HEAT: Scalable Routing in Wireless Mesh Networks Using Temperature Fields

Rainer Baumann*, Simon Heimlicher*, Vincent Lenders†, Martin May*
*Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
†Department of Electrical Engineering
Princeton University, NJ, USA

Abstract— **Existing *unicast* routing protocols like for example AODV are not suited well for wireless mesh networks as in such networks, most traffic flows between a large number of mobile nodes and a few access points with Internet connectivity. In this paper, we propose HEAT, an *anycast* routing protocol for this type of communication that is designed to scale to the network size and to be robust to node mobility. HEAT relies on a temperature field to route data packets towards the Internet gateways, as follows. Every node is assigned a temperature value, and packets are routed along increasing temperature values until they reach any of the Internet gateways, which are modeled as heat sources. Our major contribution is a distributed protocol to establish such temperature fields. The distinguishing feature of our protocol is that it does not require flooding of control messages. Rather, every node in the network determines its temperature considering only the temperature of its direct neighbors, which renders our protocol particularly scalable to the network size. We analyze our approach and compare its performance with AODV and OLSR through simulations with Glomosim. We use realistic mobility patterns extracted from geographical data of large Swiss cities. Our results clearly show the benefit of HEAT versus AODV and OLSR in terms of scalability to the number of nodes and robustness to node mobility. The packet delivery ratio with HEAT is more than two times higher than with AODV or OLSR in large mobile scenarios and we conclude that HEAT is a suitable routing protocol for city-wide wireless mesh networks.**

I. INTRODUCTION

The penetration of IEEE 802.11 (WLAN) in homes and offices around the world has been tremendous. Many home users deploy WLAN access points in their homes and connect to the Internet over broadband cable modems or DSL lines. Because these WLAN networks are often idle, they could potentially be used to offer Internet access to mobile users in densely populated areas or large cities. Our envisioned scenario is a city-wide wireless mesh network consisting of fixed access points with Internet connectivity and mobile nodes that are typically pedestrians or people moving in a vehicle and carrying a mobile device. The mobile nodes in this mesh network serve as relays, forwarding traffic for other mobile nodes and thus maintaining network-wide Internet connectivity. This approach has the advantage that only a small set of Internet gateways are necessary to provide a "city-wide" Internet coverage when the density of the mobile users is large enough.

The particular problem we address in this paper is how to route data packets from the mesh network to the Internet. The routing includes traffic from the mobile nodes to any access point (called Internet *gateway* in the remainder of

this paper) and response traffic back to the mobile nodes. This routing problem is different from what MANET routing protocols like AODV [1], OLSR [2], DSR [3], DSDV [4], or ZRP [5] were originally designed for. These protocols were designed to provide end-to-end paths between two communication hosts (unicast-type of communication). However, in our work, routing is from *any* mobile node in the network to *any* Internet gateway (anycast). In principle, unicast routing protocols can be applied, however, they scale poorly in terms of communication overhead since they establish an individual path per node-gateway pair.

Our primary contribution is HEAT, an anycast routing protocol for wireless mesh networks. As its name suggests, HEAT is inspired by the heat conduction in physics. That is, we model the gateways as heat sources which create a temperature field in the network. The higher the temperature of a node, the closer it is to an access point. Using these fields, packet forwarding is fairly simple: packets are forwarded along the nodes with the highest temperature until they eventually reach any heat source (an Internet gateway). Our protocol to establish temperature fields in the network is purely based on local information. It means that every node calculates its own temperature by only evaluating the temperature of its direct neighbors. This makes our protocol particularly scalable since no flooding of messages is required.

We compare using simulations with Glomosim the performance of HEAT with AODV and OLSR. AODV is a representative reactive unicast MANET routing protocol. OLSR is a popular proactive protocol. Our simulations are based on a detailed mobility pattern extracted from geographical data of large Swiss cities. Our simulations show that HEAT scales much better than AODV and better then OLSR to the network size, the node density, the number of gateways, and the node mobility. For example in large or/and dense networks HEAT outperforms AODV and OLSR by more then a factor of two in terms of successful packet delivery. Furthermore, the packet delivery ratio of HEAT remains above 95% even for large and dense networks with nodes moving at typical pedestrian speeds.

The rest of this paper is organized as follows. In the next Section, we highlight related work. In Section III, we explain the concept and implementation of HEAT. We present our evaluation methodology in Section IV that we use in Section V. Finally, we conclude the paper in Section VI.

II. RELATED WORK

Routing in wireless networks has undergone extensive study. In this Section, we discuss related work that has influenced the design of HEAT and we point out what we have done differently.

A. MANET and Mesh Routing Protocols

A multitude of unicast routing protocols for MANETs have been proposed (e.g. [1]–[5] to name just a few). These protocols have been extended with gateway discovery functionality [6]–[12] to allow their use for wireless mesh networks. Since these protocols provide unicast routes, individual routes must be maintained between every mobile node and one of the gateways. Therefore, these protocols scale poorly to the number of nodes in the mesh network.

In [13], scalability to the number of mesh nodes is improved with the use of location information; however, this kind of information is typically not available in our target scenario where the mobile nodes in the mesh network are commodity laptops or hand-held devices.

A problem that is common to most MANET and mesh routing protocols is that gateway announcements or gateway requests are prone to vanish due to route breaks, and the recovery procedure is often as expensive as establishing a new route. In contrast, [14] proposes an efficient mechanism to fix broken routes locally. We also go into this direction. With HEAT, all control messages are local and routes can easily be repaired locally. Moreover, since routing is based on scalar potential fields and not on route entries, alternative routes can most of the time be determined without additional overhead when the primary route fails.

Mosko et al. [15], propose to establish multiple non-disjoint paths for better performance, but again the established routes are unicast and this protocol is not scalable to the number of mesh nodes.

The authors of [16] propose a single-hop mesh network where mobile clients connect directly to the gateways. Compared to our approach, where mobile users also relay packets on behalf of their neighbors, this requires a much higher mesh node density for a comparable wireless coverage.

B. Anycast Routing

Anycast routing was first proposed for IP networks in [17]. Different protocol implementations followed for wireless ad hoc networks [18]–[20]. However, these anycast routing protocols are all based on *unicast* routing techniques such as link state or distance vector routing, and as a consequence they all inherit the same scalability problems of these protocols. Anycast in general scales poorly to the number of groups since IP anycast addresses can not be aggregated into subnets. To this end, different approaches were proposed [21]–[23] to provide scalable anycast routing in the Internet. In mesh networks however, one anycast group representing the gateways to the Internet is typically sufficient and scalability to the number of groups is not a major concern.

We proposed a model for anycast routing based on potential fields in [24]. This paper goes a step further and introduces a scalable protocol to establish potential fields using only local information.

C. Field-based Routing

Field-based or gradient-based routing has been proposed in the past for various type of applications including routing in MANETs [24], [25], load balancing in the Internet [26], data collection in sensor networks [27], [28], sensor node placement [29], guided navigation [30], or service discovery in MANETs [31]. The basic forwarding principle of HEAT which consists of forwarding along the steepest gradient is similar to these works. However, our method to establish and maintain potential fields after link breaks is unique and uses only local information from the direct neighbors by mimicking how heat dissipates.

III. CONCEPT AND IMPLEMENTATION OF THE HEAT PROTOCOL

We propose to apply anycast to provide scalable routing in wireless mesh networks. In this Section, we first present our general anycast routing concept based on temperature fields. Then, we describe our protocol implementation.

A. Concept of Routing using Temperature Fields

Over the past few years, routing using potential fields has been proposed in various contexts [24], [26], [32]. These schemes all share the same design idea: the construction of a scalar field on the network which assigns a scalar value to every node in the network. The destinations are represented as maximum scalar values and packets are always forwarded along the steepest gradient towards the destination. While being fairly simple, the concept of field-based routing provides a very versatile way of determining routing decisions. For instance, modeling shortest-path routing with a field-based scheme is straightforward and has been demonstrated in [24]. Owing to the fundamental properties of fields, loop freedom of routes is ensured, and it is guaranteed that packets are forwarded towards the destination as long as they are no local maxima in the field. The major advantages of field-based routing are the robustness and simplicity. By design, a field comprises multiple routes to a destination, thus if the link to the neighbor with the highest field intensity breaks a successor can easily be determined.

In this work, we propose *HEAT*, a novel method to establish the scalar field for field-based routing. *HEAT* has two distinguishing features. Firstly, it considers both the *length* and the *robustness* of paths in the routing decision. Secondly, the field construction and maintenance mechanism of *HEAT* scales to the number of nodes and the number of gateways since it only requires communication among *neighboring* nodes.

The *HEAT* algorithm is a fully distributed, proactive anycast routing algorithm. It is inspired by the properties of temperature fields, as discussed in the next subsection. In brief, our algorithm assigns a temperature value to every node in the mesh network. New nodes are assigned a value of zero; gateway nodes are assigned the maximum value. In contrast to strict shortest-path routing, *HEAT* determines the temperature value of a node based on (i) its distances to available gateways but also based on (ii) the robustness of the paths towards these gateways. That is, a path providing multiple alternative delivery opportunities along its way is preferred to a path over which packets cannot be naturally re-routed to alternative nodes towards the destination. An illustrative example is depicted in Fig. 1. The part of the
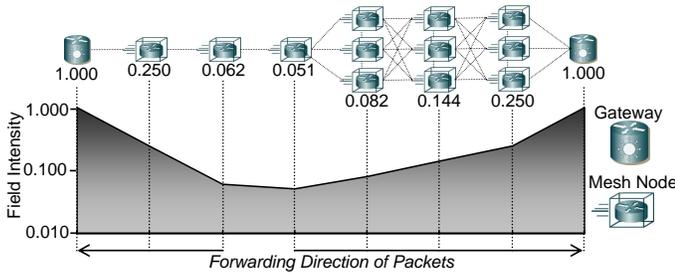
Fig. 1. Example of a temperature field with a conductivity value of $\kappa = 1/4$ and areas of different link redundancies. The packets of the node with temperature value 0.051 are forwarded to the right, across an area of high redundancy instead of to the closest gateway at the left.

network leading to the gateway on the right hand side has more links than the left part of the network leading to the gateway on the left. As a result, the temperature gradient towards the gateway on the right is steeper and packets are routed in this direction, even if the network distance to the left gateway is shorter (measured in the number of hops).

Nodes calculate their temperature based solely on the temperature values of their neighbors, which they learn through periodically broadcast messages. Data packets are then routed along the steepest gradient and finally reach a gateway.

B. Analogy to Temperature Fields

As mentioned, *HEAT* is inspired from temperature fields in physics. A temperature field assigns a single scalar value to every particle in space. The values of the temperature field are higher in the vicinity of heat sources and decrease with the distance to the source.

In a solid, heat is transferred by conduction. On a microscopic scale, conduction presents itself as hot, rapidly moving or vibrating atoms and molecules. By interactions among neighboring atoms and molecules, heat is transferred. The physical parameter *thermal conductivity, κ* indicates its ability to conduct heat. The conduction of heat is governed by *Fourier's Law*. In essence, this law demands that the temperature of the field always decrease away from sources, resulting in a temperature gradient whose maxima are at the source.

In order to map the properties of temperature fields to a given network topology, we consider nodes in the mesh network as particles and gateways as heat sources. In [24], Lenders et al. show that under the assumption that there are no local maxima in the field, following the path defined by the steepest gradient always leads to a gateway; and that there are no loops in this path. However, not all policies for assigning scalars to nodes guarantee that there are no local maxima in the potential field. In our approach, we avoid local maxima by adhering to the following policy: For every node, only neighbors with a higher temperature may contribute to the own temperature. This policy guarantees monotonicity of the field.

C. HEAT Protocol

We describe in this subsection our protocol for routing using temperature fields. According to the concept described before, the gateways supply a temperature field that enables routing from the mesh nodes to the Internet gateways. The temperature values of neighbors, which are required by *HEAT* for constructing the temperature field, are periodically exchanged between neighboring mesh nodes by *HEAT beacon* messages. Based on these messages, every mesh node calculates its own temperature using the same function.

Once the field is constructed, routing packets from the mesh nodes to the gateways is straightforward and implemented on a hop-by-hop basis: A packet is always forwarded to the neighbor with the highest temperature, resulting in steepest-gradient routing. Routing of packets back from the gateways to the mesh nodes is achieved by recording the paths that packets have taken when following the steepest gradient and using the reverse path.

1) Temperature Field Construction and Maintenance: As mentioned before, the sources of the temperature field are the gateways. Therefore, each gateway sets its temperature value to a maximum value. Then, every node (including the gateway nodes) broadcasts its temperature value to its neighbors periodically at a given *HEAT beacon* time interval. Based on these messages, all nodes build and maintain a data structure called *neighbor table,* which contains an entry for every known neighbor. Neighbor entries comprise the address, the last reported temperature, and a timestamp value of the corresponding node. Whenever an entry is added, removed, or changed, the temperature value is re-computed. In essence, we have to differentiate among three cases:

- **New neighbor.** If a beacon from an unknown neighbor is received, a corresponding entry is added to the neighbor table. In addition, the temperature value is re-computed.
- **Maintain neighbor**. If the reported temperature value of a known neighbor changes, the node re-calculates its temperature value.
- **Missing neighbor.** If no beacon is received from a neighbor for a certain period, its entry is removed and the temperature value is re-computed.

The key idea of *HEAT* is to provide scalability (with regard to protocol overhead) and robustness (with regard to link and node failures). Due to the local message exchanges, our method scales with the number of neighbors per node (scalability and convergence of the implementation are evaluated in Section V). Robustness is achieved by assigning the temperature values such that routes through network areas with high redundancy (in terms of node and link redundancy) are preferred. The more neighbors with high temperatures, the higher is the temperature of a given node. The detailed algorithm is described in Alg. 1. The algorithm calculates the temperature t_{final} of a node as follows: In a first step, the node sorts its neighbors based on their temperatures $\theta_i, i \in \{0, ..., n\}$ in ascending order (line 1) into an array a. Then, it iterates over a accumulating the temperature of the next neighbor to the sum of the temperatures of the previous neighbors t_j until the temperature of the next neighbor is less than the accumulated temperature (line 4). In each step j, the value t_{j+1} is calculated as follows (line 5): The difference between the temperature of the currently considered neighbor, denoted by $a[j]$, and the temperature accumulated so far, t_j, is calculated. Then, this difference is multiplied by the conductivity parameter κ, and the result is added to the temperature accumulated so far, denoted by t_j.

As a result, nodes which have many neighbors towards the gateways obtain higher temperatures than links with only a small number of neighbors towards gateways. This effect

Algorithm 1 Temperature Field Calculation Function

1: $a = sort_{ascending}(\theta_0, ..., \theta_n)$
2: $j = 0$
3: $t_j = 0$
4: **while** $t_j < a[j]$ **do**
5: $\quad t_{j+1} = t_j + (a[j] - t_j) \cdot \kappa$
6: $\quad j = j + 1$
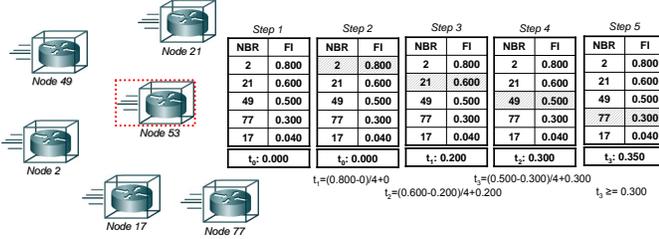7: **end while**
8: $t_{final} = t_j$

Fig. 2. Example of the temperature field calculation with a conductivity value of $\kappa = 1/4$ for node 53: step 1, sort neighbors (nbr) by temperature value; step 2-5 iterate down the table until the given temperature value of the node is higher or equal to the next neighbor; node 71 and 17 do not contribute to the temperature value of node 53: they will increase their values after the next *HEAT beacon* message of node 53 (node 77: 0.313 and node 17: 0.118)

is more pronounced the smaller the parameter κ is chosen. Figure 1 illustrates an example temperature field with $\kappa = 1/4$. As κ is set to a rather small value, one recognizes that the high link redundancy in the right half of the network is taken into account. A step-by-step example of the field calculation function is given in Fig. 2 for $\kappa = 1/4$.

2) Routing Packets From Gateways to Mesh Nodes: An open issue that has not been discussed yet is how to route packets from the Internet gateways back to the mesh nodes. Since following the steepest descent of the temperature field does not necessarily lead to the mesh node that sent packets, an alternative way is required.

Even though there are a multitude of options, most of them are not useful because they degrade the scalability properties of the temperature field approach. Striving to preserve the scalability properties that we gain from using anycast routing, we propose to record the paths of packets being forwarded towards the gateways and use the reverse path to route packets back. This approach can be viewed as some sort of source routing and has the advantage that there is no need for additional control messages. On the other hand, two issues pertaining to source routing arise:

(i) A path to a mesh node is only available after this node has sent a packet to a gateway. Since the vast majority of communication is initiated by mesh nodes, this should not be a critical limitation in practical applications. Should a mesh node act as a server, a dedicated addressing mechanism (e.g. [33]–[35]) is necessary to enable reachability from the Internet. Any mechanism providing this reachability requires periodic registration messages from the mesh node, which allows to keep the path up-to-date.

(ii) The path to a mesh node is only updated whenever packets are sent to gateways. Since most applications generate bidirectional traffic, it should not be a show stopper in practice.

Even most streaming applications require periodic keep-alive messages from the receiver. In the rare case where a mesh node only receives data, provisions need to be taken at the application level to ensure periodic updates of the path information.

D. Improvements for Better Convergence

New node arrivals are easy to handle in our approach and do not require a long time until the temperature field has converged to its steady state: A node joining the network sets its temperature value to 0 and learns a route to the Internet with the first arriving *HEAT beacon*. As more *beacons* arrive, the temperature is adjusted until it converges to its final value.

On the other hand, disappearing nodes (e.g., due to mobility) may cause individual gateways to become unreachable or part of the network partitioned. These type of events might take longer until the temperature fields has converged and in the worst case, some mesh node might not be able to reach any gateways during such period. For this purpose, we propose two additional mechanisms that aim at reducing the convergence time of the temperature field.

The first mechanism consists of an *early HEAT beacon* that is used when nodes detect that a neighbor is no longer reachable and that this neighbor departure has a significant influence on the temperature of that node.

To avoid extensive overhead caused by such *early HEAT beacons*, each node is allowed to delay the forwarding of such *early HEAT beacons* for a short period (e.g., a few broadcast intervals) to ensure that multiple *early HEAT beacon* triggered by the same event are aggregated at the relaying nodes.

If a node leaves unexpectedly, the node departure is detected by one of the neighboring nodes by its maintenance procedure of its neighbor table. If its own temperature value is affected, the node informs its neighboring nodes again using *early HEAT beacon*.

The second mechanism is to avoid a possible effect we call "swing down" of temperatures. This effect occurs for instance in the following scenario. We look at three mesh nodes. Two of them, $node_a$ and $node_c$, are neighbors of the third node, $node_b$. Assume that $node_a$ is close to a gateway and the temperature value of $node_b$ is heavily influenced by $node_a$ and the value of $node_c$ is influenced by $node_b$. In this scenario, a failure of the link between $node_a$ and $node_b$ should lead to a substantial decrease of the temperature of $node_b$ and an even heftier change at $node_c$.

However, the following problem arises. Since only $node_b$ notices the link failure, it re-calculates its temperature value first and incorporates the high temperature of its neighbor $node_c$. In the next step, $node_c$ decreases its temperature value also, and this game goes on an on until both nodes have their correct temperature values. In order to avoid such expensive back and forth adaptation, we use the following technique that is similar to poison reverse [36]. We add the identifiers of the contributing neighbors to the temperature value in each *HEAT beacon*. Using these identifiers, the *HEAT* algorithm at a particular node then ignores all temperature values from neighbors that derived their temperature from this node.

IV. SIMULATION SETUP

To evaluate the performance and scalability of our approach, we performed simulations with Glomosim [37], a network

simulator for wireless networks. We implemented the complete HEAT protocol in Glomosim. The relevant parameters of our protocol are described in Appendix I. These include the *HELLO beacon* interval, the *HELLO beacon* timeout, the delay of *early HELLO beacons*, and the conductivity value κ. As a reference for the performance of our implementation, we use an extended version of AODV as well as OLSR. The settings and assumptions we used for our simulations are described next.

A. Extended AODV

As first reference for the performance and scalability of our implementation, we extended the standard implementation of AODV [38] included in Glomosim according to [11] to support gateway discovery in mesh networks. As proposed in the cited paper, all gateways are connected to a dedicated router that acts as a proxy to the Internet. This router has two tasks: (i) on the forward path, it sends route replies on behalf of hosts in the Internet; (ii) on the backward path, it initiates route requests for nodes in the wireless mesh network. Thus AODV does not have to distinguish between the different gateways and only a common route to the Internet has to be maintained, the route to the dedicated router.

B. OLSR

As second reference for HEAT, we use the OLSR [39] implementation from the University of Niigata [40]. OLSR allows to redistribute routing information from so-called "Non OLSR Interfaces" as the gateway uplink interface to the Internet. In our experiments, we have found that the performance of OLSR drops quickly with increasing mobility. We assume that this is in part due to the long hello interval of 2 seconds. In order to achieve a fair comparison with HEAT, which has a beacon interval of 1 second, we tried to adjust the hello interval of OLSR also to 1 second. With this adjustment, the performance of OLSR improves by roughly 10% and we use this setting for all experiments presented in this paper.

C. Radio Settings

Our simulations are based on a WiFi network. All nodes are equipped with an 802.11b radio with a bandwidth of 11 $Mbps$ and a nominal range of 250 $meters$. As MAC layer protocol we use the 802.11 DFWMAC-DCF w/RTS/CTS and as propagation model the two-ray ground. Due to the large network sizes we use, we were unable to model the effect of intermediate buildings in our city scenarios. However, we expect that the trends of our results also hold when such obstacles are present.

D. City Mobility Model

Instead of relying on simple mobility models like the random waypoint or the random walk mobility models, we developed a more realistic mobility model that accounts for the actual road network of real Swiss cities. The road maps of these cities are extracted from the Swiss geographic information system (GIS) [41] which includes vectorized building and street maps together with speed information. The vectorized map of the city center of Zurich for which we present our results in this paper, is shown in Fig. 3.

The actual node movement is modeled according to the steady-state random trip mobility model [42] on the vectorized maps. That is, a node chooses a random destination in the city
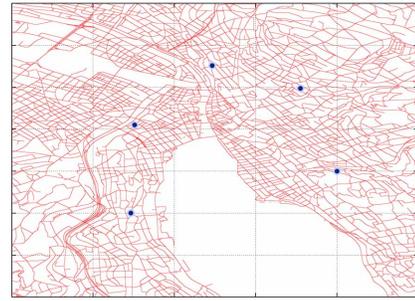

Fig. 3.   Vectorized street map of the city (5km by 7km) of Zurich.

and moves to this position with a constant speed along the shortest way. We do not introduce any pausing of the nodes, and a node therefore begins to move to a new destination as soon as it arrives at the target position. Our model is applied for pedestrians as well as cars since the movements of both are constrained by the streets in the city.

E. Traffic Pattern

We expect wireless mesh networks to be used for Internet-type of applications like web browsing, messaging, chatting, etc. Therefore, we rely on an Internet traffic model as used in [43], [44] consisting of a half-half mix of streaming and web-like traffic. Streaming traffic has a bidirectional constant bit rate of 64 kb/s and the duration of streams are exponentially distributed with an average of 480 seconds. Web-like traffic consists of sporadic 1 kB requests according to an exponentially distributed inter-request time with an average of 10 seconds, followed by response messages with a message size that is Pareto II [45] distributed (average 12 kB, minimal 0.1 kB, maximum 1000 kB).

All traffic in our simulations are from nodes within the wireless mesh network to hosts in the Internet and vice versa (there is no communication between the wireless nodes themselves). Note however, that we do not explicitly simulate the connection between the Internet gateway nodes and the hosts in the Internet because we assume the gateway nodes to be connected to the Internet over broadband connections with high bandwidth, low delays, and low packet losses compared to the wireless mesh network. This means that all results we present are for packets inside the wireless mesh network.

All simulations have a duration of at least 10000 seconds and are always an average over at least 20 runs with different random seeds.

V. SIMULATION RESULTS

We present the following metrics to compare the performance and scalability of HEAT with OLSR and AODV.
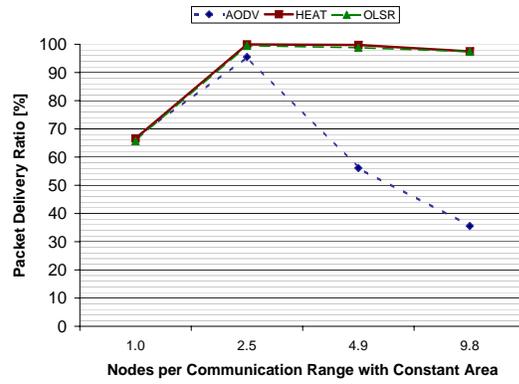
- *Packet delivery ratio* - The number of packets that are successfully received to the total number of packets sent. This metric includes all data packets from the mesh nodes to the gateways as well as packets from the gateways back to the mesh nodes.
- *Routing overhead* - The number of routing control messages that every node sends on average per second.

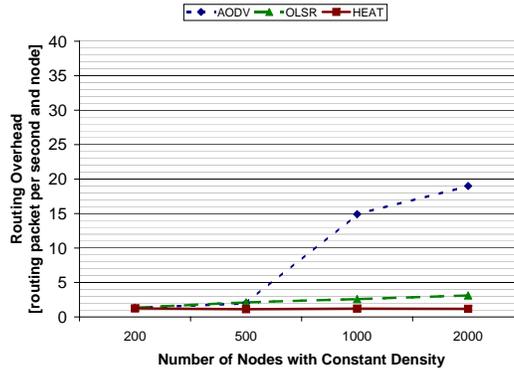### A. Scalability with the Network Size

In a first experiment, we look at how the performance is affected when increasing the network size while keeping the
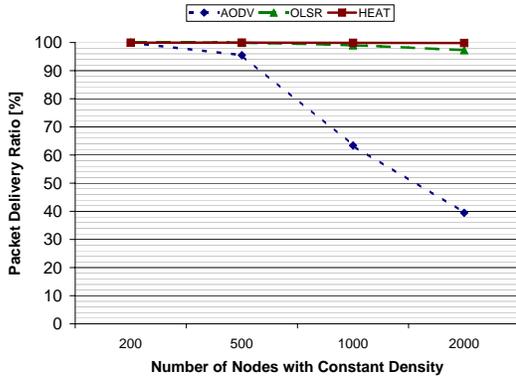
(a) Packet delivery ratio vs. number of nodes.

(b) Packet delivery ratio vs. node density.

(c) Routing overhead in packets/sec. vs. number of nodes.

Fig. 4. Scalability evaluation in static scenario.

average node degree constant. The node degree is kept constant by increasing the simulation area (the section of the maps) as we increase the number of nodes. The results for a *static* scenario with randomly placed nodes, 100 active nodes (half constant-bit rate and half web-like traffic as described in the previous section), 5 Internet gateways, and an approximate average node degree of 3 are shown in Fig 4. In Fig. 4(a), we see the packet delivery ratio. As the network size increases, this ratio for HEAT remains constant at almost 100% and for OLSR decreases only marginally, while it significantly drops for AODV for network sizes greater than 500. The routing overhead, as shown in Fig. 4(c), explains the reason for the performance degradation of AODV. As the network size increases, the average distance between the data sources and the Internet gateways also becomes larger. AODV, which then increases the flooding scope of its extending ring search route discovery algorithm [1], begins to exhaust the network with flooded control messages. In contrast, HEAT has constant overhead per node independent of the network size. The overhead for OLSR is also close to constant but still increases slightly because the link state routing protocol requires full knowledge about the whole topology. The hierarchical flooding mechanism used by OLSR mitigates the scalability problem but is not able to eliminate it completely.

### B. Scalability with the Node Density

In a second experiment, we vary the average node degree to see how the protocols scale with the node density. We obtain different node densities by varying the total number of nodes while keeping the simulation area constant. The

packet delivery ratio for a *static* scenario with randomly placed node on an area of 5 km by 5 km, 100 active nodes, and a total number of nodes ranging from 200 to 2000 is shown in Fig. 4(b). For node degrees smaller than around 2.5, the network is not always in a connected state (some data sources are partitioned from the group of Internet gateways) and the delivery ratio is thus less than 1 for all protocols. An average node degree of approximately 2.5 suffices to have a connected network and all three HEAT, OLSR as well as AODV manage to deliver almost all packets. However, as the node degree becomes larger than 2.5, the performance of HEAT and OLSR remains mainly unaffected, whereas the performance of AODV significantly drops. The reason in this case is again that AODV produces a large amount of overhead from the flooded route request messages. Note that the performance of HEAT and OLSR also slightly degrades since a higher node degree increases the probability of *HEAT beacon* respectively *hello* messages to interfere. However, compared to AODV, the degradation is only marginal (a few percent).

C. Scalability with the Network Dynamics

In a next experiment, we investigate how node mobility affects the routing performance. We consider two scenarios: (i) a scenario with mobile nodes moving at pedestrian speeds (i.e., node speeds that are uniformly distributed between 0.5 m/s and 3 m/s), and (ii) a scenario including nodes moving at car speeds in a city (i.e., node speeds that are uniformly between 10 m/s and 20 m/s).

The results for the pedestrian scenario with a simulation area of 5 km by 5 km, 5 gateways placed a strategic positions, and
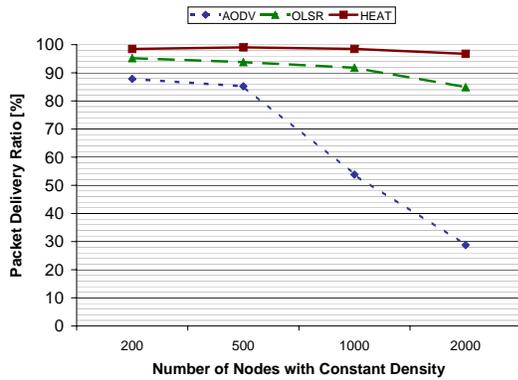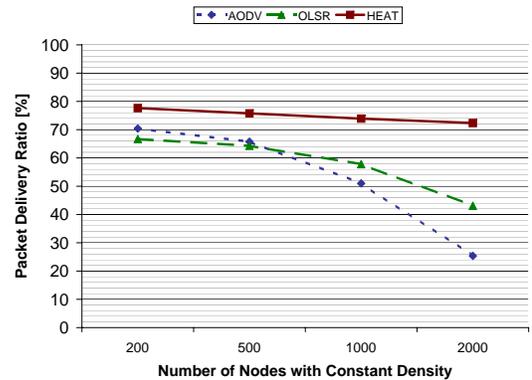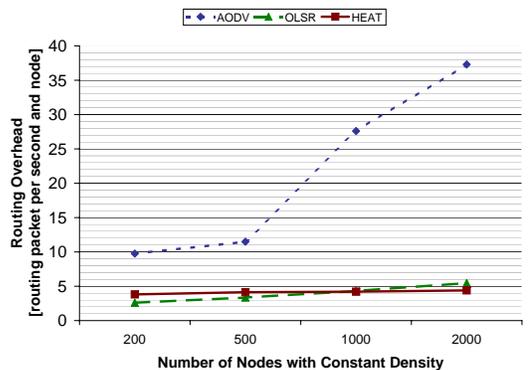
(a) Packet delivery ratio.
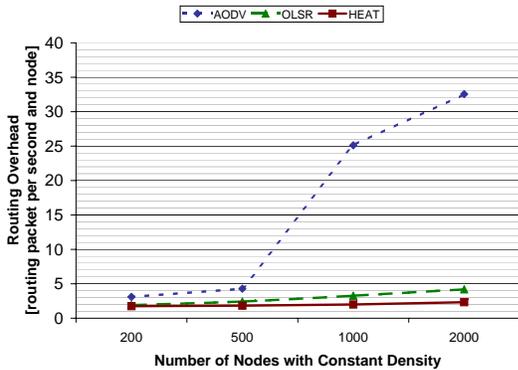
(b) Routing overhead in packets/sec.
Fig. 5. Mobile scenario at pedestrian speeds.

(a) Packet delivery ratio.

(b) Routing overhead in packets/sec.
Fig. 6. Mobile scenario at car speeds.

100 active nodes are given in Fig. 5. At this node speed, the packet delivery ratio of HEAT is almost as good as in the static scenario (as previously shown in Fig. 4) with a slightly higher routing overhead. This additional overhead originates from the protocol enhancements to improve the convergence time as proposed in Section III-D, since the core protocol has an overhead which is independent of the node speed. Looking at the results of OLSR reveals that its packet delivery ratio already decreases slightly for nodes moving at pedestrian speed, particularly in larger networks with longer routes.

Figure 6 shows the performance at car speeds using the same settings. At these node speeds, the performance of all three HEAT, OLSR and AODV is worse than at pedestrian speeds. However, the packet delivery ratio of HEAT remains above 70 percent whereas the ratio of OLSR as well as AODV drops below 40 percent for networks of 2000 nodes. At car speed, OLSR is no longer capable to maintain up-to-date routes while HEAT still does. These results show that the routing performance of HEAT scales better by the node speed in the wireless mesh network than OLSR and AODV.

D. The Effect of the Number of Internet Gateways

Finally, we investigate the scalability with respect to the number of available gateways in the mesh network. In Fig. 7, the packet delivery ratio and the routing overhead are presented for the pedestrian scenario with 1000 nodes moving on an area of 5 km by 5 km with randomly placed gateways of a total number ranging from 1 to 30 as well as 100 active nodes generating traffic.

Obviously, the packet delivery ratio increases as the number of gateways increases. This is mainly because the average distance between mesh nodes and gateways decreases as the number of gateways increases in the same area. Therefore, the average paths are smaller and the performance is less prone to link failures due to mobility. Furthermore, when the number of gateways is too small (e.g., only one gateway), the capacity of the radio interface at the gateway(s) becomes a limiting factor. In other words, the available capacity of the gateway(s) is not sufficient to support all the traffic generated by the mesh nodes.

If we consider the number of gateways that are necessary in the mesh network to provide an average packet delivery ratio that is greater than for example 0.99, we conclude that with HEAT, 5 gateways are sufficient. With 5 gateways, OLSR achieves a packet delivery ratio of about 0.9. With an increasing number of gateways, this ratio rises only slightly because the limiting factor of OLSR in mobile scenarios is that routes become invalid quickly. With AODV, a deployment of more than 30 gateways would be necessary to achieve the same performance. We conclude from this experiment that in mobile scenarios, OLSR and AODV require many more gateways than HEAT to achieve a comparable delivery ratio. A large number of gateways can be saved with our protocol, which makes it particularly suitable for mesh network deployments where the cost of the gateways is an important aspect.

VI. Conclusion

We have investigated the problem of routing between mobile user nodes and Internet gateways in wireless mesh networks. We have proposed a new anycast routing protocol that is based on temperature fields. Our protocol makes use of *local* beacon
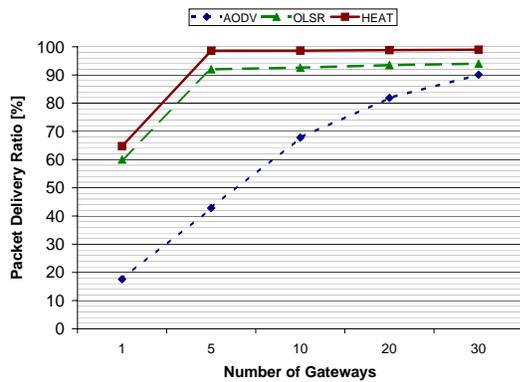
(a) Packet delivery ratio.
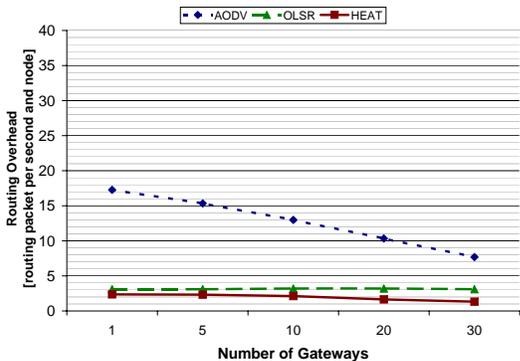
(b) Routing overhead in packets/sec.

Fig. 7. The effect of the number of gateways (mobile scenario at pedestrian speeds).

exchanges to establish routing state and is thus particularly scalable for large and dense networks. Furthermore, temperature fields account for the link diversity and redundancy towards the gateways which makes our protocol particularly robust to node mobility.

We have evaluated and compared the performance of our protocol with the performance of AODV and OLSR through extensive simulations with mobility patterns extracted from geographical data of Swiss cities. Our results show that HEAT and OLSR achieve packet delivery ratios in static dense and large mesh networks which are above 0.95. In the same settings, AODV fails to deliver more than 30 percent of the packets successfully. In mobile scenarios with car mobility, HEAT outperforms AODV as well as OLSR in terms of the packet delivery ratio by more then a factor of two. Finally, we have shown that HEAT is able to provide a packet delivery ratio that is higher than 0.99 with 5 gateways while the ratio for OLSR is 0.9. AODV would require more than 30 gateways to achieve a comparable performance in the same scenario.

REFERENCES

[1] C. Perkins, "Ad-hoc on-demand distance vector routing," in *MILCOM '97 panel on Ad Hoc Networks*, 1997.
[2] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol," IETF Internet Draft, draft-ietf-manet-olsr-11.txt, July 2003.
[3] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, Imielinski and Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353.
[4] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIG-COMM'94 Conference on Communications Architectures, Protocols and Applications*, 1994, pp. 234–244.
[5] Z. J. Haas and M. R. Pearlman, "The performance of query control schemes for the zone routing protocol," in *SIGCOMM*, 1998, pp. 167–177.
[6] J.-C. Chen and S. Li and S.-H. Chan and J.-Y. He, "WIANI: Wireless Infrastructure and Ad-Hoc Network Integration," in *Proceedings of IEEE International Conference on Communications (ICC)*, 2005.
[7] Ren-Hung Hwang and Chiung-Ying Wang and Cheng-Ying Li and Yuh-Shyan Chen, "Mobile IPv6-based Ad Hoc Networks: Its Development and Application," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 11, pp. 2161–2171, Nov 2005.
[8] C. Ahlund and A. Zaslavsky, "Extending Global IP Connectivity for Ad Hoc Networks," *Kluwer. Telecommunication Systems, Modeling, Analysis, Design and Management*, vol. 24, no. 2-4, 2003.
[9] P. Ratanchandani and R. Kravets, "A hybrid approach to internet connectivity for mobile ad hoc networks," in *Proceedings of IEEE WCNC*, 2003.
[10] Y. Sun, E. Belding-Royer, and C. Perkins, "Internet connectivity for ad hoc mobile networks," in *in International Journal of Wireless Information Networks, special issue on Mobile Ad hoc Networks*, 2002.
[11] M. Michalak and T. Braun, "Common gateway architecture for mobile ad-hoc networks," in *WONS '05: Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 70–75.
[12] R. Baumann, "VANET: Vehicular Ad Hoc Networks," Master's thesis, ETH Zurich, 2004.
[13] B.-N. Cheng, M. Yuksel, and S. Kalyanaraman, "Orthogonal rendezvous routing protocol for wireless mesh networks," in *Proc. of ICNP*. Santa Barbara, California: IEEE, Nov. 2006.
[14] M. Mosko and J. J. Garcia-Luna-Aceves, "Ad hoc routing with distributed ordered sequences," in *IEEE INFOCOM 2006*, Barcelona, Spain, April 2006.
[15] M. Mosko and J. Garcia-Luna-Aceves, "Multipath routing in wireless mesh networks," in *Proc. IEEE Workshop on Wireless Mesh Networks (WiMesh)*, Santa Clara, USA, sep 2005.
[16] H. Ju and I. Rubin, "Backbone topology synthesis for meshed wireless LANs," in *IEEE INFOCOM 2006*, Barcelona, Spain, April 2006.
[17] C. Partridge, T. Mendez, and W. Milliken, "Host Anycasting Service," RFC 1546 (Informational), Nov. 1993.
[18] V. Park and J. Macker, "Anycast Routing for Mobile Services," in *Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, USA, March 1999.
[19] Jianxin Wang and Yuan Zheng and Weijia Jia, "An AODV-based Anycast Protocol in Mobile Ad Hoc Network," in *Proc. of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communication*, Beijing, China, September 2003.
[20] J. Wang, Y. Zheng, and W. Jia, "A-DSR: A DSR-based Anycast Protocol for IPv6 Flow In Mobile Ad Hoc Networks," in *Proc. of the IEEE Vehicular Technology Conference*, Orlando, Florida, USA, October 2003.
[21] D. Katabi and J. Wroclawski, "A Framework for Scalable Global IP-Anycast (GIA)," in *Proc. of ACM SIGCOMM*, Stockholm, Sweden, August 2000.
[22] H. Ballani and P. Francis, "Towards a Global IP Anycast Service," in *Proc. of ACM SIGCOMM*, Philadelphia, USA, August 2005.
[23] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet Indirection Infrastructure," in *Proceedings of ACM SIGCOMM Conference*, Pittsburgh, PA, USA, August 2002.
[24] V. Lenders, M. May, and B. Plattner, "Density-based vs. Proximity-based Anycast Routing for Mobile Networks," in *IEEE INFOCOM*, Barcelona, Spain, April 2006.
[25] V. Park and S. Corson, *Temporally-Ordered Routing Algorithm (TORA)*, IETF Internet Draft, July 2001.
[26] A. Basu, A. Lin, and S. Ramanathan, "Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm," in *Proceedings of the ACM annual conference of the Special Interest Group on Data Communication (SIGCOMM'03)*, Karlsruhe, Germany, August 2003.
[27] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in *Proceedings of the ACM/IEEE Internation Conference on Mobile Computing and Networking (Mobicom '00)*, Boston, USA, 2000.
[28] J. Faruque and A. Helmy, "RUGGED: RoUting on finGerprint Gradients in sEnsor Networks," in *Proceedings of the IEEE International Conference on Pervasive Services (ICPS)*, Novi Sad, Serbia and Montenegro, July 2004.
[29] S. Toumpis and L. Tassiulas, "Packetostatics: Deployement of Massively Dense Sensor Networks as an Electrostatic Problem," in *IEEE INFO-COM*, Miami, USA, March 2005.
[30] Q. Li, M. D. Rosa, and D. Rus, "Distributed Algorithms for Guiding Navigation across a Sensor Network," in *Proceedings of ACM MobiCom*, San Diego, California, September 2003.

[31] Vincent Lenders and Martin May and Bernhard Plattner, "Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Taormina, Italy, June 2005.

[32] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Mobile Computing and Networking*, 2000, pp. 56–67.

[33] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host Identity Protocol (HIP)," IETF draft-ietf-hip-base-00, October 2003.

[34] C. Perkins, "IP Mobility Support," RFC 2002 (Proposed Standard), Oct. 1996.

[35] Rainer Baumann and Olga Bondareva and Simon Heimlicher and Vincent Lenders and Martin May, "A Macro Mobility Notification Protocol for Hybrid Wireless Mesh Networks," in *Proceedings of 14th IEEE International Conference on Network Protocols (ICNP)*, 2006.

[36] G. Malkin, "RIP Version 2," RFC 2453 (Standard), Nov. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2453.txt

[37] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: A library for parallel simulation of large-scale wireless networks," in *Workshop on Parallel and Distributed Simulation*, 1998, pp. 154–161.

[38] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561 (Experimental), Jul. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt

[39] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626 (Experimental), Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[40] Sota Yoshida and Masato Goto and Takashi Hasegawa and Kenichi Mase, "OLSRv2 Implementation for Glomosim," January 2006. [Online]. Available: www.net.ie.niigata-u.ac.jp/mase/olsr

[41] Federal Office of Topography, "Swiss Geographic Information System." [Online]. Available: www.swisstopo.ch

[42] J.-Y. Le Boudec and M. Vojnovic, "Perfect Simulation and Stationarity of a Class of Mobility Models," in *IEEE Infocom*, 2005.

[43] A. Feldmann, A. C. Gilbert, P. Huang, and W. Willinger, "Dynamics of IP traffic: A study of the role of variability and the impact of control," in *SIGCOMM*, 1999, pp. 301–313.

[44] U. Fiedler, "Evaluating performance in systems with heavy-tailed input - a quantile-based approach," Ph.D. dissertation, ETH Zurich, Aug. 2003.

[45] N. L. Johnson, S. Kotz, and A. W. N. Balakrishnan, *Continuous univariate distributions: Vol. 1.* Wiley, New York, 1994.

Appendix I
Selection of Parameters

In this Appendix, we present an initial set of experiments to find appropriate values for the timing parameters of our algorithm. These parameters are:

- *HEAT beacon* interval,
- *HEAT beacon* timeout,
- delay of *early HEAT beacon*,
- and the conductivity value κ.

For these simulations, we use a static scenario of 1000 mesh nodes and 5 gateways. All nodes (including the gateways) are randomly distributed over an area of $5000 \times 5000 \ m^2$.

Determining the *HEAT beacon* interval is a classical trade-off between communication overhead and convergence time. In order to measure the convergence time, we wait until the routing has converged and then eliminate 10% of all nodes (mesh nodes and gateways). The convergence time is then determined as the time it takes until the routing topology has converged again. Note that we consider the routing as converged as soon as there are no more route changes. Figs. 8 and 9 plot the convergence time and the routing overhead vs. the *HEAT beacon* interval. As expected, the convergence time increases linearly with the *HEAT beacon* interval. Also, the overhead decreases with an increasing *HEAT beacon* interval. As a result, we set the *HEAT beacon* interval to 1 second and the *HEAT beacon* timeout equal to $3 \times$ *HEAT beacon* interval. This setting of the *HEAT beacon* interval allows up to two *HEAT beacon* messages to be lost before a link is considered to be down. Further, we set the delay for *early*
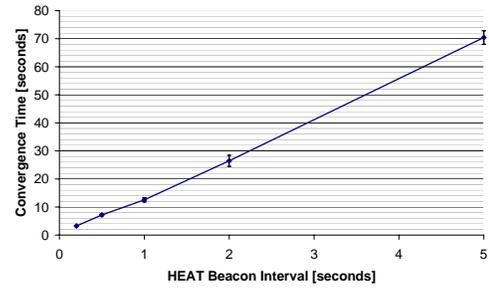

Fig. 8. Parameter selection: Convergence time versus *HEAT beacon* interval.
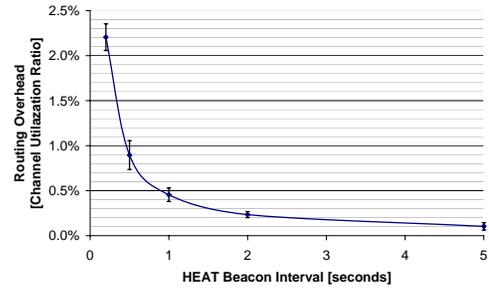

Fig. 9. Parameter selection: Per node convergence time versus *HEAT beacon* interval.
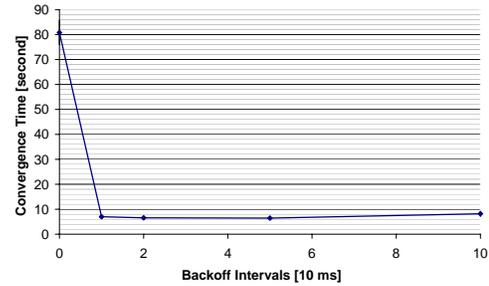

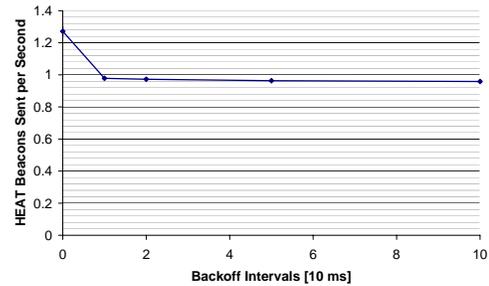Fig. 10. Parameter selection: Convergence time versus delay of *early HEAT beacon* in backoff intervals.

Fig. 11. Parameter selection: HEAT beacons sent per second versus delay of *early HEAT beacon* in backoff intervals.

HEAT beacon messages to two times the backoff interval of the 802.11 MAC layer, which results in an *early HEAT beacon* delay of $20 \ ms$. Longer delays would have a negative impact on the convergence time (see Fig. 10). Shorter delays would lead to a greater number of regular *HEAT beacons* (see Fig. 11). We set the conductivity value κ for the temperature field calculation function to $1/4$. This value has shown to provide the best tradeoff between the convergence time, the communication overhead, and the robustness from the path and link redundancy.