Smart Routing with Learning-based QoS-aware Meta-strategies

Ying Zhang[1], Markus P.J. Fromherz[1], and Lukas D. Kuhn[2]

[1] Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA
tel: 1-650-812-4654, fax: 1-650-812-4334
{yzhang,fromherz}@parc.com
[2] Ludwig Maximilian University, Munich, Germany

Abstract. Conventional Quality of Service (QoS) routing cannot be applied easily to wireless ad-hoc sensor networks due to the unreliable and dynamic nature of such networks. For these networks, we have proposed a framework of Message-initiated Constraint-Based Routing (MCBR), which consists of a QoS specification and a set of QoS-aware meta-strategies. In contrast to most existing ad-hoc routing with no QoS support, MCBR is able to take QoS specifications into account. In this paper, we focus on learning-based meta-strategies. In contrast to most existing QoS routing approaches, learning-based meta-strategies do not create and maintain explicit routes; instead, packets discover and improve the routes during the search for the destination.

keywords Mobile and Wireless Networks, Ad-hoc Sensor Networks, Meta-strategies, Reinforcement Learning

1 Motivation and Introduction

Large-scale ad-hoc networks of wireless sensors have become an active topic of research. Such networks share the following properties:

- *embedded routers* – each sensor node acts as a router in addition to sensing the environment;
- *dynamic networks* – nodes in the network may turn on or off during operation due to unexpected failure, battery life, or power management; attributes associated with those nodes (locations, sensor readings, load, etc.) may also vary over time;
- *resource constrained nodes* – each sensor node tends to have small memory and limited computational power;
- *dense connectivity* – the sensing range in general is much smaller than the radio range, and thus the density required for sensing coverage results in a dense network;
- *asymmetric links* – the communication links are not reversible in general.

Applications of sensor networks include environment monitoring, traffic control, building management, object tracking, etc. Routing in sensor networks, however, has very different characteristics than routing in traditional communication networks. First of all, address-based destination specification is replaced by a more general feature-based specification, such as geographic locations [4] [13] or information gains [3]. Secondly, routing metrics are not just shortest delay, but usually multiple objectives, including energy usage and information density. Thirdly, in addition to peer-to-peer communication, multicast (one-to-many) and converge-cast (many-to-one) are major traffic patterns in sensor networks. Even for peer-to-peer communication, the source/destination pairs often are dynamic (changing from time to time) or mobile (moving during routing).

Various routing mechanisms have been proposed and implemented for sensor networks or wireless ad-hoc networks in general [7]; however, most of them do not have Quality of Service (QoS) support. Distributed QoS routing strategies for mobile ad-hoc networks have also been proposed [2] [1], all of which, however, like most other routing strategies, first establish routes between the source and the sink and then follow up with a route maintenance phase if the route is broken.

We have proposed Message-initiated Constraint-Based Routing (MCBR) [14] for wireless ad-hoc sensor networks. MCBR is a framework of routing mechanisms composed of the explicit specification of constraint-based destinations, route constraints and QoS requirements for messages, and a set of QoS-aware meta-strategies. With the separation of routing specifications from routing strategies, general-purpose *meta* routing strategies can be applied. In contrast to most existing ad-hoc routing strategies with no QoS support, MCBR takes QoS specifications into account. In this paper, we focus on a set of *distributed* routing strategies based on real-time reinforcement learning [11]. In particular, three types of meta-strategies are proposed: real-time search, constrained flooding, and adaptive spanning tree. All of these use the same reinforcement learning core, which estimates and updates the cost from the current node to the destination. The first two strategies have been presented elsewhere [14], while the last one is newly added to this family. The contribution of this paper is twofold: first, the use of MCBR for QoS specification, and second, the introduction of learning-based meta-strategies. The performance evaluations of these strategies and comparisons to AODV [6] are presented as well, using a real application scenario for sensor networks.

The rest of the paper is organized as follows. Section 2 introduces MCBR for QoS routing. Section 3 presents three QoS-aware learning-based meta-strategies. Section 4 discusses performance evaluations for these strategies. Section 5 concludes the paper and points out future directions.

2 MCBR for QoS Routing

MCBR [14] provides a general, flexible, and compositional mechanism for providing QoS message specification and QoS-aware meta-strategies. An MCBR message specification consists of a destination constraint, a route constraint, and a QoS routing objective. An MCBR meta-strategy is QoS-aware using the message specification. This is along the line of Smart Packets for Active Networks [9]; however, in MCBR, packets do not carry code. Only the specification (and possibly an additional selection of a particular meta-strategy) is passed through the network. For networks with small data frames, one can even encode various specifications in nodes and let packets only carry a specification ID with parameters.

A network can be represented as a graph $\langle V, E\rangle$, where V is the set of nodes and E is the set of connections. For an asymmetric network, $(v, w) \in E$ does not imply $(w, v) \in E$. Given a *destination constraint* C_m^d of message m, a node v is a *destination node* for m iff C_m^d is satisfied at v. For example, address-based routing, i.e., sending a message to a node with an address a_d, can be represented using the destination constraint $a = a_d$, where a is the address attribute. Geographical routing, e.g., sending a message to a circular region centered at (x_0, y_0) with radius c, can be represented using the destination constraint $(x - x_0)^2 + (y - y_0)^2 \leq c$, where x and y are location attributes.

Given a *local route constraint* C_m^r of message m, the *active* network of $\langle V, E\rangle$ for m is a subnet $\langle V_m, E_m\rangle$, such that $v \in V_m$ iff C_m^r is satisfied at v and $(v, w) \in E_m$ iff $v, w \in V_m$ and $(v, w) \in E$. For example, a message that should avoid congested nodes while routing to its destination has a local route constraint $l \leq l_m$, where l is the message load attribute (e.g., number of messages in the node's queue) and l_m is the load limit. One can also use geographical constraints (e.g., directional routing) to reduce collision and save energy for a flooding-based strategy. In general, local route constraints redefine the network connectivity on a message-by-message basis.

MCBR explicitly specifies routing objectives. A *local objective function* o is defined on a set of attributes: $o : A_1 \times A_2 \times \ldots \times A_n \rightarrow R^+$, where A_i is the domain of attribute i and R^+ is the set of *positive* real numbers. The *value* of o at a node v, denoted $o(v)$, is $o(a_1, a_2, \ldots, a_n)$, where a_i is the current attribute value of attribute i at node v. A local objective function can be a constant such as the unit transmission cost, which induces the shortest path if the objective is minimized. For another example, an energy-aware objective can be defined as $ku + c$, where u is the amount of used energy in the node, and k and c are constants. With this objective, energy-aware routing can be achieved. Similarly, one may use $k/n+c$ as a local objective, where n is the number of neighbors. With this objective, connectivity-aware routing can be achieved. Multi-objectives can be obtained by combining individual objectives, e.g., in a weighted sum.

A local objective can be aggregated over the routing path to form a global routing objective. There are two types of global aggregation, additive and concave. Like general QoS specifications, a *global objective function* O is *additive* if $O(p) = \sum_{i=0}^{n} o(v_i)$, where o is a local objective function and p consists of a sequence of nodes $v_0, \ldots, v_n$; For the meta-strategies discussed in this paper, only additive objectives are considered.

Problems for MCBR tend to be in one of two classes. One is *anycast*, namely finding an optimal path from the source to *one* of the destination nodes. The other is *multicast*, namely finding an optimal tree from the source to *all* the destination nodes.

An MCBR *specification* for a message m is a tuple $\langle v_m^0, C_m^d, C_m^r, O_m \rangle$. The *goal* of routing is to deliver the message from v_m^0 to one (anycast) or all (multicast) of the destination nodes V_m^d satisfying C_m^d via a sequence or a tree of intermediate nodes $p : v_m^1, \dots, v_m^{n-1}$ such that C_m^r is satisfied at v_m^i and $\min_p O_m(p)$. Two messages are considered to have the same *type* if they have the same destination and local route constraint as well as the same routing objective.

One should notice that global route constraints are not defined in MCBR. It is well-known that finding an optimal path with an additive objective while satisfying an additive constraint is NP-hard. Unicast MCBR with an additive objective is essentially a *weighted shortest path problem*. Our goal is to make MCBR a simple (in terms of computation) yet still powerful (in terms of representation) mechanism for ad-hoc sensor networks.

3 QoS-aware Learning-based Meta-strategies

MCBR separates routing specification from routing meta-strategies. One can modify an existing routing strategy, such as AODV, to be a QoS-aware meta-strategy for MCBR. Most existing strategies, however, establish a route from the source to the destination via flooding the network. In this case, extra control packets are required for repairing broken routes.

Here, we propose QoS-aware learning-based meta-strategies. Real-time reinforcement learning [11] has been studied and applied mostly in agent-based path planning [5]. We apply this powerful technique to develop *distributed* meta routing strategies for sensor networks.

Given a routing specification of a message, including the destination and QoS requirements, one can define a cost function on each node, called *Q-value*, indicating the minimum cost-to-go from this node to the destination. For a distributed sensor network, the cost is initially unknown, and an initial estimation is made according to the type of message. Furthermore, a node also stores its neighbors' Q-values, *NQ-values*, which are estimated initially according to the neighbors' attributes and updated when packets are received from neighbors.

The learning-based meta routing strategies typically consist of an *initialization* phase, a *forwarding* phase, and a *confirmation* phase. Learning happens in all phases. For each packet sent out from a node, the current Q-value of the node for the type of message is attached. All the nodes are set to be in promiscuous listening mode. Whenever a node overhears a packet of type m, whether it is the designated receiver or not, it updates the corresponding NQ-value and re-estimates its own Q-value using the equation

$$Q_m = (1 - \alpha)Q_m + \alpha(o_m + \min_n NQ_m(n)) \tag{1}$$

where α is a learning rate, o_m is the current value of the local objective function, and n is a neighbor of this node.

Using the Q-value, *real-time search* passes the packet to the "best" neighbor according to the estimates, *constrained flooding* decides if and when to re-broadcast the packet according to the cost estimates, and *adaptive spanning tree* forwards the packet to its parent, with parents possibly changing over time pointing to a neighbor with the best Q-value. This approach has a number of attractive properties: (1) explicit use of destination and QoS specifications for finding optimal routes; (2) automatic adaptation with different routes when network conditions change; (3) no need for extra maintenance packets; and (4) no *infinite* looping if a path to the destination exists.

3.1 Meta-strategy 1: real-time search

The pseudo code of real-time search is illustrated in Figure 1, where $Q_m^0(n)$ is an initial estimate for node n, according to the destination and QoS requirement of the message and the attribute values of this node [15]. Please note that although this strategy is *infinite loop* free, it is not loop-free. However, it has been proved that the maximum path length is $O(N^2)$ and it will converge to the optimal path in $O(N^2)$, where N is the number of nodes in the network. For space limitations, readers are referred to [15] for the theoretical bounds and variations of this meta-strategy.

3.2 Meta-strategy 2: constrained flooding

In contrast to the search-based methods, where each node decides which of the neighboring nodes to forward the message to, flooding-based strategies decide whether or not to broadcast at each node. A few gradient-flooding type strategies have been developed [12], requiring a cost field to be established beforehand or specialized for geographical routing. We propose a constrained-flooding meta-strategy, where the cost, i.e., the Q-value, can be learned if not known a priori. Figure 2 illustrates the basic idea. Like other gradient-flooding routing protocols [12], the cost is transmitted together with the packet. In addition, the cost at each node is updated every time a packet is received. The update rule is the same as for search-based strategies. Two techniques are used here to control the flood: (1) cost difference – if the receiving node estimates a significantly higher cost than the transmitting node, no action is taken except for updating its cost field; and (2) time difference – the transmit time difference is added to the broadcast, so that nodes with better estimates transmit first, while duplicate packets are suppressed. In this algorithm, a "temperature" variable T is used to control the flood: the higher T, the higher the chance that a packet is broadcast.

If the destination is known a priori, which turns out to be the case for many routing applications in sensor networks, backward constrained flooding from the destination can be used initially to establish the cost field. If there is no initialization and the cost field is flat, one can set T high initially and let it cool down when the cost field is more settled to reduce collisions and save energy. This strategy has been briefly discussed in [14].

Forwarding phase:

```
received (m, Q) at w from node u do
  if new(m) then
    for all v with (v,w) ∈ E_m do NQ_m(v) ← Q^0_m(v); end
    Q_m ← Q^0_m(w);
  end
  if satisfied(C^d_m) then Q_m ← 0; broadcast(m, 0); return; end
  NQ_m(u) ← Q;
  Q_m ← (1 − α)Q_m + α(o_m + min_v NQ_m(v));
  if designated(m) then
    v ← argmin_n NQ_m(n); (random tie break)
    send(m, Q_m) to v;
  end
end
```

Confirmation phase:

```
timeout (m to v)
  NQ_m(v) ← max_n NQ_m(n) + 1;
  if (resend) then
    v ← argmin_n NQ_m(n); (random tie break)
    send(m, Q_m) to v;
  end
end
```

Fig. 1. Real-time search meta-strategy

Forwarding phase:

```
received (m, Q) at w from node u do
  if new(m) then
    for all v with (v,w) ∈ E_m do NQ_m(v) ← Q^0_m(v); end
    Q_m ← Q^0_m(w);
  end
  if satisfied(C^d_m) then Q_m ← 0; broadcast(m, 0); return; end
  NQ_m(u) ← Q;
  Q_m ← (1 − α)Q_m + α(o_m + min_v NQ_m(v));
  if m is in transmit queue then remove m from transmit queue; return; end
  if ((Q_m − NQ_m(u)) < T)
    broadcast(m, Q_m) to all neighbors after k(Q_m − NQ_m(u)) + δ time units;
  end
end
```

Fig. 2. Constrained-flooding meta-strategy

3.3 Meta-strategy 3: adaptive spanning tree

This is a new strategy added to the family, using the same learning core. In cases where destinations (e.g., the base station) are known, it is often more efficient to build an initial spanning tree from the destination. Typical problems with this approach are that the initial tree may be suboptimal due to collisions during tree building, and that an optimal tree may become suboptimal over time due to the dynamic aspects of the network. Rebuilding a complete tree may also result in extra energy consumption and packet loss.

In our framework, an adaptive spanning tree can be built using the same reinforcement learning core as in the previous two meta-strategies. The initialization phase builds an initial spanning tree. The forwarding phase passes the received packet to a node's parent. All the nodes are set to be in promiscuous listening mode. Whenever a node hears a packet, whether it is the designated receiver or not, it updates its corresponding NQ-value and re-estimates its own Q-value, just as in the other two meta-strategies. Similar to real-time search, implicit packet confirmation is used: if the packet is not heard from the forwarded node within a certain time period, the NQ-value is updated to be the largest among the neighbors, and the parent pointer is reset to the neighbor with minimum cost. The pseudo code is illustrated in Figure 3.

Initialization phase:

for all v **do** $NQ_m(v) \leftarrow \text{inf}$ **end**
received $(m,\, \text{Q})$ at w from node u **do**
  $NQ_m(u) \leftarrow Q$; $Q_m \leftarrow (1-\alpha)Q_m + \alpha(o_m + \min_v NQ_m(v))$;
  $p'_m \leftarrow \text{argmin}_n NQ_m(n)$; (random tie break)
  if $p_m \neq p'_m$ **then broadcast**$(m,\, Q_m)$; $p_m \leftarrow p'_m$; **end**
end

Forwarding phase:

received $(m,\, \text{Q})$ at w from node u **do**
  if satisfied$(\mathcal{C}^d_m)$ **then** $Q_m \leftarrow 0$; **broadcast**$(m,\, \mathbf{0})$; **return**; **end**
  $NQ_m(u) \leftarrow Q$; $Q_m \leftarrow (1-\alpha)Q_m + \alpha(o_m + \min_v NQ_m(v))$;
  $p_m \leftarrow \text{argmin}_n NQ_m(n)$; (random tie break)
  if designated(m) **then send**$(m,\, Q_m)$ to p_m; **end**
end

Confirmation phase:

timeout $(m$ to $v)$
  $NQ_m(v) \leftarrow \max_n NQ_m(n) + 1$;
  $p_m \leftarrow \text{argmin}_n NQ_m(n)$; (random tie break)
  if (resend) **then send**$(m,\, Q_m)$ to p_m; **end**
end

Fig. 3. Adaptive spanning-tree meta-strategy

4 Evaluations of Meta-Strategies

We have simulated the three meta-strategies for several real applications using Prowler [10], a probabilistic wireless network simulator. Prowler provides a radio fading model with packet collisions, static and dynamic asymmetric links, and a CSMA MAC layer.

We use a real application to test the performance of all three meta-strategies and also compare them to AODV. The application, Pursuer Evader Game (PEG) [8], is to use the sensor network to detect an evader and to inform the pursuer about its location. The communication problem in this task is to route packets sent out by one of the sensor nodes to the mobile pursuer. The source is changing from node to node, following the movement of the evader, and the destination is mobile. The network is a 7×7 sensor grid with small random offsets. The maximum radio range is about $3d$, where d is the distance between two neighbor nodes in the grid. Let the source be at the middle and the destination be at the upper right corner initially. Assume that the evader and the pursuer move at about the same speed $0.2d/s$, where d is the grid distance, and the source rate be 1 packet per second. The objective of MCBR in this application is simply the minimum number of hops.

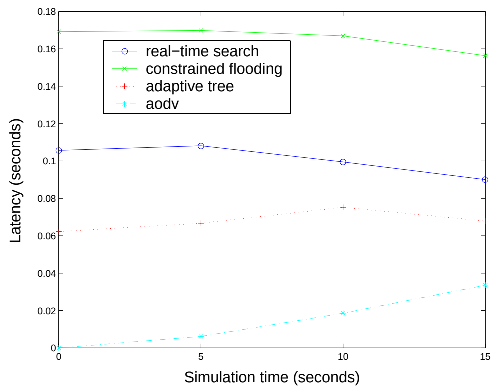
The following performance metrics are used for comparing routing strategies in this paper:

- *latency* – the time delay of a packet from the source to the destination;
- *success rate* – the total number of packets received at the destinations vs. the total number of packets sent from the source;
- *energy consumption* – assuming each transmission consumes an energy unit, the total energy consumption is equivalent to the total number of packets sent in the network;
- *energy efficiency* – the ratio between the number of packets received at the destination vs. the total energy consumption in the network;

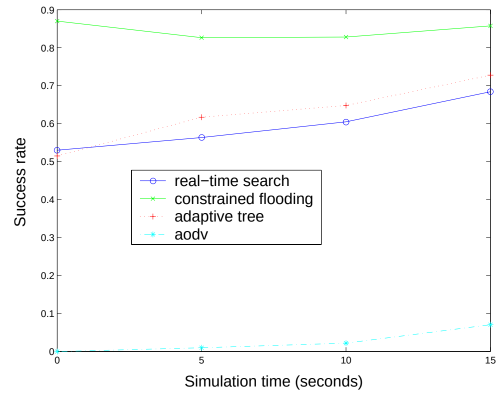
Figure 4 shows performance of four meta-strategies: real-time search, constrained flooding, adaptive tree, and AODV. The results are averaged over 10 random runs. We can see that AODV has the shortest latency, but with low success rate, high energy cost, and low efficiency, while constrained flooding has the highest success rate and efficiency. Both constrained flooding and real-time search have been implemented on the Berkeley mote platform, and tested for the real PEG with 49 motes. In practice, constrained flooding also works better. AODV is the worst algorithm in this application, as the source changes very fast, while learning-based strategies adapt to new situations quickly.

5 Conclusion and Future Work

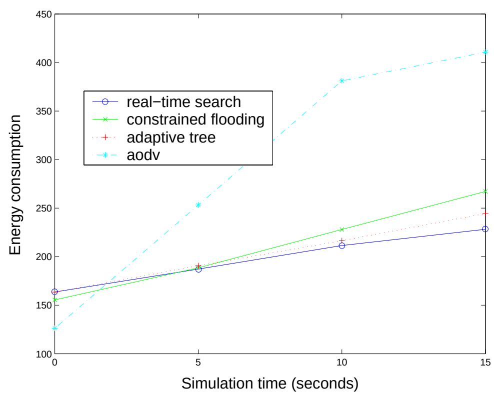
In this paper, we have presented three learning-based meta-strategies for MCBR. MCBR enables X-aware routing strategies, where X can be any attribute of the



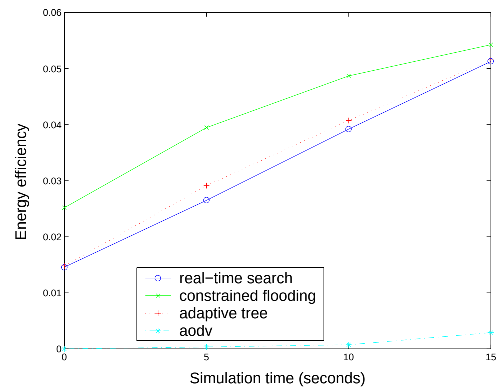
(a)



(b)



(c)



(d)

Fig. 4. Performance Evaluation: (a) Latency, (b) Success rates, (c) Energy consumption, (d) Energy efficiency

system, energy, signal strength, connectivity, even sensor readings, or combinations thereof. The three meta-strategies all use the same reinforcement learning core. Even with its minimal overhead (e.g., there are no extra control packets needed other than the initialization process), this routing scheme is highly adaptive to the dynamic changes of a network. We have also implemented ant-routing for sensor network [16]. A comparison between these two types of learning will be presented in the future. We also plan to experiment with different QoS specifications, such as connectivity-aware or reliability-aware routing.

Acknowledgment

This work is funded in part by Defense Advanced Research Project Agency contract # F33615-01-C-1904.

References

1. K. Chen, S. H. Shah, and K. Nahrstedt. Cross-layer design for data accessibility in mobile ad hoc networks. *Wireless Personal Communication*, (21):49–76, 2002.
2. S. Chen. Distributed quality-of-service routing in ad-hoc networks. *IEEE Journal on Selected Areas in Communications*, 17(8), August 1999.
3. M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *Int. Journal on High Performance Computing Applications*, June 2002.
4. B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. 6th Int'l Conf. on Mobile Computing and Networks (ACM Mobicom)*, Boston, MA, 2000.
5. S. Koenig and R. G Simmons. Complexity analysis of real-time reinforcement learning applied to finding shortest path in deterministic domains. In *National Conference on Artificial Intelligence*, pages 99–105, 1993.
6. C. E. Perkins and E. M Royer. Ad hoc on-demand distance vector routing. In *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.
7. E. Royer and C. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, April 1999.
8. S. Sastry, L. Schenato S. Schaffert, C. Sharp, and B. Sinopoli. Nest challenge problem: Midterm, final demo paln, 2003. DARPA NEST Program Demonstration Plan, http://dtsn.darpa.mil/ixo/nest/day2/UCB1Nest_PI_07_10.ppt.
9. B. Schwartz, A. W. Jackson, W. T. Strayer, W. Zhou, R. D. Rockwell, and C. Partridge. Smart packets for active networks. *ACM Transaction on Computer Systems*, 18(1), Feb. 2000.
10. G. Simon. Probabilistic wireless network simulator. http://www.isis.vanderbilt.edu/projects/nest/prowler/.
11. R. S. Sutton and A. G. Barto, editors. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.
12. F. Ye, G. Zhong, S. Lu, and L. Zhang. A robust data delivery protocol for large scale sensor networks. In *Information Processing in Sensor Networks, Lecture Notes in Computer Science*, number 2634, page 658, 2003.
13. Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks. Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, May 2001.
14. Y. Zhang and M. Fromherz. Message-initiated constraint-based routing for wireless ad-hoc sensor networks. In *Proc. IEEE Consumer Communication and Networking Conference*, 2004.
15. Y. Zhang and M. Fromherz. Search-based adaptive routing strategies for sensor networks. In *AAAI Sensor Networks Workshop*, July 2004.
16. Y. Zhang, L. Kuhn, and M. Fromherz. Improvements on ant-routing for sensor networks. In *Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence*, 2004.