

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Ant Algorithms for Routing in Wireless Multi-Hop Networks

Martina Umlauft and Wilfried Elmenreich

Abstract

Wireless Multi-Hop Networks (such as Mobile Ad hoc Networks, Wireless Sensor Networks, and Wireless Mesh Networks) promise improved flexibility, reliability, and performance compared to conventional Wireless Local Area Networks (WLAN) or sensor installations. They can be deployed quickly to provide network connectivity in areas without existing backbone/back-haul infrastructure, such as disaster areas, impassable terrain, or underserved communities. Due to their distributed nature, routing algorithms for these types of networks have to be self-organized. Ant routing is a bio-inspired self-organized method for routing, which is a promising approach for routing in such Wireless Multi-Hop Networks. This chapter provides an introduction to Wireless Multi-Hop Networks, their specific challenges, and an overview of the ant algorithms available for routing in such networks.

Keywords: ant algorithms, wireless networks, ad hoc networks, mesh networks, wireless sensor networks, multi-hop

1. Introduction

Wireless Mesh Networks (WMNs) and Mobile Ad-Hoc Networks (MANETs) are applied in situations where there is no predefined network structure consisting of routers and base station or where the network is dynamic due to a growing number of nodes or mobile nodes moving into areas that have not been previously covered by a base station. Examples for such networks are Wireless Sensor Networks (WSNs) [1], vehicle ad-hoc networks [2], Wireless Sensor Network for children's computer in developing countries [3], and open grassroots initiatives to support free computer networks such as the Freifunk initiative in Germany [4] or the Funkfeuer initiative in Austria [5].

Routing of messages is a major challenge in such networks due to the dynamic or not *a priori* known network structure. Besides the problem of finding an optimum (or acceptable route) for a message, there is also a mutual influence of a used route on other routes. This calls for a self-organizing approach [6] of choosing routes that are near-optimal on a global level with a decision based on local information.

Artificial ant algorithms give a promising approach for such algorithms. Artificial ant algorithms are bio-inspired algorithms based on real ants' foraging behavior using a local gradient-following search strategy with pheromone trails. There are different ways how an ant-inspired algorithm can be implemented in Wireless Multi-Hop Networks, for example, by representing ants via network packets and

the pheromone by values assigned to the network nodes. Besides the mapping of biological properties into a computer network, Algorithms differ in how route discovery and maintenance are implemented. This chapter investigates different ant algorithms and discusses their applicability to routing in wireless multi-hop networks.

The following section gives an introduction to ant-inspired emergence and self-organization. In Section 2, three forms of Wireless Multi-Hop Networks are introduced (*MANET*, *WSN*, or *WMN*). Section 3 describes first the seminal ant routing algorithm developed for routing in such networks, *AntHocNet*, followed by an overview of algorithms which build upon this algorithm. Section 4 provides a summary and concluding remarks.

2. Wireless Multi-Hop Networks

Recent advances in wireless communications have enabled the development of Wireless Multi-Hop Networks. Often also called Wireless Ad-Hoc Networks we will in the following use the term *Wireless Multi-Hop Network* to avoid confusion with (Mobile) Ad-Hoc Networks described in the next section. A Wireless Multi-Hop Network is a network where nodes communicate via several hops of wireless transmissions over equivalent nodes instead of a central base station. In this chapter we distinguish between (Mobile) Ad-Hoc Networks, Sensor Networks and Wireless Mesh Networks, which are described in the following.

2.1 (Mobile) Ad-Hoc Networks

An *Ad-Hoc Network* is defined as a network where all nodes communicate with one another on an ad-hoc basis without a central base station [7]. While sometimes the term is used in literature to denote a Wireless Multi-Hop Network, we use it here to denote a wireless network that does not differentiate between client nodes and dedicated routing nodes. The typical node is a somewhat powerful device such as a (ruggedized) laptop, smartphone, first-responder communication device, or a device that is integrated in a vehicle, all of which are possibly mobile (see **Figure 1**). In this case, the network is called a *Mobile Ad-Hoc Network (MANET)*. These networks were originally developed for military use to enable troop communications in areas where no communications infrastructure was previously deployed. *MANETs* are also envisioned for emergency and disaster networking where the borders between *MANETs* and *WMNs* (see below) are flowing [8].

Several “classical” routing algorithms have been developed for *MANETs*, the most prominent being:

DSR: Dynamic Source Routing Protocol was developed in 1994 by David B. Johnson [9]. It is a *reactive* protocol which means that the protocol only builds routes on-demand. This is advantageous in highly volatile networks where it makes no sense to invest routing overhead to build and maintain routes that might go stale before they are used. On the other hand, traffic incurs a route-setup delay because the route is not built in advance. DSR is being standardized by the IETF [10],

DSDV: Destination-Sequenced Distance-Vector Protocol was developed by Charles Perkins and Pravin Bhagwat in 1994 [11]. It is a *pro-active* routing protocol, which means that routes are built in advance. This avoids the route-setup delay incurred by reactive protocols but, in highly volatile networks, a lot of routing overhead might be spent to set up and maintain routes that break before they can be used,

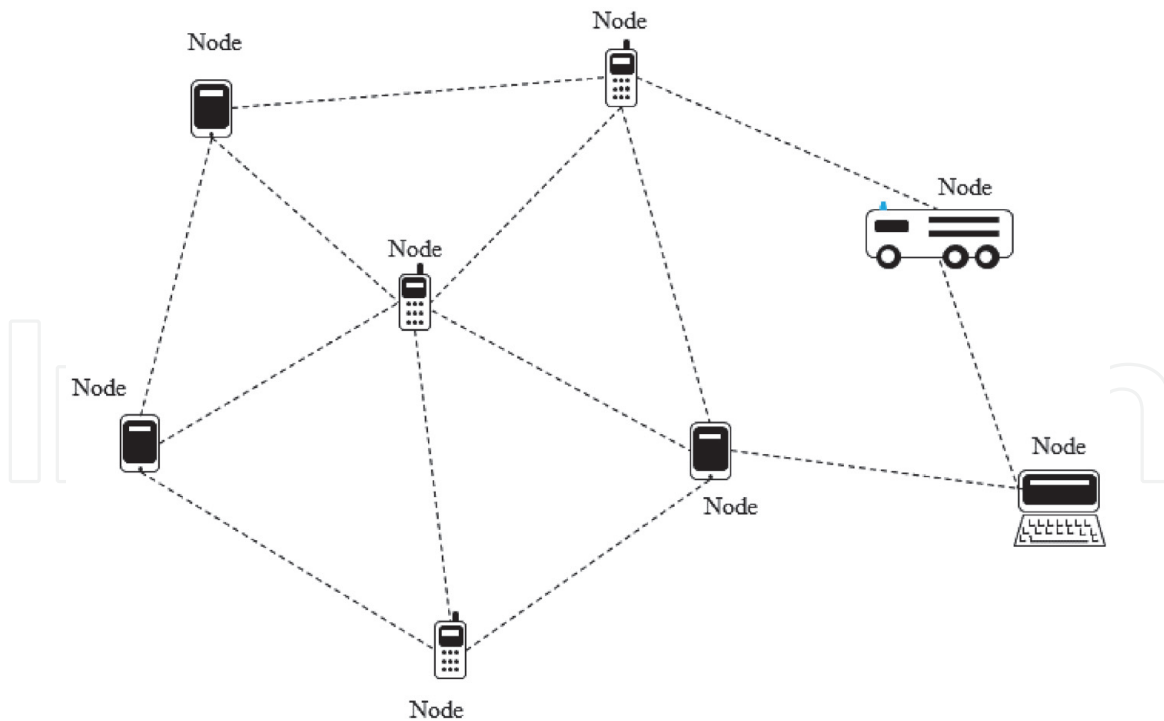


Figure 1.
 MANET architecture. Nodes connect on an ad-hoc basis without dedicated routers or base stations.

AODV: Ad hoc On-Demand Distance Vector Routing [12] developed 1999 by Charles Perkins and Elizabeth Royer which is a reactive routing protocol, and OLSR: Optimized Link State Routing Protocol [13] which was developed by Jacquet et al. in 2001. It is a proactive routing protocol and has been standardized by the IETF as experimental RFC 3626 [14].

2.2 Sensor Networks

A *Sensor Network* (or *Wireless Sensor Network*, WSN) is a wireless network that consists of many sensor nodes which are spatially deployed to cooperatively monitor physical or environmental conditions (see **Figure 2**). WSNs were initially developed for the military for applications such as battlefield surveillance. However, WSNs are now used in civilian application areas like environmental monitoring [15–17]. In contrast to MANETs sensor nodes are typically tiny (so-called “motes”) and deployed densely in huge numbers, often on inaccessible terrain. Therefore, sensor network protocols must be self-organizing.

One of the major problems in sensor networks is the limited power as each sensor only has a small battery. While some sensors use technologies like solar cells to refresh their batteries, routing protocols for sensor networks typically try to optimize for power efficiency [18]. In WSNs, typically the sink initializes routing by issuing a query for measurement data. Sensor nodes answer the query by sending their data back to the sink. To save energy, data may be aggregated along the way (*data-centric routing*). To facilitate this, the sensor field is often divided into clusters or subnets. All nodes of a cluster first send their data to the respective cluster head, which then processes and routes the data to the sink. If the sensors are equipped with location finding devices like GPS (Global Positioning System), knowledge of the position can be used to ease cluster formation or perform geo-routing. Well known non-ant routing protocols for WSNs include:

Gossiping is an early approach derived from flooding [19],

SPIN: Sensor Protocols for Information via Negotiation is a family of protocols based on data-centric routing, developed by Heinzelman et al. in 1999 [20],

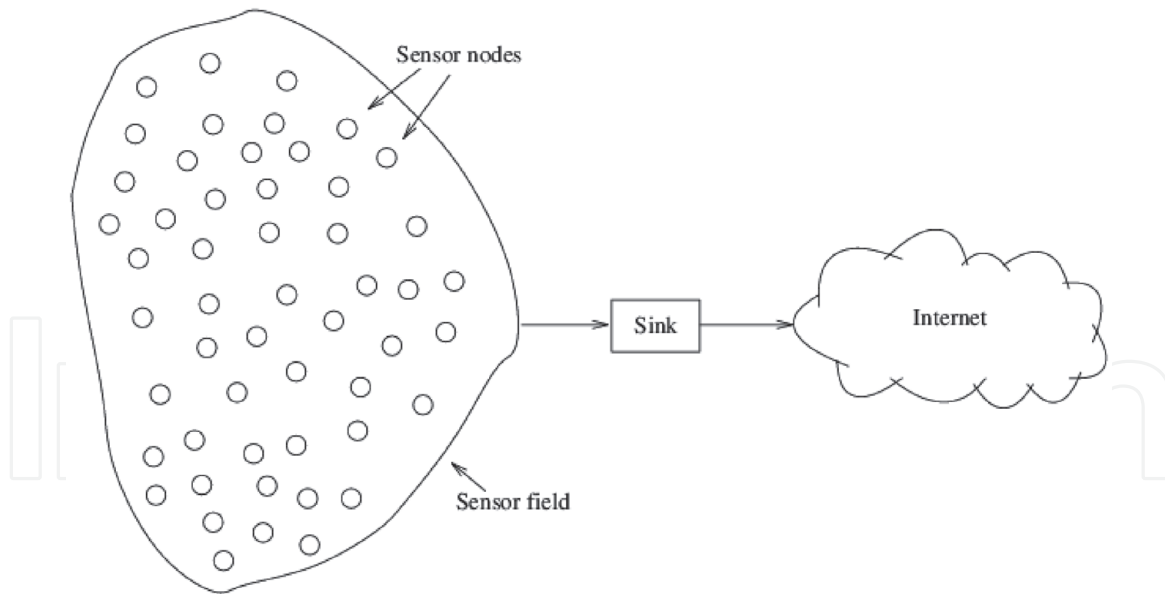


Figure 2.
WSN architecture. Sensor nodes are deployed in a sensor field and deliver their measurements through a sink to the Internet.

GPSR: Greedy Perimeter Stateless Routing in Wireless Networks, a geo-routing algorithm developed by Brad Karp in 2000 [21, 22],

LEACH: Low-Energy Adaptive Clustering Hierarchy developed by Heinzelman et al. in 2000 [23] which is a clustering-based protocol that minimizes energy dissipation,

SAR: Sequential Assignment Routing is an algorithm which selects paths based on energy resources, a QoS metric, and the packet's priority level. SAR was developed by Sohrabi et al. in 2000 [24] as part of a protocol suite for WSNs, and

Directed Diffusion, an approach using data-centric dissemination, reinforcement-based adaptation to the empirically best path, and in-network data aggregation and caching. Directed Diffusion was developed by Intanagonwiwat et al. in 2000 [25].

2.3 Wireless Mesh Networks

A *Wireless Mesh Network* (WMN) is a wireless networking architecture in which nodes are connected via a *wireless backbone* [26]. In contrast to *MANETs*, though, the wireless backbone in a WMN is typically fixed. Iow, a WMN consists of non-moving wireless mesh router nodes which constitute the wireless backbone and (potentially mobile) client nodes (see **Figure 3**). Router nodes can be mesh routers only (so-called Mesh Points, MP [27]) or act as combined *WLAN* Mesh Access Points (MAPs)/routers.

WMNs can be used as access networks to the Internet, where one or several Mesh Portal Points (MPP) connect the mesh to the Internet. They can also be used in disaster areas, for emergency response teams, and for the military. The boundary towards *MANETs* is somewhat fluid in these cases [8]. Consider, for example, a rescue operation where wireless routers are installed on top of firetrucks. When the firetrucks arrive at the scene, they stop and provide the wireless backbone for the firefighters' communication devices. This scenario can be seen as a WMN as well as as a *MANET*.

Even with a stationary wireless backbone, the characteristics of wireless channels and the interaction of the MAC layer with the higher layers in the network

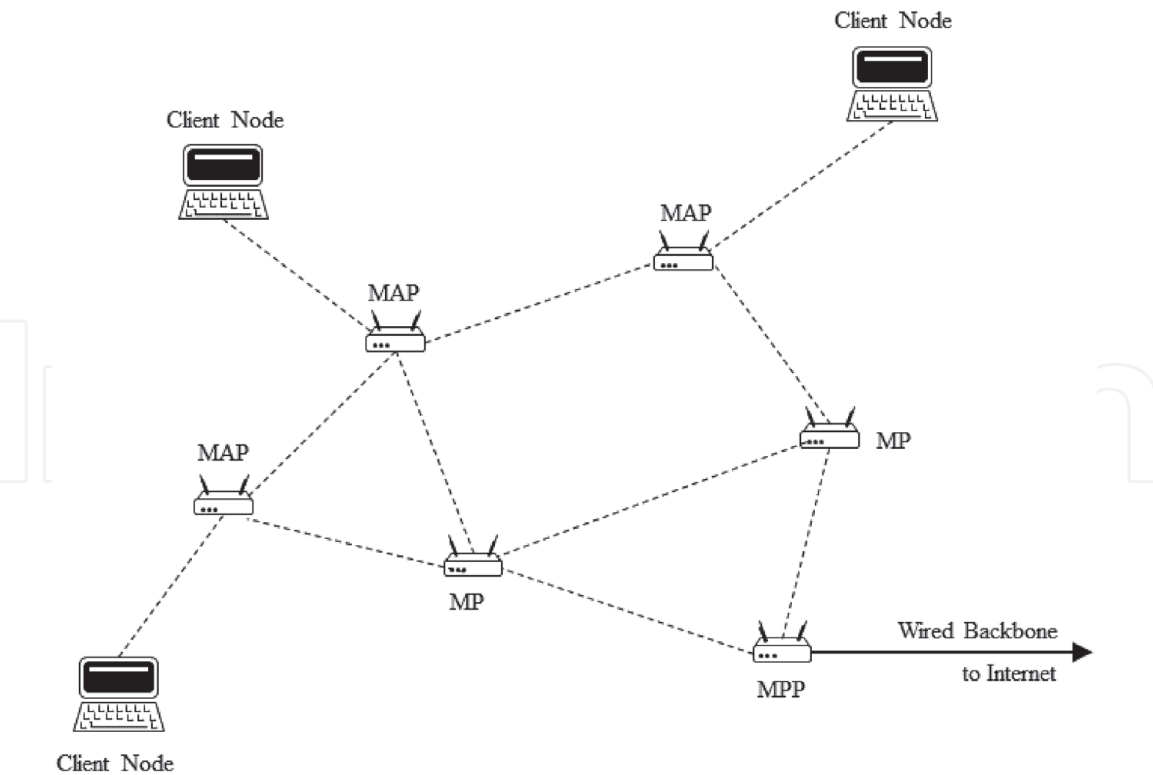


Figure 3.
WMN architecture. Client nodes connect via a wireless backbone. MP, Mesh Point; MAP, Mesh Access Point; MPP, Mesh Portal Point.

stack make routing in WMNs a hard problem. Wireless links vary over time and problems like the hidden node problem or the exposed node problem influence routing algorithm’s performance. Therefore, a “wireless-aware” routing algorithm is necessary.

WMNs have slightly different constraints and pose different problems than MANETs and sensor networks. For example, the power constraint which is very prominent in sensor networks typically does not exist in WMNs [8]. When WMNs are used as access networks to the Internet “normal internet traffic” has to be assumed. This means application traffic like streaming, web browsing, VoIP (Voice over IP) and video conference traffic, or email, which use the standard TCP (or UDP) protocol stack. Routing algorithms for WMNs have mostly been adapted from the (Mobile) Ad-Hoc Networking Community.

3. Ant Colony Optimization and Ant-Routing Algorithms

Ant algorithms are inspired by the natural foraging behavior of certain species of ants. Based on the famous double bridge experiment, reported on by Goss, Aron, Denebourg and Pasteels in 1989 [28], ant-inspired optimization was then codified into an *Ant Colony Optimization* metaheuristic [29] which was originally implemented in algorithms such as *Ant System* [30] and *Ant Colony System* [31].

In general, algorithms using the *Ant Colony Optimization* metaheuristic work as follows: an optimization problem is transformed into a graph $G = (V, E)$, ants travel along the graph using pheromones (if present) to choose a path stochastically and after the ants have finished their travel, the pheromone values in the graph are updated according to the “goodness” of the solutions found by the ants. Many algorithm variants, also improve their results with a local search phase that is applied before updating the pheromone values. Besides other combinatorial

optimization problems, these algorithms have been shown to be able to solve the traveling salesman problem. In *Ant System*, the first algorithm to implement the *Ant Colony Optimization* metaheuristic, the ants choose their path according to Eq. (1):

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{c_{il} \in N(s^p)} \tau_{il}^\alpha \eta_{il}^\beta} & \text{if } c_{ij} \in N(s^p), \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where an ant k in a city i chooses the next city j with probability p_{ij}^k with s^p the partial solution constructed so far and $N(s^p)$ the set of possible edges leading only to cities not visited so far. The parameters α and β balance the importance of pheromone versus the local heuristic $\eta_{ij} = 1/d_{ij}$ with d_{ij} the distance between city i and city j .

Pheromones are updated using the update rule in Eq. (2):

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k. \quad (2)$$

with ρ the evaporation rate, m the number of ants and $\Delta \tau_{ij}^k$ proportional to the inverse of the length of the tour ant k took if that link was chosen (0 otherwise).

A variant aimed specifically at (wired) networks is *AntNet* [32]. These algorithms were not developed with Wireless Multi-Hop Networks in mind, though. As described above, Wireless Multi-Hop Networks have their own challenges in addition to the challenges of routing in a fixed network.

In the following, we will describe the seminal ant routing algorithm developed for Wireless Multi-Hop Networks, *AntHocNet* [33], and then give an overview of the typical features of other ant routing algorithms for these types of networks.

3.1 AntHocNet

AntHocNet [33], 2005, by Di Caro, Ducatelle, and Gambardella is the seminal ant algorithm developed for mobile ad-hoc networks. It addresses the special challenges that such wireless networks pose: bandwidth is typically less than in fixed networks, and links can change their quality or break. Therefore, *AntHocNet* is realized as a hybrid algorithm that combines features from pro-active and reactive routing protocols. In this way, it does not waste resources to set up paths before any packet is sent, which might not exist anymore by the time they are eventually needed.

Like all wireless routing algorithms, nodes running *AntHocNet* need to determine which other nodes are reachable by wireless transmission (iow. the one-hop neighborhood). *AntHocNet* nodes do this by broadcasting very short “hello” messages at regular intervals. Receiving nodes then set up these neighboring nodes in their respective routing tables, but without any routing information yet. These “hello” messages are also used to detect link failures.

When a new data packet is to be sent from a source node s to a destination node d , the algorithm enters its reactive path setup phase. There exist two possibilities: either there already exists routing information for a path between s and d (after the protocol has run for a while and packets have already been sent) or not. Depending on whether routing information already exists or not, *AntHocNet* sends so-called “forward ants” either by broadcasting them (if no routing information for the required route exists yet) or by unicasting them stochastically along one of the

already known routes. For unicasting, the pheromone routing tables at the intermediate nodes are exploited and the next hop n towards the destination d is chosen stochastically with a probability P_{nd} according to Eq. (3):

$$P_{nd} = \frac{(T_{nd}^i)^\beta}{\sum_{j \in \mathcal{N}_d^i} (T_{jd}^i)^\beta}, \beta \geq 1. \quad (3)$$

with i the current node, n the next hop, T the respective pheromone value, $\mathcal{N}_d^i$ the set of possible neighbors and a coefficient β that controls how explorative the algorithm behavior is.

If there is no pheromone information available yet, the forward ant is broadcasted. To avoid flooding the network with too much traffic, these broadcast ants are restricted in several ways: 1) after a number of hops, they are killed, 2) when a node receives several ants stemming from the same broadcast (that took different paths to reach this node), it will only pass on those ants which came via sufficiently good paths (using the number of hops and travel time as metrics). The threshold for this can be set by a parameter α_1 . In this way, several parallel paths can be explored while the worst paths are quickly excluded and overhead (which is always of special concern in wireless networks) is kept at a reasonable level. A second parameter α_2 is used to spread paths more widely among the network: broadcast ants which took a different first hop than previous ants stemming from the same broadcast, this less restrictive parameter α_2 is applied instead of α_1 .

Ants memorize the path they travel and when a forward ant has reached the destination node, a so-called backward ant is created which travels back the path $\mathcal{P} = s \rightarrow n_1 \rightarrow n_2 \rightarrow \dots \rightarrow d$ it came. This backward ant then updates all the pheromone information along the path according to Eq. (4):

$$T_{nd}^i = \gamma T_{nd}^i + (1 - \gamma) \tau_d^i, \quad \gamma \in [0, 1]. \quad (4)$$

where τ_d^i is an expression of the “goodness” of the path, based on an estimate of the average time to send a packet over the path $\mathcal{P}$ calculated from measurements at each node’s MAC layer.

After one or several path(s) has been found and while a data session is running, *AntHocNet* forwards the data packets stochastically along all the available paths using the same Eq. (3) as the forward ants but with a higher value of β . This means that data packets have a more exploitative behavior than ant packets which explore more.

During a data session, *AntHocNet* enters its pro-active phase and sends forward ants in addition to the data packets. These again use Eq. (3) but have a small probability of being broadcast instead. The ants that follow the existing path via Eq. (3) update the current quality of the existing path while those ants that are broadcast can potentially find new, better paths which will then be immediately used as potential paths to route data. Due to the way paths are determined and updated during the pro-active phase and due to the stochastic nature of the data routing, data packets are sent in an automatically load-balanced way through the network which especially helps with wireless transmission as two parallel paths use the same transmission medium and therefore can potentially greatly influence each other.

AntHocNet also addresses link failures, which occur much more frequently in the wireless domain. As mentioned before, link failure can be detected via “hello” messages – if there has not been an “hello” message for a certain amount of time

(several times the regular sending interval), a link to a node will be considered broken. A link is also considered broken if a unicast message to a node fails. The algorithm then enters its local path repair mode where it broadcasts so-called “path repair ants” that work just like the forward ants in reactive mode, except that they are more limited in their maximum number of allowed broadcasts. If a path can be repaired within a certain amount of time, the data packets (which will have been buffered in the meantime) will be sent to the destination node. If the path can not be re-established within a reasonable time limit, the data is discarded and link failure notifications are broadcasted to the surrounding nodes.

3.2 ARA

ARA (Ant-Colony-based Routing Algorithm) was proposed for MANETs by Güneş, Sorges and Bouazizi in 2002 [34]. ARA is based on a version of *Ant Colony Optimization* which the authors call “Simple ACO”. Its main goal is to reduce the overhead of routing as compared to classical routing algorithms. The algorithm consists of three phases: route discovery, route maintenance, and route failure handling. It uses routing tables at each node n_i which consist of records of the form $\langle n_d, n_n, \varphi_{i,n} \rangle$ where n_d is the destination node, n_n the next hop node and $\varphi_{i,n}$ the pheromone value for this link and destination. Ants carry a sequence number and the source address they originated from.

The transition rule looks very much like that of *AntHocNet* (cf. also Eq. (3)) with a coefficient of $\beta_1 = 1$:

$$p_{i,n} = \begin{cases} \frac{\varphi_{i,n}}{\sum_{j \in N_i} \varphi_{i,j}} & n \in N_i \\ 0 & n \notin N_i \end{cases} \quad (5)$$

where $p_{i,n}$ is the transition probability of going from the current node n_i to node n_n and N_i is the set of one-hop neighbors of n_i .

In contrast to *AntHocNet*, though, ants are used differently as follows:

- During route discovery, forward ants do not follow the transition rule but are broadcasted instead. Duplicate forward ants can be detected by their sequence number and source address and are not forwarded.
- On its way through the network, the forward ant immediately updates the pheromone tables at the nodes – for the way back to the source. The forward ant is interpreted similarly to backward ants in *AntHocNet*. When a forward ant arrives at a node, the routing table entry where n_d equals the source address in the ant and n_n equals the last hop the ant took is updated. Backward ants are used analogously and establish the path to the destination node as usual.

The pheromone update rule in ARA is quite simple; an ant changes the pheromone value moving from node n_i to n_n by a constant amount:

$$\varphi_{i,n} := \varphi_{i,n} + \Delta\varphi. \quad (6)$$

Güneş et al. suggest that the number of hops an ant has traveled to the current node could also be included in the calculation of the new amount of pheromone. Pheromone is evaporated in regular intervals according to the evaporation rule shown in Eq. (7). The authors suggest that the link quality measurements should be

incorporated into the evaporation rule rather than into the pheromone update rule as usual. This has the advantage that nodes can update local changes of link quality much more quickly. On the other hand, the disadvantage of this method is that the quality reflected by the amount of pheromone reflects local link quality only instead of end-to-end path quality.

$$\varphi_{i,n} := (1 - q) \cdot \varphi_{i,n}, \quad q \in (0, 1] \quad (7)$$

Route maintenance is done by observing the traffic flowing through the network. Traffic does not have to be encapsulated in ant packets; rather, nodes autonomously update the pheromone tables according to the pheromone update rule already shown in Eq. (6). For each packet of traffic observed by the node, the pheromone value is increased by the constant amount $\Delta\varphi$. This has the advantage that route reinforcement happens “automatically” without the need for extra ant packets.

To prevent the creation of loops, ARA implements a simple loop avoidance mechanism. When a node recognizes the duplicate reception of a data packet (identifiable by sequence number and address), it sets an error flag and sends the packet back to the previous node which removes the link from its routing table.

Route failures are recognized by missing acknowledgements. When a link fails, a node first checks whether it has another route to the required destination in its routing table. If this is the case, it sends the packet via this alternative link. If not, the node informs its neighbors anticipating that they can relay the packet. Failing this, the mechanism tracks back until it arrives back at the source node. In that case, a new route discovery phase has to be initiated by the source node.

3.3 ARAMA

Ant Routing Algorithm for Mobile Ad-hoc networks (ARAMA) was published by Hussein, Saadawi and Lee in 2005 [35]. It is targeted at *MANETs* and *WSNs* and focuses on fair resource usage – esp. node energy – across the network. To achieve this, the forward ants carry not only source and destination address and intermediate node IDs but also quality information about the path. To prevent ants from growing too big, the path information is calculated as a normalized local index and computed into a cumulative path index as shown in Eqs. (8) and (9) below. The ant only carries the path index. This novel path index is the main contribution of the paper.

Let $p_{i,m}$ node i 's normalized optimization parameter m with $0 < p_{i,m} < 1$. This can be the number of hops, battery power, delay, bandwidth, etc. Then the local normalized index I_i for node i is

$$I_i = \sum_m a_m p_{i,m} \quad (8)$$

where a_m is the weight of this parameter with $\sum_m a_m = 1$. This leads to $0 \leq I_i \leq 1$. As the forward ant passes a node it updates the path information it carries by calculating the path index I_{path} as follows:

$$I_{path} = \prod_i I_i. \quad (9)$$

Since $0 \leq I_i \leq 1$ also $0 \leq I_{path} \leq 1$. A bottleneck link on the path correctly influences the overall path index as the value of I_{path} is smaller than the smallest I_i along

the path. When the forward ant reaches the destination, the path grade ρ is computed as

$$\rho = f(I_{path}, I_{path,best}) \quad (10)$$

where $I_{path,best}$ is the best I_{path} received in the last W number of ants (W a suitable window size).

With d the destination node, i the current node, and n the next hop node the **transition rule** used by ARAMA is given as

$$p_{d,i,n} = \begin{cases} \frac{fun(\tau_{d,i,n}, \eta_{i,n})}{\sum_{j \in N_i} fun(\tau_{d,i,j}, \eta_{i,j})}, & n \in N_i \\ 0 & n \notin N_i \end{cases} \quad (11)$$

where $\tau_{d,i,n}$ is the pheromone value for going from current node i to destination d via next neighbor n and $\eta_{i,n}$ is the local heuristic value of the link (i, n) . Function $fun(\tau_{d,i,n}, \eta_{i,n})$ is chosen to give a high function value when $\tau_{d,i,n}$ and $\eta_{i,n}$ are high; eg. as in the transition rule of *AntHocNet*.

When the backward ant traverses the network back to the source, the pheromones are updated with the **pheromone update rule**:

$$\tau_{d,i,n} := \begin{cases} f_{evap}(\rho_d)\tau_{d,i,n} + g_{enf}(\rho_d) & \text{if } n \in \text{Path} \\ f_{evap}(\rho_d)\tau_{d,i,n} & \text{if } n \notin \text{Path} \end{cases} \quad (12)$$

with $f_{evap}()$ the evaporation function, g_{enf} the enforcement function, and ρ_d the path grade for this path calculated from the information in the forward ant as shown above.

The authors also propose two very interesting extensions to the algorithm:

Negative Backward Ants are sent if a forward ant dies due to running out of TTL (time-to-live) or loop detection. In this case, a negative backward ant is sent which deemphasizes the path by decreasing its pheromone levels.

Destination Trail Ants implement the *RARE* (Receiver Assisted Routing Enhancement) concept by the same group (Abdelmalek, Hussein, and Saadawi [36]). With this technique, destination nodes send so called “destination trail ants” into the network which randomly mark paths leading to the destination. When forward ants search for this destination there is a probability that they will hit a destination trail left behind by a destination trail ant. This helps to speed up connection setup time.

3.4 AMQR

Ant colony based Multi-path QoS-aware Routing (*AMQR*) was developed for *MANETs* by Liu and Feng in 2005 [37]. It is based on *ARA* (introduced in Section 3.2). In contrast to *ARA* it supports link-disjoint multi-path routing.

Like *ARA* it uses the transition rule from *AntHocNet* with a factor $\beta_1 = 1$ (cf. Eq. (3)).

$$p_{i,n} = \begin{cases} \frac{\varphi_{i,n}}{\sum_{j \in N_i} \varphi_{i,j}}, & \text{if } n \in N_i \\ 0 & \text{if } n \notin N_i \end{cases} \quad (13)$$

with $p_{i,n}$ the probability to go from node n_i to n_n and N_i the set of neighbors of n_i in dependence of the pheromone value $\varphi_{i,n}$.

As in ARA, forward ants mark the trail back to the source as they move while backward ants update the trail to the destination. The pheromone values are updated as follows:

$$\varphi_{i,n} := (1 - \alpha) \cdot \varphi_{i,n} + \Delta\varphi_{i,n} \quad (14)$$

with α the pheromone decay parameter and $\Delta\varphi_{i,n}$ calculated as

$$\Delta\varphi_{i,n} = q^{-m} h^{-n} \quad (15)$$

where q is the delay time and h the hop count experienced by the ant so far. The parameters m and n are the weights that determine the relative importance of time delay and hop count.

Forward ants use a frame format of $\langle n_s, n_d, SeqN, HopC, [(PasN, ArrT), (.,.), \dots] \rangle$. where n_s is the ID of the source node, n_d the ID of the destination node, and $SeqN$ and $HopC$ the sequence number and hop count of the ant respectively. The list $[(PasN, ArrT), \dots]$ contains the IDs of the nodes $PasN$ passed by the ant and the relevant arrival times $ArrT$. Backward ants use the same frame format as forward ants but without $SeqN$ and $HopC$.

The routing table has the usual entries $\langle n_d, n_n, \varphi_{i,n} \rangle$ with n_d the destination, n_n the next hop and $\varphi_{i,n}$ the pheromone value for this link.

During route discovery, a source node first sends hello packets to determine its neighbors and then broadcasts a forward ant. Therefore, there is more than one copy of the forward ant in the system. When an intermediate node receives a forward ant more than once and the ant's hop count $HopC_{new} \leq HopC_{old} + \Delta hops$ another entry is made in the routing table to record this alternative path. Parameter $\Delta hops$ is the threshold for an acceptable additional path length to avoid overly long alternative paths. Backward ants always choose the best path back to the source.

Nodes exchange routing information by additional communication and build their own view of the topology, and only link-disjoint routes are used. The same concept is used in PPRA shown later (see Section 3.6). Load balancing and route failures are handled as in ARA.

To support QoS, nodes monitor their state and the delay recorded in the ants they receive. If the delay in an ant exceeds a certain limit, the pheromone for the respective link is set to 0 and the other pheromone values in the routing table are adjusted to eliminate this high-delay link. If the node itself is overloaded, it initiates a new backward ant to the source to change the route.

3.5 Scalable Ant-based Routing

Ohtaki et al. [38] focus on the scalability of ant-based routing for MANETs. Their algorithm is based on uniform ant routing [39] and borrows the TTL-limiting technique from HSLs (Hazy Sighted Link State) routing [40].

As in uniform ant routing, a probability routing table is kept at each node. For each entry (d, n) there exists a value p_{dn} which gives the probability of routing a packet destined for node d via neighboring node n . Nodes send periodic control messages (ants) of the form $\langle h_s, c, TTL \rangle$ which wander the network randomly. Here, h_s is the source address, c the cost of all links traversed so far, and TTL the remaining time-to-live. Whenever a node receives such a control message from a neighboring node l it updates its routing table as follows

$$p_{dn} = \begin{cases} \frac{p_{dn} + \Delta p}{1 + \Delta p}, & \text{if } n = l \\ \frac{p_{dn}}{1 + \Delta p} & \text{if } n \neq l \end{cases} \quad (16)$$

with $n \in N_i$ the set of neighbors of the current node i and Δp given by

$$\Delta p = \frac{k}{f(c)}, (k > 0) \quad (17)$$

where $f(c)$ is a function of the total cost c and k the so-called learning rate of the algorithm. It defines the weight of one ant and is generally less than 0.1.

As the number of nodes in the network increases, the number of ants goes up and becomes a burden on the network. To improve scalability, the Scalable Ant-based Routing algorithm borrows a technique from HSLS [40] to limit the *TTL* of the control packets. The *TTL* T_k of the k -th ant is calculated as

$$T_k = 2^{x_k+1} \quad (18)$$

where

$$x_k = \min (x_{max}, \max (x | k \equiv 0 \pmod{2^x})). \quad (19)$$

The authors suggest that x_{max} should be set to half the number of nodes in the network.

Another improvement of this algorithm is the novel ant migration scheme. Instead of a purely random walk, ants try to move as far away from the source as possible. The idea is that ants should not “waste” their *TTL* in the neighborhood but rather try to cover the whole network. They can find the “direction away from the source” by following those links which have a low probability as a way to the source in the routing table. Low. when an ant originated from source s was received from node m the probability q_j that node j will be chosen as next-hop node among all neighboring nodes except m is calculated as

$$q_j = \frac{\frac{1}{p_{sj}}}{\sum_{k=1}^n \frac{1}{p_{sk}} - \frac{1}{p_{sm}}} \quad (20)$$

to find the next link in direction away from the source. In this way, they use their *TTL* most efficiently to reach nodes as far away from the source as possible and get good coverage of the network.

3.6 PPRA

PPRA Prioritized Pheromone Aided Routing Algorithm was published by Jeon and Kesidis in 2005 [41]. It is a multipath routing algorithm that considers both energy and latency and supports dual-priority traffic. It is aimed at sensor networks (*WSNs*) and *MANETs* with battery constraints and based on *ARA* [34] (see also Section 3.2). Multipath routes are used to guard against route failures; in case of a route breaking, the already set-up backup route can be used without waiting for another route discovery phase. The multiple paths are also used for load balancing. As the primary path’s pheromones degrade, traffic switches to the alternate routes.

The routing tables at each node n_i consist of entries of the form $\langle n_d, n_n, \delta^i(d, n), e^i(d, n) \rangle$ or $\langle n_d, n_n, \partial^i(d, n), e^i(d, n) \rangle$ where n_d is the destination node, n_n the next hop node, $\delta^i(d, n)$ the TTL-pheromone or $\partial^i(d, n)$ the Delay-pheromone respectively, and $e^i(d, n)$ the Energy-pheromone described later.

During **route discovery** the forward ants are broadcasted. Like in source routing, they carry the source address and record all node addresses along the way. Duplicate forward ants are *not* discarded as in single path algorithms. Instead, they are used to set up alternative routes. Note that out of the paths found by the duplicate ants only those which are link-disjoint are kept. Once a forward ant reaches the destination, a backward ant is created, which takes the path found by the forward ant back to the source. For the measurement of energy and delay, periodic control packets are sent in addition to route discovery ants.

Pheromone types: There are three kinds of pheromones in PPRA: TTL-pheromone (δ), Delay-pheromone (∂), and Energy-pheromone (e). The algorithm has two variants. Variant 1 uses TTL-pheromone and Energy-pheromone, while Variant 2 uses Delay-pheromone and Energy-pheromone.

TTL-pheromone is used to express the distance in hops (iow. the time-to-live a packet traveling this path will use) and is calculated as

$$\delta^i(d, n) := \delta^i(d, n) + \beta_1 \cdot TTL(d, n) \quad (21)$$

with β_1 a scaling constant and $TTL(d, n)$ the number of hops between node d and node n . Evaporation for TTL-pheromone is calculated as

$$\delta^i(d, n) := \delta^i(d, n) \cdot \beta_2 \quad \text{with} \quad 0 < \beta_2 < 1. \quad (22)$$

For highly volatile networks a higher value of β_2 can be used to decay stale routes faster.

Energy-pheromone represents the battery status of the nodes in the path.

Similar to Eq. (21) it is calculated as

$$e^i(d, n) := e^i(d, n) + \alpha_1 \cdot E_{min}(d, n) \quad (23)$$

with α_1 a scaling constant. E_{min} represents the energy bottleneck on the path, i.e. the lowest battery level encountered in a node along the path. Evaporation for Energy-pheromone is calculated as

$$e^i(d, n) := e^i(d, n) \cdot \alpha_2 \quad \text{with} \quad 0 < \alpha_2 < 1. \quad (24)$$

Delay-pheromone marks the cumulative queuing delay experienced along a path. It is calculated analogously to TTL-pheromone as

$$\partial^i(d, n) := \partial^i(d, n) + \gamma_1 \cdot D(d, n) \quad (25)$$

with γ_1 a scaling constant and $D(d, n)$ the cumulative queuing delay between node d and node n . Evaporation for Delay-pheromone is calculated as

$$\partial^i(d, n) := \partial^i(d, n) \cdot \gamma_2 \quad \text{with} \quad \gamma_2 > 1. \quad (26)$$

The algorithm distinguishes between latency-critical and non-critical traffic. For latency-critical traffic, it always uses both pheromone levels (Energy- and TTL-pheromone or Energy- and Delay-pheromone respectively) to determine the route, for non-critical traffic only Energy-pheromone is considered.

The **transition rule for non-critical traffic** is given in Eq. (27).

$$p_e^i(d, n) = \frac{e^i(d, n)}{\sum_{j \in N_i} e^i(d, j)} \quad (27)$$

where N_i the set of neighboring nodes of i .

The **transition rule for latency-critical traffic** is calculated by combining Energy-pheromone and TTL-pheromone (algorithm variant 1) or Energy-pheromone and Delay-pheromone (algorithm variant 2) as shown in Eqs. (28)–(31) respectively.

$$\text{Variant 1 : } p_{lat}^i(d, n) = \frac{\theta \cdot p_e^i(d, n) + p_\delta^i(d, n)}{\sum_{j \in N_i} [\theta \cdot p_e^i(d, j) + p_\delta^i(d, j)]} \quad (28)$$

where

$$p_\delta^x(w, z) = \frac{\delta^x(w, z)}{\sum_{j \in N_x} \delta^x(w, j)} \quad (29)$$

and N_x the set of neighboring nodes of x .

$$\text{Variant 2 : } p_{lat}^i(d, n) = \frac{\theta \cdot p_e^i(d, n) + p_\delta^i(d, n)}{\sum_{j \in N_i} [\theta \cdot p_e^i(d, j) + p_\delta^i(d, j)]} \quad (30)$$

where

$$p_\delta^i(d, n) = \frac{1/\delta^i(d, n)}{\sum_{j \in N_i} 1/\delta^i(d, j)} \quad (31)$$

and N_i the set of neighboring nodes of i .

3.7 EEABR

Energy-Efficient Ant-Based Routing (*EEABR*) is a routing algorithm based on the *Ant Colony Optimization* metaheuristic. It was developed for WSNs by Camilo et al. in 2006 [42]. The major goal of this algorithm is to increase energy efficiency. The authors propose three algorithms, basic, improved, and energy-efficient ant routing.

$$p_k(r, s) = \begin{cases} \frac{[\tau(r, s)]^\alpha [E(s)]^\beta}{\sum_{u \notin M_k} [\tau(r, u)]^\alpha [E(u)]^\beta}, & \text{if } s \notin M_k \\ 0 & \text{if otherwise} \end{cases} \quad (32)$$

where an ant k chooses with probability $p_k(r, s)$ to move from node r to node s , $\tau(r, s)$ the amount of pheromone for link (r, s) , and $E(s)$ being the factor η in the *Ant Colony Optimization* metaheuristic. In this case, $E(s)$ is calculated from the initial energy level of the nodes C and e_s the actual energy level of the node by

$$E(s) = \frac{1}{C - e_s}. \quad (33)$$

The backward ant of forward ant k drops pheromone according to the pheromone update rule given in Eq. (34).

$$\tau_k(r, s) := (1 - \rho) \cdot \tau_k(r, s) + \Delta\tau_k. \quad (34)$$

Here, $(1 - \rho)$ represents the evaporation and $\Delta\tau_k$ is calculated from the total number of nodes N and the distance Fd_k traveled by forward ant k as

$$\Delta\tau_k = \frac{1}{N - Fd_k}. \quad (35)$$

The first improvement of this basic algorithm uses a refined function for calculating τ_k where the ant carries an energy vector. From this, the average energy level of the path is calculated when the backward ant is created. While this makes it possible to better monitor the energy level on the path it can lead to quite big forward ants. Since in WSNs communication costs much more energy than local calculation, the authors propose to save ant size by storing only the average (E_{avg_k}) and minimum energy (E_{min_k}) found on the path so far in the forward ant. The pheromone update rule in the final *EEABR* algorithm is then given as:

$$\tau_k(r, s) := (1 - \rho) \cdot \tau_k(r, s) + \left\lceil \frac{\Delta\tau_k}{\varphi B d_k} \right\rceil \quad (36)$$

where φ is a coefficient, Bd_k the traveled distance of the backward ant in hops and

$$\Delta\tau_k = \frac{1}{C - \left\lceil \frac{E_{min_k} - Fd_k}{E_{avg_k} - Fd_k} \right\rceil}. \quad (37)$$

Through factors φ and Bd_k the backward ant loses part of its pheromone strength while it travels back to the source – thereby giving shorter paths an advantage in the routing table.

The authors also reduce the memory M_k of already visited nodes in the forward ant to just the last two nodes visited. This means no full path information is stored in the ant anymore, further reducing its size to achieve a so-called “light-weight” ant. The tasks of loop detection and remembering the path back to the source now fall to the nodes themselves. Nodes keep track of the forward ants using a structure $\langle n_p, n_s, ant_{ID}, t \rangle$ where n_p is the previous node, n_s the next (forward) node, ant_{ID} the ID of the ant and t a timeout value.

When a forward ant is received, a node checks the table whether this ant has been received before. If yes, the ant is discarded as a loop was detected. If no, the ant is forwarded according to the transition rule. When the backward ant returns to the node, it looks up the way back to the source in this same table. The timeout timer t controls how long the node keeps the entry in the table. This also determines the maximum time a backward ant may take to come back via this node.

3.8 DDCHA

The Distributed, Data-Centric, Hierarchical Ant algorithm (*DDCHA*) is a combination of a data-centric protocol with ants developed for WSNs. To aggregate data, the sensor field is divided into subnets where the biggest distance between nodes is still within communication range. Nodes are location-aware and join a subnet based on their location relative to the sink. In each subnet, a core head and a

gateway are chosen with the Distributed Energy-Core Generating Algorithm (DECGA) described in the same paper as follows:

1. Initially, the sink node is the core head, and all sensor nodes are member nodes.
2. Every node exchanges information about its function (core head, gateway, member) and energy level with its neighbors periodically.
3. In every period, every node p computes its new state
 - If there is no core head in a subnet, then the node with the largest surplus energy becomes the core head.
 - If p is neither a core head nor gateway but neighbor with at least one node of a different subnet then p becomes a gateway.

The authors prove that this generates an energy-core Ψ in the network graph.

Routing is done with the *DDCHA* ant algorithm on top of this network structure as follows: initially, all pheromone values are 0. Every core head can be seen as an ant nest (source) which sends forward ants towards the sink of the WSN. Forward and backward ants both mark their path with pheromones immediately. Unlike *ARA* (and closer to ant behavior in nature), the ants do not mark the path back to the source/destination but simply drop a fixed amount $+\Delta$ of pheromone on the forward path. Ants also do not follow a probabilistic routing table but choose the path with the highest amount of pheromone. If an ant can not move on anymore (i.e., it got caught in a loop) it backtracks its path, decreasing the amount of pheromone by $-\Delta$ on the way until it finds a new path. Loop detection is achieved by keeping a forbidden-list of already visited nodes in the ant. Each member node sends its data to the core head first which aggregates the data. The core head then sends the data onto the sink of the WSN.

3.9 Approaches using colored pheromones

Several algorithms are known which make use of “colored pheromones”. This means that trails can be distinguished not only by the amount of pheromone dropped but also by the “color” of the pheromone. In the following, we introduce three approaches for *MANETs*, *WSNs*, and *WMNs* respectively.

3.9.1 MACO

The original *Ant Colony Optimization* metaheuristic has some drawbacks in terms of stagnation and adaptiveness. Stagnation occurs when a network reaches convergence, and an optimal path is found and chosen by all ants. This, in turn, reinforces this path so much that the probability of selecting other paths becomes very low, which can lead to congestion on the “optimal” path. Adaptiveness describes the ability of an algorithm to react to changes in the network. *MACO* (Multiple Ant Colony Optimization) was developed to mitigate these problems in *MANETs* by Sim and Sun in 2002 [43].

MACO uses several ant colonies in parallel, which each use their own color of pheromone. The colonies are entirely separated and cannot sense pheromone other than that of their own color.

Similar to nature, the forward ants immediately drop pheromone on the paths they take. I.e., a “red” forward ant A^r will drop $+\tau_r$ on the forward direction of a link it chooses. The backward ant inherits the color from the forward ant and chooses the path back to the source with the highest amount of pheromone in its respective color. Backward ants also drop additional pheromone on the link on their way back.

Depending on the number of ant colonies used in parallel, several paths from the source to the sink can be found. These paths can then be used as alternative routes for load-balancing. Consider the following example with two ant colonies, “red” and “blue”. Lets assume that there are three paths through the network, one long route R_1 and two similarly short (=good) ones R_2, R_3 with $R_1 > R_2 \approx R_3$. With just one ant colony, the true minimum route would be chosen as the optimal path. With two ant colonies, there is a possibility that one colony will find R_2 as the shortest path and the other colony R_3 . In this case, two alternative paths of similar quality were found. Therefore, MACO increases the probability of exploring alternatives.

3.9.2 Division of labour in SANETS

Wireless SANETs (Sensor/Actuator Networks) are a form of WSN which also contains actuators (eg. robots). The same energy constraints as in WSNs apply.

Labella and Dressler develop in [44] an ant-based algorithm for division of labour and the routing of the respective traffic in SANETs. In their model, nodes can perform different tasks (measurement of temperature, recording of sound, recording of video, and movement). The goal is to distribute the tasks evenly in the network to get good measurement coverage and not overload single paths with high-load communications (video and audio transmissions).

Nodes choose tasks by employing the *AntHocNet* transition rule. The probability for a node to choose task i from the task list T_{agent} is

$$P(i) = \frac{\tau_i^{\beta_{\text{task}}}}{\sum_{k \in T_{\text{agent}}} \tau_k^{\beta_{\text{task}}}}. \quad (38)$$

Pheromones are updated depending whether the task could be completed successfully or not by

$$\tau_i := \min \tau_{\max}, \tau_i + \Delta\tau \quad (39)$$

if successful and

$$\tau_i := \max \tau_{\min}, \tau_i - \Delta\tau \quad (40)$$

if not.

Since tasks are inherently linked with the traffic they generate (simple temperature values, sound, video, or command traffic for movements) tasks imply traffic classes. To deal with these different classes, the authors employ colored pheromones.

Each node i keeps separate routing tables ${}_c\mathbf{R}^i$ for each color c . Each entry ${}_c\mathbf{R}_{nd}^i$ for going from node i via node n to destination d is of the form $\langle t, h, e, s, m, v \rangle$ where t is an estimation of the transmission time, h the number of hops in the route, e the energy required for transmission, s the minimal signal-to-noise ratio on the path, and m and v are flags that indicate whether a node is mobile and still valid for routing.

The transition rule is taken from *AntHocNet* and evaluated for each color:

$$\mathcal{P}_{nd}^i = \frac{r(c\mathbf{R}_{nd}^i)^\beta}{\sum_{j \in N_d^i} r(c\mathbf{R}_{nd}^j)^\beta} \quad (41)$$

where N_d^i is the set of neighbors for which a path to d is known. A different value for β is used during route discovery and data routing. $r(\cdot)$ is a function $r : \mathbf{R} \rightarrow \mathbb{R}^+$ which evaluates the link statistics.

The algorithm also employs an elaborate probabilistic scheme to filter packets and deliberately drop them to avoid congestion (eg. if a measurement value did not change).

3.10 Other Ant-based Algorithms and Techniques

Other ant-based algorithms for Wireless Multi-Hop Networks include the following:

AARAI: Ant Colony Algorithm with Adaptive Improvement was developed by Zeng and He in 2005 [45]. It is targeted at *MANETs*, based on *Ant Colony Optimization* and implements multipath routing.

IAQR: Improved Ant colony QoS Routing, developed by Liu et al. in 2007 [46] is based on *Ant Colony System* [31] and has the goal to improve QoS routing in *MANETs*. The transition rule is taken from *AntHocNet*. It measures each link delay, bandwidth, jitter, and cost and for each node queueing delay, packet loss, jitter, and cost. It uses a global update rule for QoS optimization which changes the decay factor of the pheromone to account for link quality.

Ant-AODV: Ant-Ad Hoc On-Demand Distance Vector Routing by Marwaha et al. [47] combines AODV [12] with ants. It is targeted at *MANETs*. Ants are used during the route discovery phase to reduce route discovery latency.

Zhang, Kuhn, and Fromherz [48] introduce three new ant routing algorithms for *WSNs*: Sensor-driven and Cost-aware ant routing (SC), Flooded Forward ant routing (FF), and Flooded Piggybacked ant routing (FP). In SC, they introduce the concept of “sensing ants” which can “smell” other nodes from afar. This is facilitated by exchange of cost estimates from neighboring nodes via HELLO-messages. In FF, forward ants are flooded using the broadcast channel of the *WSN* and in FP forward ants and data ants are combined to save communications overhead.

Other techniques use ants not directly to perform routing but to support the actual routing algorithm. Examples of these are described in the following.

GPSAL: GPS/Ant-Like Routing Algorithm by Câmara and Loureiro [49, 50] is a location based routing algorithm designed for *MANETs* and *WMNs*. Its goal is to use location information to reduce the number of routing messages and speed up route recovery. The assumption is that nodes are equipped with location finding devices like GPS (Global Positioning System). Nodes n_i keep routing tables which contain $\langle n_d, loc_{curr}(d), loc_{prev}(d), TTL(loc_{prev}(d)), type(d) \rangle$ where n_d is the destination node, $loc_{curr}(d)$ the current location of n_d , $loc_{prev}(d)$ the previous location of n_d , $TTL(loc_{prev}(d))$ the time-to-live of the previous location of n_d , and $type(d)$ the mobility type (mobile, stationary) of n_d . Ants are only used to collect and seminate location information to the routing tables of the nodes. The actual routing is then calculated on the location entries using a shortest-path algorithm.

T-ANT was developed in 2006 by Selvakenedy et al. [51]. It uses ants for cluster head election in WSNs. The algorithm uses elements from *Scalable Ant-based Routing* [38] described before (see Section 3.5). The actual routing is performed using a greedy routing scheme.

Ant-aggregation was developed by Misra and Mandal in 2006 [52] for use in WSNs. Here, ants are not used for routing but to determine the optimal in-network data-aggregation points (the optimal nodes for data-aggregation in the WSN). The algorithm is based on *Ant Colony Optimization*.

4. Conclusions

The versatile and dynamic nature of Wireless Multi-Hop Networks requires routing algorithms that are robust, adaptable, and scalable. Ant Algorithms are inspired from the self-organizing foraging behavior of natural ants, which show an incredible ability for robustness, adaptation and scalability despite being based on a set of simple mechanisms. In this chapter, we have first reviewed the seminal ant routing algorithm developed for routing in such networks, *AntHocNet*. In the following, we investigate more specific algorithms: *ARA* is a simple version of an ant colony optimization approach for routing in *MANETs*. *ARAMA* is targeted at *MANETs* and *WSNs* and focuses on fair energy use between the nodes of the network. *EEABR* is another algorithm focusing on energy efficiency, providing a more fine-grained selection of routing mechanisms. *DDCHA* is a data-centric protocol which divides a sensor field into subnets of nodes within communication range. *AMQR* is a routing algorithm for *MANETs* that extends upon *ARA*. A concept for dual-priority traffic, together with a notion of energy and latency constraints, is reported in the *PPRA* algorithm. To match requirements of different traffic types, we have also reviewed approaches using colored pheromones – here the colored pheromones form separate routing layers representing different route properties such as latency, jitter, or bandwidth. Two representatives of this class of algorithms are *MACO* and *SANETs* have been reviewed in this chapter.

Ant-inspired algorithms can be successfully applied for routing in Wireless Multi-Hop Networks, but due to the difficulty of the problem and different requirements priorities among Wireless Multi-Hop Networks, they have not converged into a single solution. Instead, we are facing an increasing number of algorithms and protocols following this idea. A short insight into this is given in the section on other ant-based algorithms for Wireless Multi-Hop Networks. Considering the constant change of sensor network hardware and software together with probably slightly different requirements, we are expecting this trend to continue and foresee new ant routing algorithms in the future.

Acknowledgements

The publication of this chapter was supported by the University of Klagenfurt.

IntechOpen

Author details

Martina Umlauf¹ and Wilfried Elmenreich^{2*}

¹ Lakeside Labs, Klagenfurt, Austria

² Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria

*Address all correspondence to: wilfried.elmenreich@aau.at

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann, 2014.
- [2] Y. Toor, P. Muhlethaler, A. Laouiti, and A. D. La Fortelle. Vehicle ad hoc networks: Applications and related technical issues. *IEEE Communications Surveys Tutorials*, 10(3):74–88, 2008.
- [3] V. Rastogi, V. J. Ribeiro, and A. D. Nayar. Measurements in olpc mesh networks. In *2009 7th international symposium on Modeling and optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 1–6, 2009.
- [4] freifunk.net. <http://start.freifunk.net/>, last visited: Dec. 2010.
- [5] FunkFeuer Wien – Verein zur Förderung freier Netze (ZVR: 814804682). 0xFF – FunkFeuer Free Net. <http://funkfeuer.at/>, last visited: Dec. 2010.
- [6] W. Elmenreich and H. de Meer. Self-organizing networked systems for technical applications: A discussion on open issues. In J.P.G. Sterbenz. K.A. Hummel, editor, *Proceedings of the Third International Workshop on Self-Organizing Systems*, pages 1–9. Springer Verlag, 2008.
- [7] Charles E. Perkins. *Ad Hoc Networking*. Addison Wesley, 2001. ISBN 0-201-30976-9.
- [8] Ian F. Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: A survey. *Computer Networks*, 47(4), March 2005.
- [9] David B. Johnson. Routing in ad hoc networks of Mobile hosts. In *proceedings of the workshop on Mobile computing systems and applications*, pages 158–163, Santa Cruz, CA, December 1994. IEEE Computer Society.
- [10] David B. Johnson, David A. Maltz, and Yih-Chun Hu. The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR). Ietf internet draft, IETF, July 2004. draft-ietf-manet-dsr-10.txt, work in progress.
- [11] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.
- [12] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, February 1999.
- [13] Philippe Jacquet, Paul Muhlethaler, Thomas Clausen, Anis Laouiti, Amir Qayyum, and Laurent Viennot. Optimized Link State Routing Protocol for Ad hoc Networks. In *INMIC 2001*, Pakistan, 2001.
- [14] Thomas Clausen and Philippe Jacquet, editors. *Optimized Link State Routing Protocol (OLSR)*. IETF, October 2003. IETF Experimental RFC 3626.
- [15] David Culler, Deborah Estrin, and Mani Srivastava. Overview of sensor networks. *IEEE computer*, pages 41–49, august 2004. Special Issue in Sensor Networks.
- [16] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, pages 102–114, August 2002.
- [17] Kay R"omer and Friedemann Mattern. *The Design Space of Wireless*

Sensor Networks. IEEE Wireless Communications, 11(6):54–61, December 2004.

[18] Salem Hadim and Nader Mohamed. Middleware Challenges and Approaches for Wireless Sensor Networks. IEEE Distributed Systems Online, 7(3), 2006. art. no. 0603-o3001.

[19] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman. A survey of gossiping and broadcasting in communication networks. Networks, 18:319–349, 1988.

[20] W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In ACM MobiCom’99, pages 174–185, Seattle, WA, USA, 1999.

[21] Brad Karp. Geographic Routing for Wireless Networks. PhD thesis, Harvard University, Cambridge, MA, USA, October 2000.

[22] Brad Karp and H. T. Kung. Greedy Perimeter Stateless Routing for Wireless Networks. In Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000), Pages 243–254, Boston, MA, USA, August 2000.

[23] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In 33rd Annual Hawaii International Conference on System Sciences, pages 1–10, January 2000.

[24] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. Protocols for self-Organization of a Wireless Sensor Network. IEEE Personal Communications, 7(5):16–27, October 2000.

[25] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication

Paradigm for Sensor Networks. In ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’00), pages 56–67, Boston, MA, USA, 2000.

[26] Yan Zhang, Jijun Luo, and Honglin Hu, Editors. Wireless Mesh Networking, Architectures, Protocols and Standards. Auerbach Publications, 2007. ISBN 0-8493-7399-9.

[27] Jim Hauser, Dennis Baker, and W. Steven Conner. Draft PAR for IEEE P802.11 ESS Mesh. Technical report, IEEE, January 2004. IEEE P802.11 Wireless LANs, document 11-04/0052r2.

[28] S. Goss, S. Aron, J. L. Deneubourg, and J.M. Pasteels. Self-organized shortcuts in the argentine ant. Naturwissenschaften, 76:597–581, 1989.

[29] M. Dorigo and T. Stützle. Ant Colony Optimization. A Bradford Book, The MIT Press, 2004.

[30] M. Dorigo, V. Maniezzo, and A. Coloni. The ant system: Optimization by a Colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1):1–13, 1996.

[31] M. Dorigo and L. M. Gambardella. Ant Colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1(1):53–66, April 1997.

[32] G. D. Caro and M. Dorigo. AntNet: Distributed Stigmergy control for communications networks. Journal of Artificial Intelligence Research (JAIR), 9:317–365, 1998.

[33] Gianni Di Caro, Frederick Ducatelle, and Luca Maria Gambardella. Anthocnet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks. European Transactions on

Telecommunications, 16(5):443–455, 2005.

[34] Mesut Günes, Udo Sorges, and Imed Bouazizi. ARA – The Ant-Aolony Based Routing Algorithm for MANETs. In Stephan Olariu, editor, *International Workshop on Ad Hoc Networking (IWAHN 2002)*, Pages 79–85, Vancouver, British Columbia, Canada, August 18–21 2002. IEEE Computer Society Press.

[35] Osama H. Hussein, Tarek N. Saadawi, and Myung J. lee. Probability routing algorithm for Mobile ad hoc networks. *Journal in selected areas in communications*, 23(12):2248–2259, 2005. IEEE.

[36] Y. Abdelmalek, O. Hussein, and T. Saadawi. A Simplified Receiver Assisted Routing Enhancement (RARE) in MANET. In *BIONETICS*, Budapest, Hungary, December 10–13 2007.

[37] Langgui Liu and Guangzeng Feng. A Novel Ant Colony Based QoS-Aware Routing Algorithm for MANETs. In L. Wang, K. Chen, and Y. S. Ong, editors, *Advances in Natural Computation (ICNC 2005)*, Lecture notes in computer science, LNCS 3612, pages 457–466. Springer, 2005.

[38] Yoshitaka Ohtaki, Naoki Wakamiya, Masayuki Murata, and Makoto Imase. Scalable ant-based routing algorithm for ad-hoc networks. *IEICE Transactions on Communications*, 89(4):1231–1238, 2006. ISSN 0916-8516.

[39] D. Subramanian, P. Druschel, and J. Chen. Ants and reinforcement learning: A case study in routing in dynamic networks. Technical Report tr96-259, Rice University, July 1998.

[40] C. A. Santiv  nez, R. Ramanathan, and I. Stavrakakis. Making link-state routing scale for ad hoc networks. In *2nd ACM International Symposium on*

Mobile ad hoc networking and computing, pages 22–32, October 2001.

[41] Paul Barom Jeon and George Kesidis. Pheromone-Aided Robust Multipath and Multipriority Routing in Wireless MANETs. In *2nd ACM Intl. Workshop on Performance Evaluation of wireless ad hoc, sensor and ubiquitous networks (PE-WASUN)*, pages 106–113, Montreal, Quebec, Canada, October 10–13 2005.

[42] Tiago Camilo, Carlos Carreto, Jorge S   Silva, and Fernando Boavida. An energy-efficient ant-based routing algorithm for wireless sensor networks. In Marco Dorigo, Luca Maria Gambardella, Mauro Birattari Alcherio Martinoli, Riccardo Poli, and Thomas St  tzle, editors, *5th Intl. Workshop on ant Colony optimization and swarm intelligence (ANTS)*, lecture notes in computer science, LNCS 4150, pages 49–59, Brussels, Belgium, September 4–7 2006. Springer.

[43] Kwang Mong Sim and Weng Hong Sun. Multiple Ant-Colony Optimization for Network Routing. In *Proc. 1st Intl. Symposium on Cyber Worlds (CW’02)*, pages 277–281, 2002.

[44] Thomas Halva Labella and Falko Dressler. A Bio-Inspired Architecture for Division of Labour in SANETs. In *Proc. 1st IEEE ACM Intl. Conference on Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS’06)*, Cavalese, Italy, Dec. 2006.

[45] Yuan-yuan Zeng and Yan-xiang He. Ant Routing Algorithm for Mobile Ad-hoc Networks Based on Adaptive Improvement. In *IEEE Wireless Communications Networking and Mobile Computing (WiCOM)*, volume 2, pages 678–681, Wuhan, China, September 23–26 2005.

[46] Ming Liu, Yange Sun, Rui Liu, and Xiaoyan Huang. An Improved Ant

- Colony QoS Routing Algorithm Applied to Mobile Ad Hoc Networks. In IEEE Wireless Communications, Networking and Mobile Computing (WiCOM), pages 1641–1644, Shanghai, China, September 21–25 2007.
- [47] Shivanajay Marwaha, Chen Khong Tham, and Dipti Srinivasan. Mobile Agents based Routing Protocol for Mobile Ad Hoc Networks. In IEEE Global Telecommunications Conference (GLOBECOM'02), volume 1, pages 163–167, Taipei, Taiwan, Nov. 17–21 2002.
- [48] Ying Zhang, Lukas D-Kuhn, And Markus P. J. Fromherz. Improvements on ant routing for sensor networks. In Marco Dorigo, Mauro Birattari, Christian Blum, Luca M. Gambardella, Francesco Mondada, and Thomas Stützle, editors, 4th Intl. Workshop on ant Colony optimization and swarm intelligence (ANTS), lecture notes in computer science, LNCS 3172, pages 154–165, Brussels, Belgium, September 5–8 2004. Springer.
- [49] Daniel Câmara and Antonio Alfredo F. Loureiro. A Novel Routing Algorithm for Ad Hoc Networks. In 33rd Hawaii International Conference on System Sciences, Hawaii, January 4–7 2000.
- [50] Daniel Câmara and Antonio Alfredo F. Loureiro. A novel routing algorithm for hoc networks. *Baltzer Journal of Telecommunications Systems*, 18(1–3):85–100, 2001. Kluwer Academic Publishers.
- [51] S. Selvakenedy, S. Sinnappan, And Yi Shang. T-ANT: A nature-inspired data gathering protocol for wireless sensor networks. *Journal of communications*, 1(2):22–29, may 2006. Academy Publisher.
- [52] Rajiv Misra and Chittaranjan Mandal. Ant-aggregation: Ant Colony Algorithm for optimal data aggregation in Wireless Sensor Networks. In IFIP Conference on Wireless and Optical Communications Networks, Bangalore, India, April 11–13 2006.