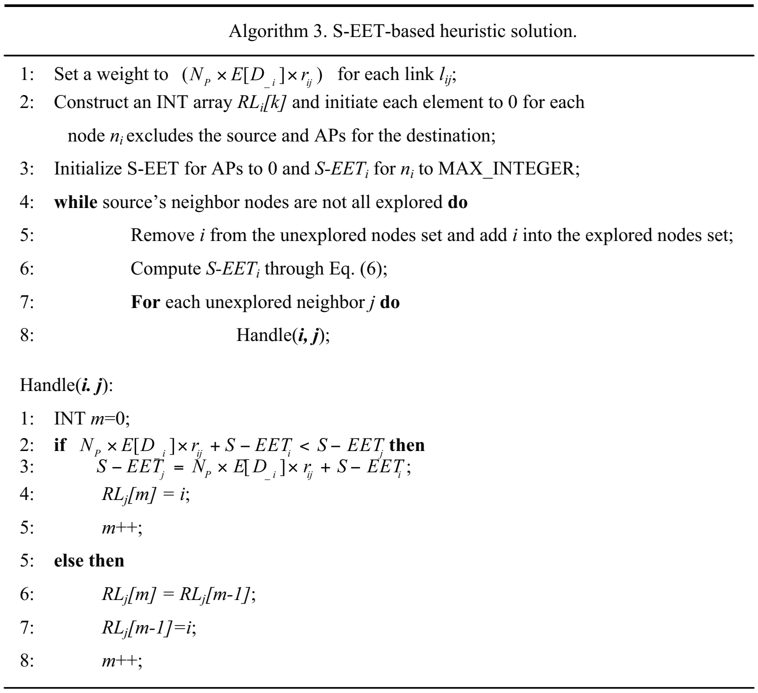
Supposing that there are n nodes and m layers in the OR-tree (excluding the APs), then the computation complexity for a node in L-EET is about ($\lfloor \frac{n}{m} \rfloor$!), which is significantly less than that of the optimal solution.

4.4.4. S-EET-based heuristic solution

The second heuristic EET solution is the single path EET (denoted as S-EET). Each node determines its relay priority by ordering each transmission delay between itself and its relay nodes. The minimum EET value is selected as its EET. The solution can further degrade the computation complexity.

$$S - EET_i = \min_{j=1}^{k}(N_P \times E[D_i] \times \frac{1}{r_{ij}} + S - EET_{ij}) \tag{6}$$

The S-EET of node n_i is calculated according to Eq. (6) and is presented in Algorithm 3. The algorithm is carried out from the destination APs. A node determines its relay priority similar to that in the Dijkstra shortest path algorithm. We set a weight to denote the link transmission time, $(N_P \times E[D_i] \times \frac{1}{r_{ij}})$, for each link l_{ij}. Initially, all the unexplored neighbors of the destination APs compute their S-EET. Each of their unexplored neighbors (destination AP's 2-hop neighbors) then updates their S-EET and their relay order. Its neighbors will then update their S-EETs accordingly and so on. Finally, we can obtain the relay priorities for all the relay nodes (i.e. RL_i for node n_i in Algorithm 3).



Algorithm 3. S-EET-based heuristic solution.

```
Set a weight to $(N_P \times E[D_{\_i}] \times r_{ij})$ for each link $l_{ij}$;
Construct an INT array $RL_i[k]$ and initiate each element to 0 for each
  node $n_i$ excludes the source and APs for the destination;
Initialize S-EET for APs to 0 and $S\text{-}EET_i$ for $n_i$ to MAX_INTEGER;
while source's neighbor nodes are not all explored do
        Remove $i$ from the unexplored nodes set and add $i$ into the explored nodes set;
        Compute $S\text{-}EET_i$ through Eq. (6);
        For each unexplored neighbor $j$ do
                Handle(i, j);

Handle(i. j):
INT $m$=0;
if $N_P \times E[D_{\_i}] \times r_{ij} + S - EET_i < S - EET_j$ then
    $S - EET_j = N_P \times E[D_{\_i}] \times r_{ij} + S - EET_i$;
    $RL_j[m] = i$;
    $m$++;
else then
    $RL_j[m] = RL_j[m-1]$;
    $RL_j[m-1]=i$;
    $m$++;
```

Although the S-EET-based solution cannot obtain both the global and local EETs for the OR model, the solution can be proven to have close performance compared to the optimal solution for a small network, as shown in Section 7. Obviously, the complexity of the solution is O(n), which is reduced substantially.

4.5. Implementation

Although the optimal solution demonstrates high computational complexity and may have high control overhead in its implementation, it may still be useful in the suburban environments. Therefore, this section proposes a distributed implementation for all the optimal and heuristic solutions.

The optimal solution can be implemented as follows. 1) Each node computes its link reliability and lifetime and then determines its relay nodes based on the link lifetime. Each node also computes the transmission time of each link between itself and its downstream neighbor nodes. 2) Each node broadcasts a HELLO message to other nodes. The message includes its location, moving speed and direction (speed is 0 for an AP), the transmission time, and the reliability of each link between itself and its relay nodes. The HELLO message of the source node also includes the payload size. 3) Each node constructs an OR-tree by exhaustively searching all the paths from the source to the destination and finding the optimal relay priorities of each node. 4) Upon receiving a packet, a node applies the general OR process [19] to forward the packet. To capture the dynamic topology, the source node maintains a timer with a duration equal to the minimum link lifetime in the OR tree. After timeout, the source node reinitiates the OR construction process.

The L-EET-based heuristic solution is implemented similarly. 1) Each node determines the set of relay nodes as that in the optimal solution. 2) Each node constructs a sub-OR-tree rooted at itself to compute its L-EET as well as its relay priority. 3) Each node broadcasts its L-EET value and the minimum link lifetime between itself and the relay nodes to its one-hop neighbors. 4) Upon receiving the broadcast message, the upstream nodes construct a sub-OR-tree rooted at itself to calculate its L-EET value and the relay priority. The process repeats until the source completes the calculation of the relay priority. 5) Finally, each node transmits packets as the general OR process. Similar to the optimal solution, the source node also maintains a timer to reinitiate the OR tree construction process.

In the distributed implementation of the S-EET-based heuristic solution, each node determines its relay priority locally based on a single path. First, the source node broadcasts its payload size to the destination APs. The destination APs broadcast their S-EETs to initiate a relay priority computation process. Each node then calculates its S-EET value, records the relay priorities of its downstream relay nodes, and broadcasts its S-EET to its upstream neighbor nodes. The implementation is executed like a route discovery process in the AODV or DSR routing protocols. The subsequent steps and the timer set for the source node are the same as those in the other 2 solutions.

5. OR protocols

As mentioned above, 3 OR protocols below are proposed based on 3 different heuristics for $E[D]$ estimation, which is used to further calculate the EET metric. We introduce each OR protocol and the corresponding $E[D]$ estimation heuristic in the following sections.

5.1. EETOR

5.1.1. E[D] estimation considering network layer transmission

The most widely used method to estimate the average transmission delay $E[D]$ between 2 communicating nodes is EET [18]. EET can conveniently derive $E[D]$ by considering only the packet transmission at the network layer transmission. However, the method is proposed based on the fixed link bandwidth, which is not applicable for the bus networks. In bus networks, the actual transmission rate is impacted by the realistic communication scenario and changes as the distance between the 2 nodes changes (as in Eq. (2)). Therefore, we improve the EET to derive our first heuristic.

Similar to EET, EETOR considers the packet transmission behavior at the network layer to quickly calculate the $E[D_i]$ of node n_i without any knowledge of the MAC layer. $E[D_i]$ in EETOR is relevant to the transmission payload and the actual transmission rate. Thus, the average transmission time $E[D_i]$ can simply be expressed as:

$$E[D_i] = \frac{E[P]}{R_i}, \tag{7}$$

where $E[P]$ is the packet size and R_i is the maximum achievable transmission rate, which is determined by the distance between n_i and its furthest relay node, as in Eq. (2).

5.1.2. EETOR

In EETOR, relay set determination and relay priority computation are the same as those in the general EET-based OR model. The OR model with optimal relay priority computation is just denoted as EETOR, and 2 OR models based on heuristic solutions are denoted as L-EETOR and S-EETOR, respectively.

5.2. EETMcOR

5.2.1. E[D] estimation considering MAC contention

The average transmission delay heuristic in EETOR is based on the network layer behavior. In fact, the existence of the MAC contention also greatly affects the transmission delay. The MAC layer contention time (backoff time, backoff frozen time, transmission time, etc.) therefore becomes the main constituent of $E[D]$. In this section, we derive the second heuristic by considering the MAC layer behaviors. Thus, the OR based on this is named EETMcOR.

At the MAC layer, a bus usually accesses the channel to transmit its packets through broadcast by applying the 802.11 DCF scheme. The average transmission delay for the general WLAN can be derived by constructing a Markov chain model for a node's backoff behavior [29]. Bus networks own more dynamic characteristics than WLANs. Therefore, EETMcOR combines all the dynamic behaviors to evaluate transmission delay $E[D]$ through a similar probabilistic computation.

We combine the mobility, transmission rate diversity, and backoff behavior of a bus to construct a 3-dimensional Markov chain model. First, we divide the roads into several zones. With this zone-based system model, a bus moving along an AP can be viewed as crossing zones. The mobility of buses can be represented by zone transition using a Markov chain model. In the model, each state represents one spatial zone. We then analyze the probabilities of each transmission rate for a bus in a zone, and the transmission rate diversity of a bus in a zone further can be similarly represented by a Markov chain model. Each state also denotes a transmission rate; furthermore, the backoff time of a bus varies as the channel usage changes at a zone. The broadcast backoff behavior can also be represented by a Markov chain model [29]. With this 3-dimensional Markov chain model, we present the dynamic communication process for a bus.

Through solving the Markov chain model as in the Appendix, the average transmission delay $E[D_z]$ in Zone z is obtained as:

$$E[D_z] = \frac{W}{2}[(1-p_{b,z})\sigma + (p_{b,z} + (1-p_{b,z})(\frac{\lambda l_z}{l} - 1)\frac{W}{(W-1)^2})((L_H + E[P])/E[R] + DIFS + \delta)]. \tag{8}$$

The average system delay $E[D]$ is:

$$E[D] = \frac{\sum\limits_{z=0}^{n} E[D_z]}{n}, \tag{9}$$

where W is the maximum backoff window size; $p_{b,z}$ is the conditional probability of a busy channel given that the node in Zone z, σ, be the backoff slot time duration unit in DCF; λ is the average bus density; l_z is the length of Zone z; L_H is the packet header size; $E[P]$ is the packet size; $\delta$is the propagation delay; DIFS is the time period for a distributed interframe space; and $E[R_z]$ is the average transmission rate in Zone z. The detailed computation of these parameters can be found in the Appendix.

5.2.2. EETMcOR

We denote the OR model with optimal relay priority computation as EETMcOR and the 2 OR models based on heuristic solutions as L-EETMcOR and S-EETMcOR.

The distributed implementations of the above 3 protocols are similar to those in Section 4.4.3. The differences lie in the initial estimation of the critical point of each node and in the transmission time computation, which should consider the $E[D]$ calculation at the MAC layer.

5.3. EETCcOR

5.3.1. E[D] estimation considering MAC congestion control

EETOR or EETMcOR is a minimizing-delay routing protocol. They both usually incur heavy traffic at the relay node with the highest priority. To decrease the congestion incurred by EETMcOR, we further propose the third heuristic to compute average transmission delay $E[D]$. Thus, the corresponding adaptive OR based on this heuristic is called EETCcOR. With this heuristic, EETCcOR protocol can also achieve load balancing for all buses.

To capture the congestion in the routing, we improve the $E[D]$ computation in EETMcOR by considering the number of queuing packets in each node's buffer. The end-to-end transmission time at node n_i can be expressed as:

$$E[D] = N_B \times E[D]_{MAC}, \tag{10}$$

where N_B is the number of the packets to be transmitted in node n_i's buffer. $E[D]_{MAC}$ is the packet transmission time with MAC contention consideration (i.e. $E[D]$ in EETMcOR).

5.3.2. EETCcOR

Similar to EETMcOR, the 3 OR models based on the optimal and 2 heuristic relay priority computations are called EETCcOR, L-EETCcOR, and S-EETCcOR, respectively.

In the distributed implementation, each node must include its current number of packets in the buffer to be transmitted in the periodic HELLO messages. The rest of the steps are the same as those in EETMcOR.

6. Performance evaluation

This section evaluates the performance of our proposed OR protocols by comparing them to the other related OR protocols through simulations.

6.1. Bus mobility model

The bus mobility model used to evaluate the performance of the proposed protocols is set to the same model as that in our previous work [6].

For a bus, its velocity V depends on its current position, as in the following:

$$V = \begin{cases} V_0, & (X \leq X_{Fixed} - D_-) \\ V_0\sqrt{\frac{X_{Fixed}-X}{D_-}}, & (X_{Fixed} - D_- < X \leq X_{Fixed}) \\ 0, & (X = X_{Fixed)}) \\ \frac{V_{1\sqrt{D_+(X-X_{Fixed}}}}{D_+}, & (X_{Fixed} \leq X X_{Fixed} + D_+) \\ V_1, & (X \geq X_{Fixed} + D_+) \end{cases} \tag{11}$$

where X_{Fixed} denotes the position of a nearby AP or traffic light. X denotes the position of a bus. X_{Fixed}can be obtained from the SWC and X can be gotten through the positioning system installed in the bus. t is the moving time and V denotes the velocity at time t. t_- and t_+ denote the experienced moving time in an acceleration and deceleration period, respectively. D_- andD_+ are the deceleration and the acceleration distance, respectively. V_0 and V_1 are 2 velocities when buses are moving stably before and after getting close to an AP.

6.2. Simulation experiments

The fact that the optimal solutions require a very high computational cost makes the NS-2 simulator very slow. Hence, we first conduct an OR model construction to get the relay set, relevant link parameters, and relay priorities in our customized simulator. We then evaluate its performance under NS-2 with these results as input data. The rest of the simulations are conducted in NS-2. Simulation experiments include 3 groups, as follows:

(1) To verify the effectiveness of our heuristic solutions, we first compare the optimal solution with the heuristic solutions for 3 OR models in delay performance, i.e. EETOR vs. L-EETOR and S-EETOR, EETMcOR vs. L-EETMcOR and S-EETMcOR, and EETCcOR vs. L-EETCcOR and S-EETCcOR. We also compare the control overhead among the optimal solutions, the L-EET-based heuristic solutions, and the S-EET-based solutions.

(2) To investigate the performance of the proposed OR protocols, we compare their throughput and delay with the 3 optimal solutions (i.e. EETOR, EETMcOR, and EETCcOR) and the traditional OR protocols, ExOR [19] and GeRaF [20]. Furthermore, we employ the metric ETT [18] instead of ETX in ExOR to obtain another OR protocol and call it ETTOR. The only difference between ETTOR and ExOR lies in the metric.

(3) To illustrate the effectiveness of our OR protocol in congestion control, we compare EETCcOR with ORCD [21] in delay performance.

Simulations are conducted with the real street map of the Afton Oaks area of Houston, Texas, a 1900 m × 1900 m area. The map data were obtained from the publicly available TIGER (Topologically Integrated Geographic Encoding and Referencing) [30] database from the US Census Bureau, which gives detailed street maps for the entire United States. We set up the traffic lights and APs randomly in the intersections and in the middle of roads of the map. The bus mobility described in Section 6.1 is used to model the bus movement in the map. We set 8 different bus lines, and most of them have a portion of common roads. The interarrival time of each bus is fixed at some values and the bus number is generated randomly. The simulator will always generate buses until the simulation time expires. When a bus reaches its destination, it moves back along the route towards its starting station. Each bus maintains a Poisson arrival process for its data packets. We just set the number of data sources as 20% of the total bus number to decrease serious collisions. Some referred parameters are listed in Table 1. We run each algorithm 100 times and use the average value of the results to evaluate the performance.

Table 1. Simulation parameters.

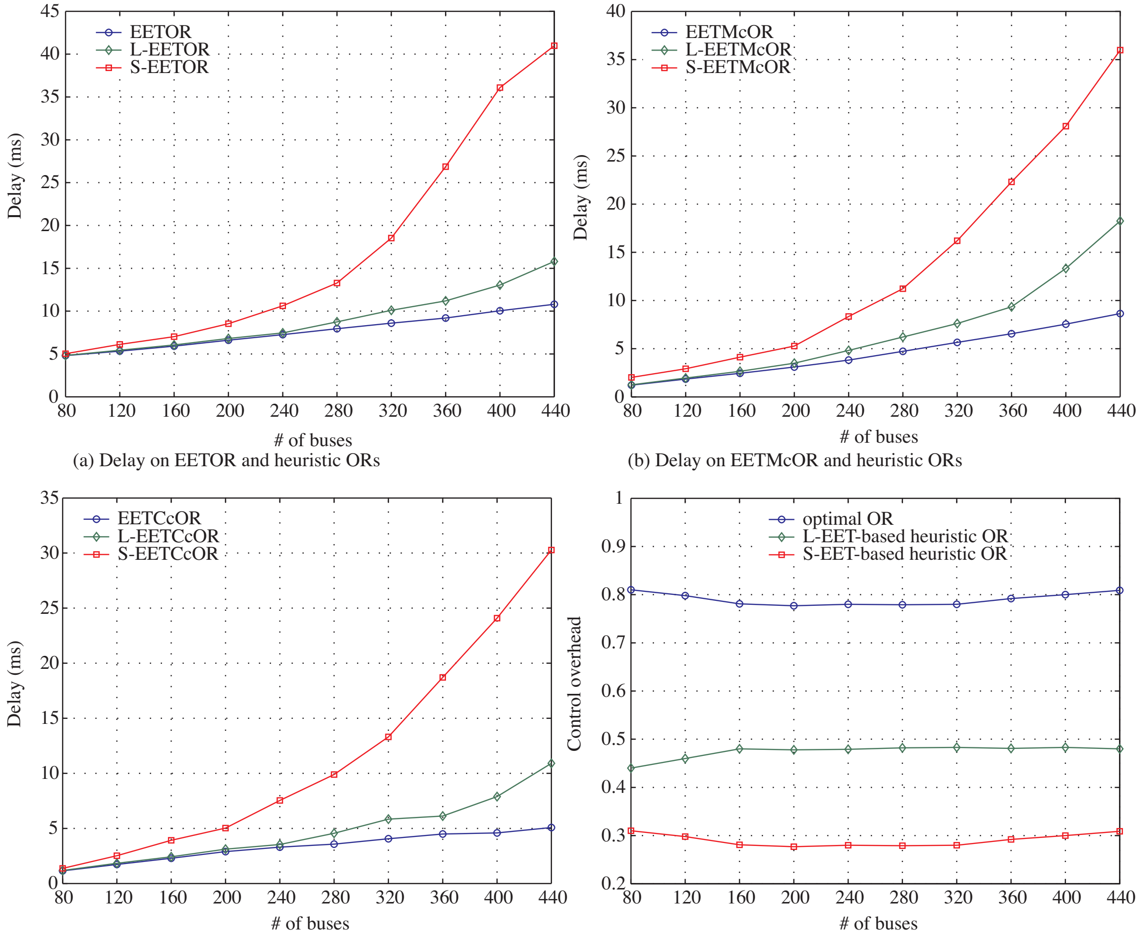
Maximum transmission range	250 m
Propagation model	Shadowing model
Shadowing model standard deviation	4.0
Path loss exponent	1.8
Link data rate	6 Mbps, 11 Mbps, 18 Mbps, 27 Mbps, 54 Mbps
Data packet type	UDP
Data packet interarrival time	2 s
Data packet size	500 bytes
Bus velocity	[20 km/h, 40 km/h]
Bus interarrival time	5 s
Decel./accel. distance	20 m
Road cell length	5 m
MAC header	272 bits
PHY header	128 bits
Propagation delay	1 μ s
DIFS	50 μ s
Slot time	10 μ s
Maximum backoff window size	128

Three performance measurement metrics are used and are defined as follows: *throughput*, the average number of successfully received bytes per second between each requesting bus and its corresponding AP; *delay*, the average time between the generation of a data packet and the delivery to the final recipient of the packet for each end-to-end communication pair; and *control overhead*, the ratio between the number of control packets sent and the total number of packets sent.

First, we verify the effectiveness of our heuristic solutions compared to their corresponding optimal solutions. Figure 3 demonstrates the performance in terms of the delay and the control overhead. From Figures 3a–3c, we can see that the L-EET-based ORs can provide performances that are close to their optimal solutions for relay selection and priority computation. This is due to the fact that the L-EET-based ORs select nodes that contribute most to the improvement of the EET. Figure 3d shows that the L-EET-based solutions (i.e. L-EETOR, L-EETMcOR, and L-EETCcOR) can achieve much smaller control overhead. Note that the control overhead in the 3 cases is the same because the implementation of sending control packets is the same. Furthermore, due to the considerations of the MAC contention and the congestion in EETCcOR, its delay is shorter than that in EETOR. Although the performance of the S-EET-based ORs is much worse when the number of buses is very large, it may still be useful in the case of small networks.

6.2.1. Optimal ORs vs. other ORs

The performance comparisons between the proposed OR protocols and the ExOR, GeRaF, and ETTOR are shown in Figure 4. Here the average bus velocity is fixed at 30 km/h (i.e. [20 km/h, 40 km/h]). In EETOR, buses are supposed to use the transmission rate when they request to communicate with others along their movement. From Figure 4, we can clearly see that the proposed solutions are better in both throughput and delay. ExOR captures the local link transmission characteristics to prioritize relays, but it cannot improve the end-to-end performance. GeRaF focuses on the distance to the destination AP for relay nodes in order to decrease the number of transmission hops, but it omits the considerations of the link quality. The performance of the 2 OR protocols therefore is worse than that of our proposed OR protocols. Especially as the number of nodes increases, although ExOR has more choices, it usually has more hop counts, which results in lower reliability and longer delay. GeRaF usually has higher path reliability due to its low hop count and hence has better performance than that of ExOR in small bus networks. As EETOR considers communication path throughput improvement, it has better performance than ExOR and GeRaF. However, ExOR, GeRaF, and ETTOR do not consider bus mobility and transmission rate diversity to reflect the MAC access contention and congestion behavior, and hence they perform poorly.



(a) Delay on EETOR and heuristic ORs

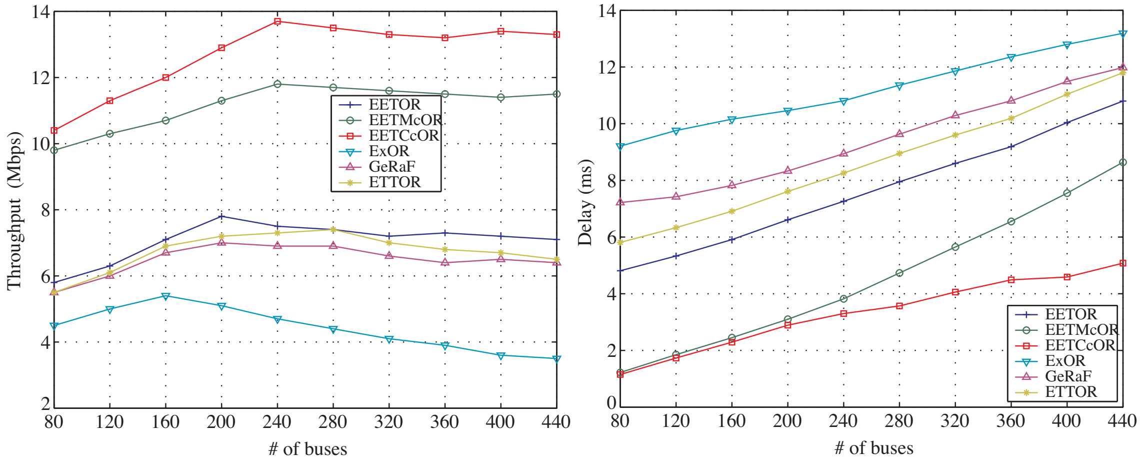
(b) Delay on EETMcOR and heuristic ORs

(c) Delay on EETCcOR and heuristic ORs

(d) Control overhead on all solutions

Figure 3. Performance comparison among the optimal solution and the heuristic solutions.

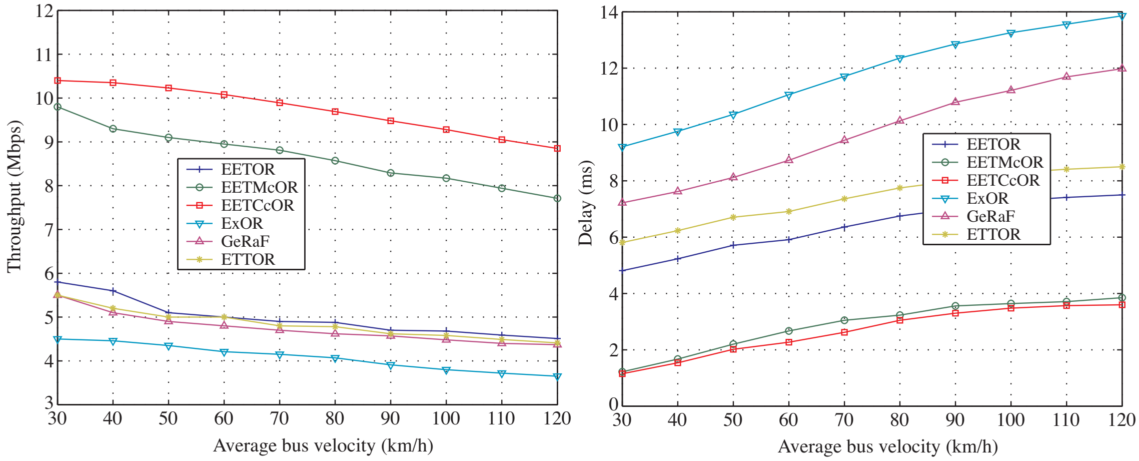
Figure 5 plots the performance of the 6 OR protocols under various average bus velocities from 30 km/h to 120 km/h. We can see that the performance becomes dramatically worse as the average bus velocity increases. The performance of all protocols is influenced greatly by the enhanced mobility. The dynamic links make the relay set determination and relay priority of ExOR, GeRaF, ETTOR, and EETOR unstable and further make them inefficient. Even in this case, we can still conclude that our protocols, especially EETMcOR and EETCcOR, are more adaptive to bus networks.



(a) Throughput comparison among several ORs

(b) Delay comparison among several ORs

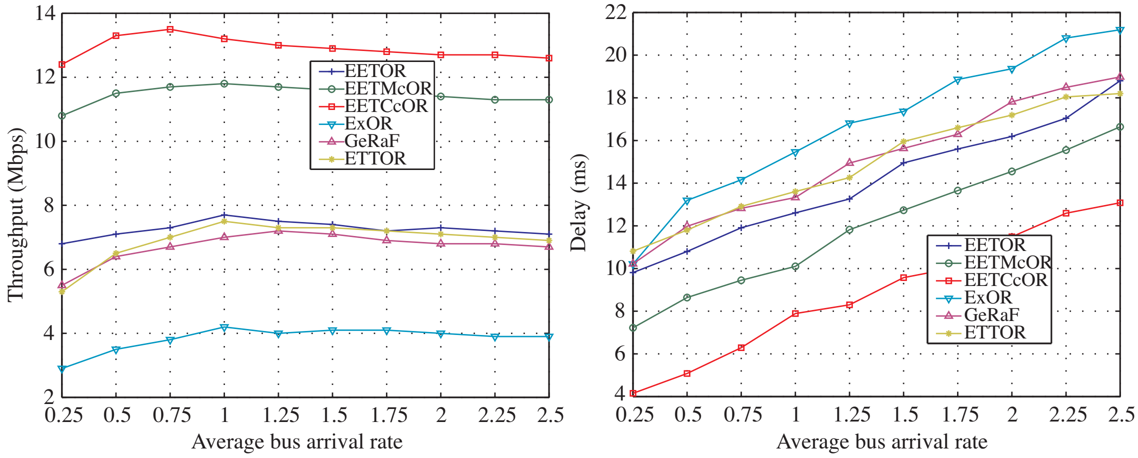
Figure 4. Performance under fixed average bus velocity.



(a) Throughput comparison among several ORs

(b) Delay comparison among several ORs

Figure 5. Performance under varied average bus velocity.

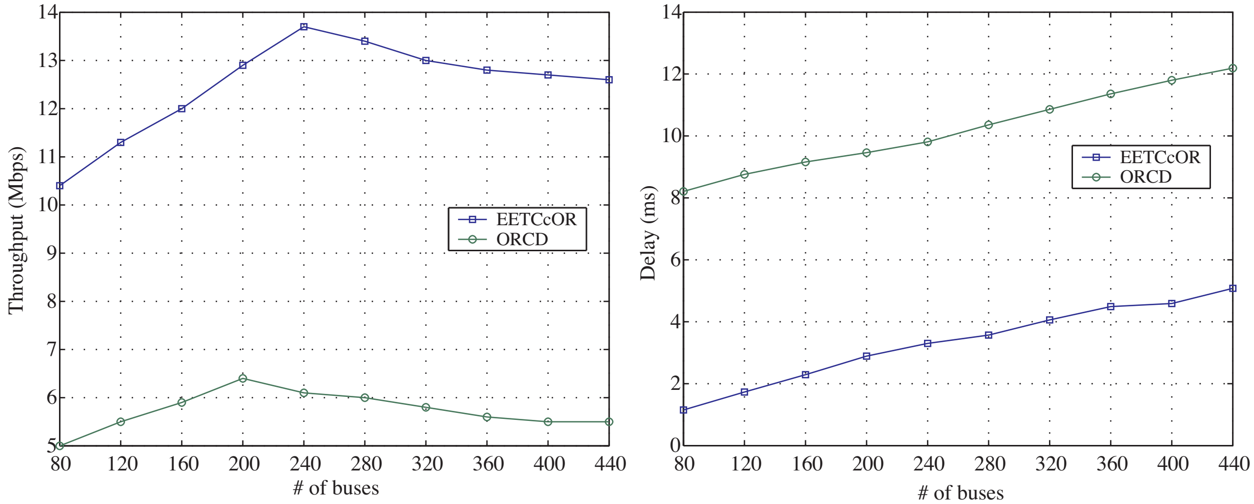


(a) Throughput comparison among several ORs

(b) Delay comparison among several ORs

Figure 6. Performance under varied average bus arrival rate.

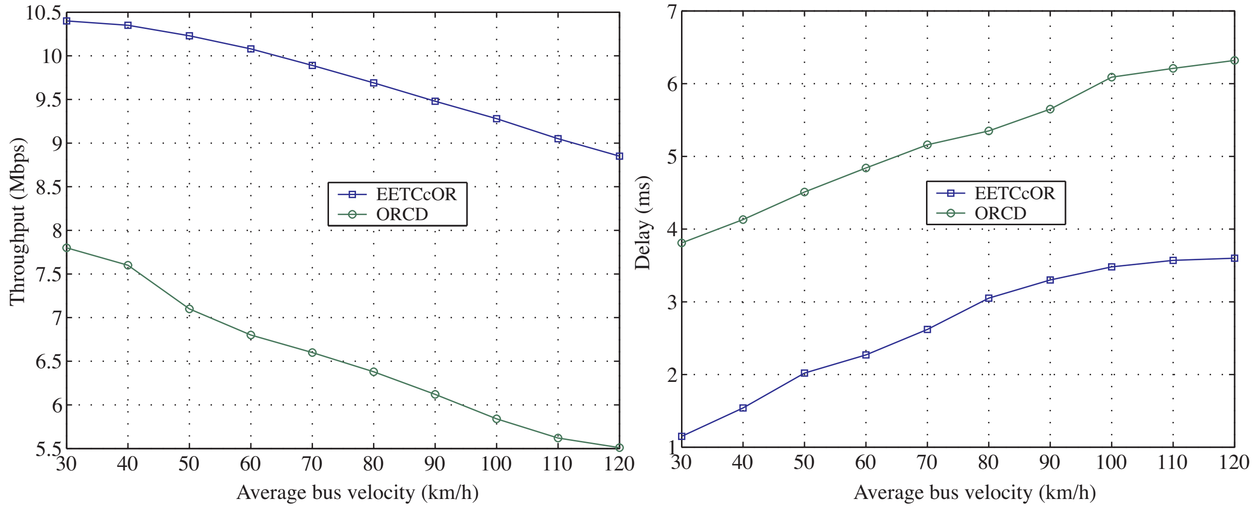
Figure 6 investigates the performance of the 6 OR protocols under various average bus arrival rates. As the average bus arrival rate increases, MAC contention becomes more serious. ExOR, GeRaF, ETTOR, and EETOR demonstrate worse performance since they have not considered the MAC layer behavior of a bus. However, EETMcOR and EETCcOR can still perform much better.



(a) Throughput comparison

(b) Throughput comparison

Figure 7. Performance comparison between EETCcOR and ORCD under fixed average bus velocity.

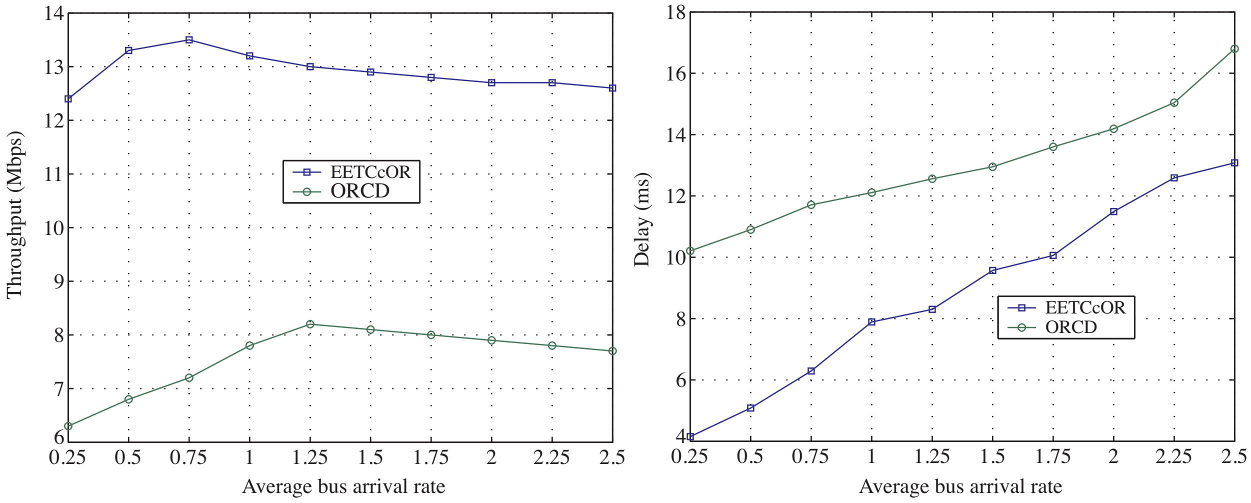


(a) Throughput comparison

(b) Delay comparison

Figure 8. Performance comparison between EETCcOR and ORCD under varied average bus velocity.

To capture the good congestion control ability of our proposed OR protocol, we compare the EETCcOR to ORCD under fixed average bus velocity, different average bus velocity, and different average bus arrival rate. The results are shown in Figures 7–9. Our proposed OR metric reflects the congestion from the result (more delay) to balance the relay priority. Hence, it is more effective than ORCD, which only considers the number of packets being queued at each node.



(a) Throughput comparison

(b) Delay comparison

Figure 9. Performance comparison between EETCcOR and ORCD under varied average bus arrival rate.

7. Conclusion

A bus network is a kind of special wireless network that has its own characteristics of mobility and transmission rate diversity. How to select a communication path to maximize the system throughput and minimize the access delay is important in bus networks. Single path routing protocols generally cannot address this problem efficiently. Therefore, this paper investigated the EET-based OR for bus networks to improve the end-to-end performance. First, we proposed an EET-based OR framework. The details of EET calculation methodology, relay set determination, relay priority computation, and distributed implementation were presented. Among them, the work of relay priority was mainly investigated. To capture a more accurate EET, EETMcOR computes the end-to-end transmission time through a 3-dimensional Markov chain model, which captures the transmission time at the MAC layer. EETMcOR considers the bus mobility and transmission rate diversity, which most existing OR protocols have not investigated. Based on these, EETCcOR further obtains the EET with congestion consideration. EETCcOR can decrease congestion by controlling the MAC layer transmission time rather than scheduling the number of packets being queued at each node, and hence it is more effective in the bus networks. The simulation experiments showed that our proposed OR protocols outperform the existing ones.

In our future work, we will focus on the routing problem with the consideration of multiradio multichannel communication in bus networks, which incurs more interference to buses and will bring new challenges to relay selection.

Acknowledgment

This study was supported by the National Science Foundation of China under Grant No. 60773017 and No. 61003304.

Appendix

Calculation of the average transmission delay $E[D]$ in EETMcOR

- Zone-based system model

We first consider the scenario for the buses moving along the roads, as shown in Figure 10. A road is presented as a line of cells, each with length l (usually longer than the bus length and smaller than twice the bus length). Each cell can either be empty or be occupied by a bus. The coordinate of cell i is denoted as $x_i = i \times l$. Suppose the average bus density is λ (number of buses/cell). The probability of a bus to exist in a cell is denoted by $p = \lambda$.

Suppose the maximum communication range of all the buses is the same, denoted by CR_m. When a bus moves toward a destination AP, initially it cannot communicate with the AP (out of the communication range of the AP), or it communicates with a very low transmission rate (located at the border of the communication range to the AP). Hence, the bus will select a relay path to forward its access request to an AP. After the bus moves to some point closer to the AP, it can directly communicate to the AP at a higher transmission rate. We mark this special point as the *critical point* (e.g., the points A and $--A$ in Figure 10). Based on the different access patterns during the movement, we divide the road into spatial zones, as shown in Figure 10. When the critical points are determined (can be figured out as in [31]), we can easily get the position $B(-B)$, which is CR_m away from $A(-A)$. We can then get the similar position CR_m away from the previous point. The process continues to the interzone including the requesting bus. We define Zone 0 as the zone the requesting bus is located in and Zone i as the zone between 2 consecutive points (e.g., Zone 1 is determined by $--B$ and $--A$in Figure 10).

With zone division, a bus moving along an AP can be viewed as crossing from Zone 0 to Zone n (i.e. the maximum number of zones for a bus in an environment is n+1). When the bus moves close to the next AP, it enters Zone 0 again. Therefore, as buses traverse the consecutive APs along the road, they are regarded as transiting iteratively among the zones. The mobility of buses can be represented by the zone transition using a Markov chain model.

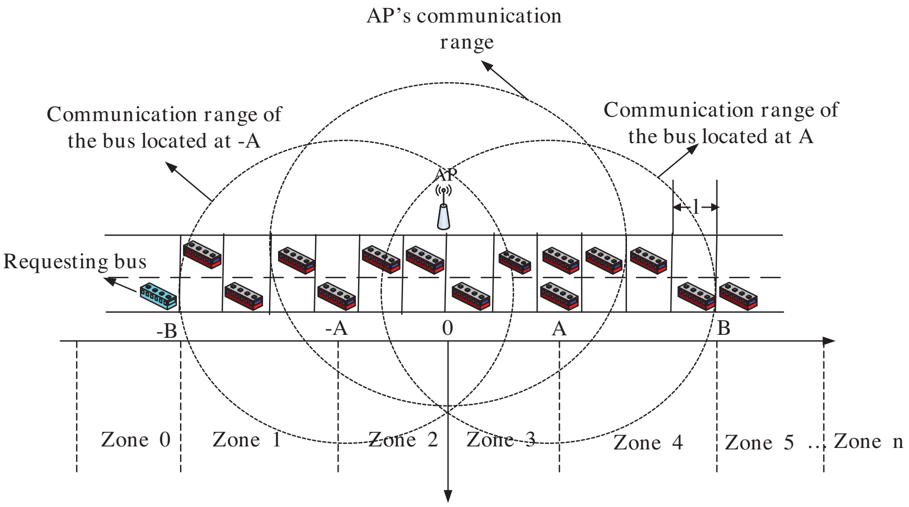


Figure 10. System model for moving buses.

In the zone-based model, a requesting bus in Zone i first selects a relay bus in Zone $i+1$, then the relay bus selects a relay bus in Zone $i+2$, and so on. The buses in the zone nearest to the destination AP are the candidates of the last relay nodes. A bus determines its transmission rate based on the furthest bus as mentioned in Section 3. To increase the system throughput, the bus is assumed to use the maximum achievable rate for a given distance d. The transmission rate can be dynamically changed as the distance changes. Suppose a bus has J different transmission rates R^0, R^1, $\ldots$, R^{J-1} in ascending order, and the transmission rate of a bus entering a zone is set to be the minimum rate R^0. The rate is also set to R^0 if the furthest bus is missed. The rate may then switch to one of the other *J-1* rates according to the value of d. As d changes, the bus increases or decreases the rate to its neighboring one.

At any specific zone, a requesting bus usually accesses the channel to transmit its packets through broadcast by applying the DCF scheme. The backoff time varies as the channel usage changes. For simplicity, the transmission rate switching time is ignored, and the transmission status can be inherited from the previous zone when a bus passes into a new zone.

- MAC access model

To evaluate the DCF broadcast performance of an individual bus, we examine a random requesting bus and represent its status by a discrete 3-dimensional Markov chain $\{Z\ (t), R(t),\ B(t)\}$at time slot t. $Z(t)$ (takes values in the range (0, 1, ..., n)) denotes the spatial zone that the node is currently in; $R(t)$ (takes values in the range (0,1, $\ldots$, *J-1*), stands for (R^0, R^1, $\ldots$, R^{J-1})) denotes the transmission rate at the current time slot; and $B(t)$ (takes values in the range (0,1,$\ldots$, W-1), W is the maximum number of backoff slots) denotes the backoff time of the requesting bus at the current time slot. Slot times $t$and $t+1$ correspond to the beginning of 2 consecutive statuses of a node. The state transition diagram of the 3-dimensional Markov chain is sketched in Figure 11.

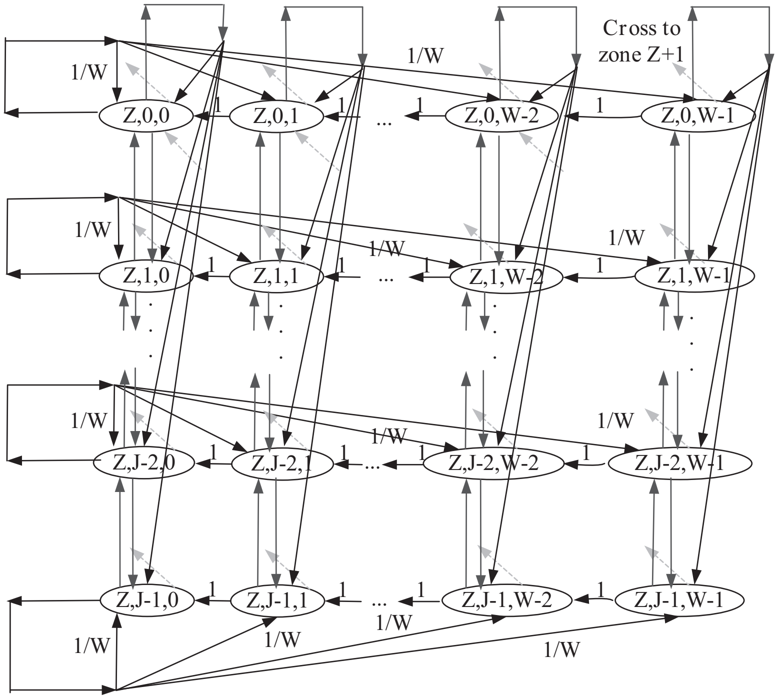


Figure 11. Three-dimensional Markov chain for bus broadcast at Zone Z.

Our model is distinct from the model in [29] by considering the bus mobility and transmission rate diversity in 3 aspects. First, after the deduction of the backoff time, the requesting node either stays in the current zone or moves to the next zone with renewed transmission probabilities. Second, a node will switch to the lowest transmission rate when it passes to the next zone. Third, the backoff values smoothly change independent of the transmission rate and zone switch. The transmission history will be inherited in the new zone.

Given t_z , $z \in$Z, the one-step nonnull transition probabilities of the Markov chain from time slot t to $t+1$ are:

$$\left\{\begin{array}{lll}
P(z,r,b|z,r,b+1) = (1-\frac{E[T_{slot}]}{t_z})(1-\frac{1}{2}\times\frac{V_{Max}-V_{\min}}{d_r-d_{r+1}}E[T_{slot}]), & z\in[0,n], r\in[0,J), b\in[0,W-1), & (a)\\
P(z,0,b|z-1,r,b+1) = \frac{E[T_{slot}]}{t_{z-1}}, & z\in[1,n], r\in[0,J), b\in[0,W-1), & (b)\\
P(z,r,b|z,r,0) = \frac{p_{s,z}}{W}(1-\frac{E[T_{s,z}]}{t_z}), & z\in[0,n], r\in[0,J), b\in[0,W), & (c)\\
P(z,0,b|z-1,r,0) = \frac{p_{s,z}}{W}\frac{E[T_{s,z-1}]}{t_{z-1}}, & z\in[1,n], r\in[0,J), b\in[0,W), & (d)\\
P(z,r+1,b|z,r,b) = (1-\frac{E[T_{slot}]}{t_z})(\frac{1}{4}\times\frac{V_{Max}-V_{\min}}{d_r-d_{r+1}}E[T_{slot}]), & z\in[0,n], r\in[0,J-1), b\in[0,W), & (e)\\
P(z,r-1,b|z,r,b) = (1-\frac{E[T_{slot}]}{t_z})(\frac{1}{4}\times\frac{V_{Max}-V_{\min}}{d_r-d_{r+1}}E[T_{slot}]), & z\in[0,n], r\in[1,J), b\in[0,W), & (f)\\
P(z,J-1,b|z,J-1,b) = (1-\frac{E[T_{slot}]}{t_z})(1-\frac{1}{4}\times\frac{V_{Max}-V_{\min}}{d_{J-2}-d_{J-1}}E[T_{slot}]), & z\in[0,n], b\in[0,W), & (g)\\
P(z,r,b|z,0,b) = (1-\frac{E[T_{slot}]}{t_z})[\frac{1}{CR_m}(d_r-d_{r+1})], & z\in[0,n], r\in[0,J), b\in[0,W), & (h)
\end{array}\right. , \quad (12)$$

where $p_{s,z}$ is the probability that a node transmits in a slot time; $E[T_{slot}]$ is the mean duration of one time slot; and $E[T_{s,_z}]$ is the mean time of one successful transmission for the nodes in Zone z. In Eq. (12), the meaning of each subequation is similar to that in [31].

Let $\pi_{z,r,b} = \lim_{t\to\infty} P\{Z(t)=z, R(t)=r, B(t)=b\}$ be the steady state probability of the Markov chain. By solving the Markov model, we can express each steady state probability by $p_{s,z}$, $E[T_{slot}]$ and $E[T_{s,z}]$. We then compute the related parameters, $E[T_{slot}]$ and $E[T_{s,z}]$, as follows.

Let σ be the backoff slot time duration unit in DCF and a packet P with an average packet length $E[P]$. The packet header includes both the physical and the MAC layer headers: $L_H = PHY_{hdr} + MAC_{hdr}$. Let δ be the propagation delay and DIFS be the time period for a distributed interframe space. The average transmission rate is denoted by $E[R_z]$, and $E[R_z] = \sum_{i=0}^{J-1} R^i \cdot (\sum_{b=0}^{W-1} \pi_{z,i,b})$. We then have the following.

(1)

$$E[T_{s,z}] = (L_H + E[P])/E[R_z] + DIFS + \delta \quad (13)$$

(2) $p_{s,z}$:

Let $p_{t,z}$ denote the conditional transmission probability given that the node in Zone z. We then have the following.

$$p_{t,z} = \frac{\sum_{r\in[0,J)} \pi_{z,r,0}}{l_z/\sum_{m\in Z} l_m} \quad (14)$$

The conditional probability of a busy channel given that the node in Zone z can be derived as:

$$p_{b,z} = 1-(1-p_{t,z})^{\lambda l_z/l-1}, \quad (15)$$

where λl_z denotes the number of nodes in Zone z, including the request node. $p_{s,z}$ can be derived as:

$$p_{s,z} = (\lambda l_z/l-1)p_{t,z}(1-p_{t,z})^{\lambda l_z/l-1}. \quad (16)$$

(3) $E[T_{slot}]$:

The average duration of one time slot includes the unit backoff time σ and average frozen duration of the backoff time $E[T_{frozen}]$.

Let $\tau_z(i)$ $(i = 1,2,...)$ be the probability that a station transmits i number of packets consecutively in a virtual slot. We then have $\tau_z(i) = \frac{p_{t,z}}{W^i}$. Therefore, the probability $p_{b,z}(i)$ that the channel is busy in the ith frozen stage is $p_{b,z}(i) = 1 - (1 - \tau_z(i))^{\lambda l_z/l-1}$. The average frozen stage $E[N_{frozen}]$ is:

$$E[N_{frozen}] = \sum_{i=1}^{\infty} i \times p_{b,z}(i) = \sum_{i=1}^{\infty} i \times (1 - (1 - \frac{p_{t,z}}{W^i})^{\lambda l_z/l-1}) \approx p_{t,z}(\frac{\lambda l_z}{l} - 1)\frac{W}{(W-1)^2}. \quad (17)$$

Now $E[T_{slot}]$ can be expressed as:

$$E[T_{slot}] = \sigma + E[T_{frozen}] = \sigma + E[N_{frozen}](p_{s,z} \times E[T_{s,z}] + (p_{b,z} - p_{s,z})E[T_{c,z}]), \quad (18)$$

where $E[T_{c,z}]$ denotes the average time of one transmission collision for the node in Zone z and can be derived as $E[T_{c,z}] = (\mathrm{L}_H + E[P]) / E[R_z] + \mathrm{DIFS} + \delta$.

- $E[D]$ computation

The average saturation delay $E[D_z]$ in Zone z is given by the time for the delayed packet of each successful transmission. It can be evaluated by multiplying the average number of virtual slots used for each successful transmission by the average length of a virtual slot time. That is:

$$E[D_z] = E[X][(1 - p_{b,z})E[T_{slot}] + p_{s,z} \times E[T_{s,z}] + (p_{b,z} - p_{s,z})E[T_{c,z}]], \quad (19)$$

where X denotes the random variable representing the number of virtual slots for a successful transmission. As we know, the backoff time for each packet is selected in the range $\{0, 2, \ldots, W-1\}$. Therefore, $E[X] =$ *W/2*.

The average system saturation delay $E[D]$ is:

$$E[D] = \frac{\sum\limits_{z=0}^{n} E[D_z]}{n}. \quad (20)$$

References

[1] J. Ott, D. Kutscher, “Drive-thru Internet: IEEE 802.11b for ‘automobile’ users”, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 1, pp. 362–373, 2004.

[2] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, S. Madden, “CarTel: a distributed mobile sensor computing system”, Proceedings of ACM SenSys, pp. 125–138, 2006.

[3] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, S. Madden, “A measurement study of vehicular Internet access using in situ Wi-Fi networks”, Proceedings of ACM MOBICOM, pp. 50–61, 2006.

[4] BBC UK, “City-wide Wi-Fi rolls out in UK”, available at http://news.bbc.co.uk/2/hi/technology/4578114.stm.

[5] M. Grebb, “Cities unleash free Wi-Fi”, available at http://www.wired.com/gadgets/wireless/news/2005/10/68999.

[6] L. Kuang, M. Xu, Z. Wang, “An adaptive routing protocol for bus networks”, Proceedings of IEEE AINA, pp. 98–104, 2010.

[7] C.E. Perkins, P. Bhagwat, “Highly dynamic destination-sequenced distance-vector (DSDV) routing for mobile computers”, Proceedings of the Conference on Communications Architectures, Protocols and Applications, pp. 234–244, 1994.

[8] M. Musolesi, C. Mascolo, “CAR: Context-aware adaptive routing for delay tolerant mobile networks”, IEEE Transactions on Mobile Computing, Vol. 8, pp. 246–260, 2009.

[9] C. Lochert, M. Mauve, H. Füßler, H. Hartenstein, “Geographic routing in city scenarios”, Proceedings of ACM MOBICOM, pp. 69–72, 2005.

[10] V. Naumov, R. Baumann, T. Gross, “An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces”, Proceedings of ACM MobiHoc, pp. 108–119, 2006.

[11] V. Naumov, T.R. Gross, “Connectivity-aware routing (CAR) in vehicular ad hoc networks”, Proceedings of IEEE INFOCOM, pp. 1919–1927, 2007.

[12] H. Menouar, M. Lenardi, F. Filali, “Movement prediction-based routing (MOPR) concept for position-based routing in vehicular networks”, in Proceedings of IEEE Vehicular Technology Conference, pp. 2101–2105, 2007.

[13] W. Shen, J. Tang, R. Wolff, “Reliable routing for roadside to vehicle communications in rural areas”, in Proceedings of IEEE ICC, pp. 3017–3021, 2008.

[14] C. Hung, H. Chan, E. Wu, “Mobility pattern aware routing for heterogeneous vehicular networks”, in Proceedings of IEEE WCNC, pp. 2200–2205, 2008.

[15] V. Namboodiri, L. Gao, “Prediction-based routing for vehicular ad hoc networks”, IEEE Transactions on Vehicular Technology, Vol. 56, pp. 2332–2345, 2007.

[16] Y. Yang, J. Wang, R. Kravets, “Interference-aware load balancing for multihop wireless networks”, Technical Report UIUCDCS-R-2005-2526, Department of Computer Science, University of Illinois at Urbana-Champaign, 2005.

[17] D.S.J. De Couto, D. Aguayo, J. Bicket, R. Morris, “A high-throughput path metric for multi-hop wireless routing,” Proceedings of ACM MOBICOM, pp, 134–146, 2003.

[18] R. Draves, J. Padhye, B. Zill, “Routing in multi-radio, multi-hop wireless mesh networks”, Proceedings of ACM MOBICOM, pp. 114–128, 2004.

[19] S. Biswas R. Morris, “ExOR: Opportunistic multi-hop routing for wireless networks”, Proceedings of ACM SIGCOMM, pp. 133–144, 2005.

[20] M. Zorzi R. Rao, “Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance”, IEEE Transactions on Mobile Computing, Vol. 2, pp. 349–365, 2003.

[21] M. Naghshvar, T. Javidi, “Opportunistic routing with congestion diversity in wireless multi-hop networks”, Proceedings of IEEE INFOCOM, pp. 1–5, 2010.

[22] I. Leontiadis, C. Mascolo, “GeOpps: Geographical opportunistic routing for vehicular networks”, Proceedings of IEEE WoWMoM, pp. 1–6, 2007.

[23] B. Xu, A. Ouksel, O. Wolfso, “Opportunistic resource exchange in inter-vehicle ad-hoc networks”, Proceedings of IEEE MDM, pp. 4–12, 2004.

[24] J. LeBrun, C.N. Chuah, D. Ghosal, M. Zhang. “Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks”, Proceedings of IEEE VTC, pp. 2289–2293, 2005.

[25] I. Leontiadis, P. Costa, C. Mascolo, “Extending access point connectivity through opportunistic routing in vehicular networks”, Proceedings of IEEE INFOCOM, pp. 486–490, 2010.

[26] Cisco Systems Inc., Cisco Aironet 802.11a/b/g Wireless LAN Client Adapters (CB21AG and PI21AG) Installation and Configuration Guide, 2004.

[27] T.S. Rappaport, Wireless Communications: Principles and Practice, 2nd ed., New Jersey, Prentice Hall, 2002.

[28] M. Lu, J. Wu, "Opportunistic routing algebra and its applications", Proceedings of IEEE INFOCOM, pp. 2374–2382, 2009.

[29] X. Ma, X. Chen, "Performance analysis of IEEE 802.11 broadcast scheme in ad hoc wireless LANs", IEEE Transactions on Vehicular Technology, Vol. 57, pp. 3757–3768, 2008.

[30] US Census Bureau, TIGER, TIGER/Line and TIGER-Related Products, available at http://www.census.gov/geo/www/tiger/.

[31] L. Kuang, Z. Wang, M. Xu, Y. Chen, "Performance study of IEEE 802.11 broadcast protocols in vehicular networks", Information Technology Journal, Vol. 10, pp. 2060–2067, 2011.