# Dynamic Routing Selection for Wireless Sensor Networks*

W. Ke and T.D.C. Little
Department of Electrical and Computer Engineering
Boston University, Boston, Massachusetts
{*ke, tdcl*}*@bu.edu*

June 2, 2008

MCL Technical Report No. 06-02-2008

**Abstract**–With the decrease in the cost and in the size of computing devices, wireless sensor networks (WSNET) have the potential of being composed by an extremely large number of nodes offering multiple services. Such networks have the capability of executing multiple tasks concurrently by allocating simply a fraction of their resources. Alternatively, many smaller wireless networks may collaborate to execute a larger, unforeseen application. In both cases a routing scheme other than the prevailing one may improve the efficiency of the task or the application being executed, reducing the energy consumption in the network.

We posit that to fully tap into the potential of such networks a new routing infrastructure is needed, one that allows switching between different routing schemes dynamically as required by the applications being deployed, the conditions of the network as a whole and the existing locality information. We show in this paper how dynamic routing scheme selection can be achieved when sensor networks are overlaid with a virtual attribute based cluster hierarchy. We present analytical results for our scheme and show the expected improvement that can be achieved.

*In *Proc. Intl. Symp. on Network Computing and Applications (NCA 2008)*, July 11, 2008, Cambridge, MA. This work is supported by the National Science Foundation under grant No. CNS-0435353. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

# 1 Introduction

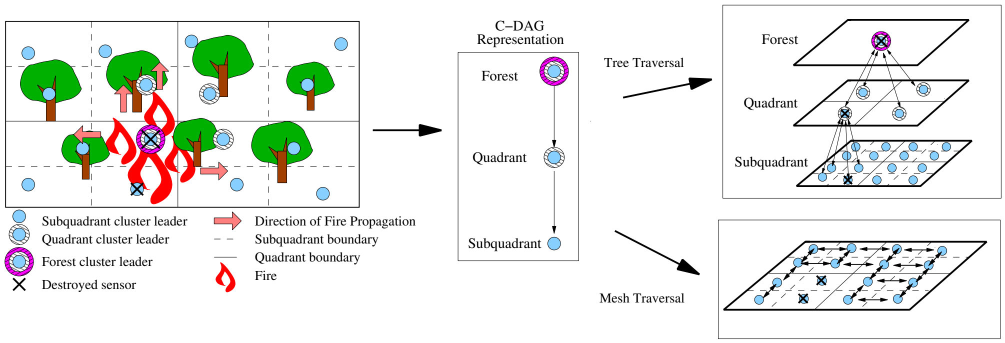


Figure 1: Sensors deployed in a forest under $\{Subquadrant \subset Quadrant \subset Forest\}$ attribute hierarchy and two modes of cluster traversal: *Tree* or *Mesh*.

Current technological advances are enabling the deployment of wireless sensor networks [6] for many different applications. Such applications are also varied in scope and purpose, ranging from object tracking, structural health monitoring, habitat monitoring and monitoring of the environment and its resources. In the future, with such sensors attached to cars, other electronic devices and human bodies, we will live immersed in an all pervading sensorsphere that is composed by the internetworking of various sensor network applications, i.e., a huge networked collection of spatially distributed data collector and actuator points.

To utilize the resources existent in such sensorsphere, an appropriate routing infrastructure must exist. We propose establishing an attribute-based hierarchical clustering structure in the sensor network. The attributes chosen must satisfy locality based relationships (we propose two: containment and adjacency relationships) and may reflect the frequency with which they are present in the inquiries. The clustering mechanism joins attribute equivalent sensors together. Clusterheads in this context act as attribute-based routers, and can support different routing schemes based on the application needs.

Consider the following scenario: multi-modal sensors are deployed over an area for climate monitoring, and are collecting average values of temperature and humidity when suddenly fire is detected. One local application, designed to detect and track how the fire propagates, is awakened and immediately alerts neighbor sensors so that the fire front can be detected. This scenario is depicted in Fig. 1.

The communication needs of the sensor network while in the first stage of monitoring average temperature and humidity can be thought of as hierarchical. Data is slowly aggregated within each cluster by the cluster leader and sent to the base station. Thus sensors communicate using the "Tree traversal" mode found on the upper right side of Fig. 1. However, the communication needs of the fire detection application add a new component: the necessity for clusters to communicate with neighbor clusters, so that the fire propagation can be tracked over time. The way the fire propagates is also recorded and this information is spread to contiguous clusters, as in the event of a fire there is no guarantee that the top hierarchical leader has survived the fire. This situation is also depicted in Fig. 1, in which the sensor which plays the role of Forest leader, as well as Quadrant SouthWest leader has been destroyed by the fire. If the tree traversal hierarchical mode is the only communication mode, then other quadrant leaders would not be able to detect the fire in time. However, by using the "Mesh traversal" mode (lower right side of Fig. 1) at the lowest level of the attribute hierarchy (Subquadrant clusters), sensors are able to spread the alarm and continue detecting the fire front.

The example above illustrates how different applications may require different communication patterns. It is definitely possible, given the sensors are multi-modal [6] that other applications are also present, e.g., wildlife tracking (needs to be able to communicate with neighboring sensors, to alert them of the tracked object, and needs to be able to send logged data back to base station), which would further drive the need for a common, yet flexible routing infrastructure [10].

We show in this paper how routing schemes can be built on top of attribute based hierarchical clustering schemes, and demonstrate, through theoretical analysis, how switching between different traversal modes of the clusters result in higher gains for different metrics. We present in the next section related work in the area. We delineate our design choices and show the basic functionality specification, as well as data structure and algorithms related to the implementation of the routing infrastructure in Sec. 3. We present theoretical performance analysis in Sec. 4 and conclude in Sec. 5.

# 2 Related Work

Past approaches such as diffusion [3, 7] flood inquiries to the network, and build gradients that collect data back. Such approach is limiting, for different applications may have different needs, and if sensor networks are shareable resources, then a single communication paradigm is not sufficient for fully utilizing the resource. Moreover, if the sensor network is shared, requests may arrive for all different forms and types of data, causing frequent floods that may be irrelevant to most of the nodes in the network and wasting energy.

In order to reduce the redundant transmission of packets, location information is explored in order to direct how data can be routed. GPSR (Greedy Perimeter Stateless Routing [8]) and GEAR (Geographical and Energy Aware Routing [16]) are two examples of geographical based routing. Both rely on the presence of location services to operate, and in both the addressing scheme is independent of the applications they support. In other words, a data sink must know *a priori* the region to which send the data request and vice-versa. Data-centric models built on top of such geographic models, such as GHT [14], DIM [11], DIFS [5] and DIMENSIONS [4] do not offer lower level control over communication patterns that different applications may benefit from when transmitting data.

Semantic Routing Trees (SRT) are proposed in [12], in which tree structures are formed in the sensor network based on sensed values and queries are forwarded to children that have values within the range requested. Like SRT but with a more generalized filtering approach, CBCB (Combined Broadcast and Content Based routing [2]) adopts a two layer approach (one broadcast layer and one content-based layer) to place predicates (a set of constraints on the attributes) at the routers. Data that matches a predicate will be forwarded to the appropriate sinks. Our work differs from SRT and CBCB in that we do not attempt filtering at sensor level, but instead form attribute equivalent regions that help route traffic. We also support different communication needs and patterns in such regions.

The advantages of being able to select the routing protocol at run-time have been pointed out by the active network community [13]. Work in [15] proposes encapsulating packets in SAPF

(Simple Active Packet Format) headers, which carry indicators to an active node's FIB (Forwarding Information Base), guiding packet forwarding behavior at run-time. The routing example shown in [15] is tree based. In [1] the authors propose an overlay scheme that allows active nodes to coexist with passive nodes. The active nodes track communication paths to each other reactively. Our work shows how dynamic routing protocol selection can be implemented in attribute clustered WSNETs. We show the routing rules and the performance analysis for both the tree and the mesh traversal modes. Furthermore, we show how the changing density of "active routers" (in our case attribute based routers or cluster leaders) in the network, achieved through changing the number of levels in the attribute hierarchy, affects the expected performance of the two routing schemes. We present in the next section our design choices and some basic functionality specification.

# 3 Routing

In traditional host centric routing schemes identifiers are given to network nodes that are independent of any attributes or data the host may possess. Such approach may be justified when in a network the emphasis is in finding the host, that is, data sets of interest map to a relatively few hosts. However, in the situation in which multiple hosts share common data sets, reaching a specific host is pointless, and the reversal situation should be attempted: to reach the data of interest rather than a specific host.

The challenge in such situations is to propose a scheme that can identify data sets of interest at varying degrees of accuracy. We propose using an attribute hierarchy for this purpose. Attribute hierarchies can be determined by each sensor network locally, have flexible degrees of accuracy (ranging from including all sensors in a large geographic region to being able to pinpoint specific sensors, given enough attributes), can be easily manipulated to provide basic units on top of which routing takes place (e.g., routing between rooms, or floors, or buildings), and can be overlaid (two attribute hierarchies may be used by different applications to target the same set of sensors in different ways, e.g., routing may happen between rooms, floors and buildings, or offices, departments and colleges).

We summarize next some design considerations and characteristics of one mechanism that can provide a logical overlay of attribute based hierarchical clusters on the sensor network. More specific details can be found in [9].

## 3.1 Attribute Based Hierarchical Clustering

The set of attributes used to address sensors must satisfy containment relationships, that is, higher level attributes contain lower level attributes, and have all adjacency relationships defined, that is, which attribute values are spatially contiguous to each other. We represent containment relationships via directed acyclic graphs (C-DAG).

Nodes that have the same attributes are clustered together, and such clustering happens in a hierarchical manner. Each cluster represents an attribute-equivalent region, and by controlling which and how many attributes are part of the hierarchy we can control whether the propagation of inquiries is pure flooding (one level in the hierarchy), host centric (attributes have resolution that can pinpoint individual sensors uniquely), or a hybrid approach, in which we have attribute equivalent regions communicating with each other.

Once attribute equivalent regions have been established, clusterheads can coordinate intra- and inter-cluster data dissemination based on the application requirements. Currently we posit that attributes should be selected based on an *a priori* assumption on the frequency such attributes will be called by users in their inquiries. We support dynamic modifications to the C-DAG after deployment to insert or remove nodes to more efficiently guide data propagation. For more details on the clustering process, see [9].

Table 1: Different Routing Schemes – (a)*Left:* Tree traversal, (b)*Right:* Mesh traversal)

*CDAG* $\leftarrow \{Subquadrant \subset Quadrant \subset Forest\}$;<br>
*RoutingTable* $\leftarrow$ Routing table used by current application;<br>
*SensorAttributes* $\leftarrow$ Attributes current sensor possesses;<br>
*SensorClusters* $\leftarrow$ Set of clusters the current sensor belongs to;<br>
*SensorClusterLeader* $\leftarrow$ Set of clusters the current sensor is leader of;<br>
$\mathcal{N}(X,Y)$ = function that returns the number of consecutively matched attributes between $X$ and $Y$, starting from the first attribute in both $X$ and $Y$;<br>
Received packet `P`;<br>
*DestAttrList* $\leftarrow$ list of attribute name-value pairs of the destination in `P`;<br>
Find $E \in$ *RoutingTable* | ($\mathcal{N}$(*DestAttrList*,$E$) is maximized) ;<br>
**if** ($E$ = *DestAttrList*) **then**<br>
    **if** ($E \in$ *SensorClusters*) **then**<br>
        Flood `P` in $E$; Return;<br>
    **else if** (`P.PrevHop` $\notin$ {path between current sensor $\wedge$ $E$}) **then**<br>
        Send `P` to $E$; Return;<br>
<br>
**if** ($\exists$ $L \in$ *SensorClusterLeader* | ($L$ = `P.NextHop`)) **then**<br>
    **if** (`P.PrevHop` is parent node in *CDAG*) $\vee$ (sensor is root leader) **then**<br>
        **if** ($\exists$ children node | known attributes of children node match *DestAttrList*) **then**<br>
            Send `P` to children node in *CDAG*;<br>
        **else**<br>
            Drop packet `P`;<br>
    **else**<br>
        **if** ($\exists$ unmatched attribute at level $L$ or higher between the sensor and *DestAttrList*) **then**<br>
            Send `P` to parent of $L$;<br>
        **else if** (all attributes from root to level $L$ match between the sensor and *DestAttrList* $\wedge$ $\exists$ child cluster with increased attribute match) **then**<br>
            Send `P` to sibling clusters;<br>
            Send `P` to child cluster;<br>
        **else**<br>
            Drop `P`;<br>
**else**<br>
    Send `P` to leader of `P.NextHop`;

*CDAG* $\leftarrow \{Subquadrant \subset Quadrant \subset Forest\}$;<br>
*RoutingTable* $\leftarrow$ Routing table used by current application;<br>
*SensorClusters* $\leftarrow$ Set of clusters the current sensor belongs to;<br>
*SensorClusterLeader* $\leftarrow$ Set of clusters the current sensor is leader of;<br>
$\mathcal{N}(X,Y)$ = function that returns the number of consecutively matched attributes between $X$ and $Y$, starting from the first attribute in both $X$ and $Y$;<br>
Received packet `P`;<br>
**if** (`P` was received before) **then**<br>
    Return;<br>
*DestAttrList* $\leftarrow$ list of attribute name-value pairs of the destination in `P`;<br>
Find $E \in$ *RoutingTable* | ($\mathcal{N}$(*DestAttrList*,$E$) is maximized) ;<br>
**if** ($E$ = *DestAttrList*) **then**<br>
    **if** ($E \in$ *SensorClusters*) **then**<br>
        Flood `P` in $E$; Return;<br>
    **else if** (`P.PrevHop` $\notin$ {path between current sensor $\wedge$ $E$}) **then**<br>
        Send `P` to $E$; Return;<br>
<br>
**if** ($\exists$ $L \in$ *SensorClusterLeader* | ($L$ = `P.NextHop`)) **then**<br>
    **if** ($\exists$ children node | known attributes of children node match *DestAttrList*) **then**<br>
        Send `P` to children node in *CDAG*;<br>
    **else if** ($\exists$ adjacent cluster $C$ at same level of $L$ with matching attribute $\wedge$ no copy of `P` came from $C$) **then**<br>
        Forward `P` to all such $C$;<br>
    **else**<br>
        Drop `P`;<br>
**else**<br>
    Send `P` to leader of `P.NextHop`;

## 3.2 Rules Based Routing

In our proposed framework the routing process is an interpreted one. Nodes in a WSNET process incoming packets based on routing rules that are grouped by sets. Different sets of routing rules define different communication patterns (e.g., “tree” traversal, or “mesh” traversal). By setting routing as an interpreted process, we allow dynamic configuration of nodes to support different communication patterns and thus meet different communication needs from the various applications that share the network.

Each “rule” in our rules based routing is composed of two parts: (1) a conditional statement and (2) an action statement. If the conditions specified are true, then the action is carried out. Otherwise, the following rule in the rule set is checked. If no conditional statement turns out true after going through all the rules, the packet is simply dropped. Our rules based approach essentially imposes a priority scheme over possible next-hop destinations.

In the right side of Fig. 1 two communication patterns can be established, the “Hierarchical Tree Traversal” mode, in which lower level cluster leaders communicate with higher level cluster leaders, routing in a hierarchical virtual tree, or the “Mesh Traversal” mode, in which cluster leaders at the lowest level in the C-DAG communicate with adjacent cluster leaders, routing on a logical mesh. Routing rules for both traversal modes are shown in Table 1. In the tree traversal, unknown destination packets may be sent to higher level cluster leaders (Line 24 of Table 1a), and these may eventually forward the packets back (Line 27 of Table 1a). The Mesh traversal algorithm forwards packets of unresolved attributes to neighbor clusters (Line 21 in Table 1b). Notice the different approach each routing rule establishes on resolving unknown addresses: while in the tree case the packets are forwarded up the hierarchy level, in the mesh the packets are simply spread towards other adjacent clusters. These two resolution modes also characterize the intrinsic communication pattern each rules set supports. Sensor networks that are deployed for different applications will benefit from being able to support switching between the two modes, as we will show in the next section.

# 4 Performance Analysis

In this section we study the performance of the Mesh routing scheme represented by Table 1 as applied to a “line” C-DAG (center of Fig. 1), and of schemes that rely on “flooding” for data propagation, as well as schemes that have full knowledge of all sensors in the network. Due to space limitations we will offer only analysis on the Mesh traversal scheme. Readers are referred to [10] for analysis on the Tree traversal scheme details.

The network is consisted of $N$ sensors spread uniformly over a square region of area $L^2$ and there are $l_h$ levels in the line attribute hierarchy. Since the C-DAG representation of the attribute hierarchy is a line, there are $l_h$ nodes in the C-DAG. The root node (at level $1$) in the C-DAG covers the whole region, while subsequent nodes (at levels $l_i$, $i \in \{2, \ldots, l_h\}$) have four possible values each (a quadtree format), with each value covering a square region of side $L/2^{(i-1)}$. In the right of Fig. 1 a three level C-DAG is shown.

The metrics we will be studying for each scheme include: (1) total memory requirement from all nodes for implementation; (2) the estimated number of transmissions taken when routing one packet from a source to an unknown destination in the worst case (considering that the sensors are deployed over a square region, the worst case is when source and destination lie at opposite corners across a diagonal) and (3) the estimated number of hops that separate source from destination after the destination’s address has been “resolved” in (2). Essentially (1) allows us to gauge how scalable each scheme is in terms of the amount of memory needed. Metric (2) allows us to compare the cost of resolving an unknown destination address, while (3) is an estimate of how quickly the destination address can be found or how quickly data can be transmitted to the destination, assuming both being directly proportional to the hop distance that separates source from destination.

When estimating item (2) and (3) above, for non-flooding type of schemes, we consider that the path the packet takes is composed of consecutive straight line segments. One estimate of the number of transmissions (or the number of hops) is the product of the length of the segment by the linear node density. The node density is given by $\rho = N/L^2$, thus one estimate of the number of neighbors that lie on a line segment within transmission radius $R$ is $R\sqrt{\rho}$. On the average, assuming the sensors are uniformly distributed and the whole network connected, the number of transmissions should not be greater than this value, for this value reflects the number nodes that lie in the segment. If this value is $\gg 1$, then we are overestimating the number of transmissions needed. Estimates made in this way can still be used for comparison between different routing schemes, though, since the overestimation comes from the high node density value and will be reflected by all routing schemes.

An estimate that is closer to the minimum number of transmissions (or number of hops) needed to cover the path between source and destination is obtained by dividing the path length by the transmission range $R$. However, when the "line" C-DAG has a very high number of nodes (i.e., high $l_h$), the leaf node's covered region may be smaller than the transmission range ($L/2^{(i-1)} \ll R$, when $i \gg 1$). Because our hierarchical routing scheme stores routing information based on attribute regions, and routes according to containment and adjacency relationships, the lower bound in the number of transmissions is the number of attribute regions traversed. The results of our performance comparison are summarized in Table 2.

Table 2: Performance Metrics for different Routing Schemes

| | | Flooding | Full | Tree (One level information) | Tree (Full cluster information) | Mesh |
|---|---|---|---|---|---|---|
| Memory | | 0 | $E\,N(N-1)$ | $E\,\frac{4}{3}\,(4^{(l_h-1)}-1)+E\,N\,l_h$ | $E\,2\,N\,l_h$ | $E\,(4^{l_h}+N+2(2^{(l_h-1)}-1)\sqrt{N})$ |
| Num Tx | Max | $N$ | $\sqrt{2N}$ | $\sqrt{N}\,(2^{(l_h-1)}-1)(\frac{2\sqrt{2}}{2^{(l_h-1)}}+\frac{3\sqrt{2}}{2}+\sqrt{5})+\frac{N}{2^{(2l_h-2)}}$ | $4\sqrt{2N}(1-\frac{1}{2^{(l_h-1)}})+\frac{N}{2^{(2l_h-2)}}$ | $2\sqrt{2N}(2^{l_h}-\frac{2}{2^{(l_h-1)}})+\sqrt{N}(\frac{8-4\sqrt{2}}{2^{(l_h-1)}})+\frac{N}{2^{(2l_h-2)}}$ |
| | Min | $N$ | $L\,\sqrt{2}/R$ | $\max(\frac{L}{R}\,(2^{(l_h-1)}-1)(\frac{2\sqrt{2}}{2^{(l_h-1)}}+\frac{3\sqrt{2}}{2}+\sqrt{5}),\sum_{i=2}^{l_h}(4^{(i-2)}\lceil\frac{L\sqrt{2}}{R2^{(i-1)}}\rceil+4^{(i-2)}\,2\lceil\frac{L\sqrt{5}}{R2^{(i-1)}}\rceil+(4^{(i-2)}+1)\lceil\frac{L\sqrt{2}}{R2^{(i-2)}}\rceil))+\frac{N}{2^{(2l_h-2)}}$ | $\max(\frac{4L\sqrt{2}}{R}(1-\frac{1}{2^{l_h-1}}),\sum_{i=2}^{l_h}2\lceil\frac{L\sqrt{2}}{R2^{(i-2)}}\rceil)+\frac{N}{2^{(2l_h-2)}}$ | $\max(\frac{L2\sqrt{2}}{R}(2^{l_h}-\frac{2}{2^{(l_h-1)}})+\frac{L(8-4\sqrt{2})}{R\,2^{(l_h-1)}},2^{l_h}\,(2^{(l_h-1)}-1))+\frac{N}{2^{(2l_h-2)}}$ |
| Num Hops | Max | $\sqrt{2N}$ | $\sqrt{2N}$ | $4\sqrt{2N}(1-\frac{1}{2^{(l_h-1)}})$ | | $2\sqrt{2N}(1-\frac{1}{2^{(l_h-1)}})+\frac{\sqrt{N}(4-2\sqrt{2})}{2^{(l_h-1)}}$ |
| | Min | $L\,\sqrt{2}/R$ | $L\,\sqrt{2}/R$ | $\max(\frac{4L\sqrt{2}}{R}(1-\frac{1}{2^{l_h-1}}),\sum_{i=2}^{l_h}2\lceil\frac{L\sqrt{2}}{R2^{(i-2)}}\rceil)$ | | $\max(\frac{L}{R}(2\sqrt{2}(1-\frac{1}{2^{(l_h-1)}})+\frac{4-2\sqrt{2}}{2^{(l_h-1)}}),2\,(2^{(l_h-1)}-1))$ |

**Flooding** A flooding based routing scheme does not need to store any routing information about the network. Every packet is flooded to the whole network. Consequently, the memory requirement is zero[1] and it takes $N$ transmissions to deliver the packet. The farthest any two sensors may be from each other is if they lie at opposite corners across a diagonal. Thus, transmission across the diagonal must cross a minimum of $L\sqrt{2}/R$ hops. Given the node density of $\rho = N/L^2$, then an estimate of the number of nodes lying in the diagonal is $L\sqrt{2\rho} = \sqrt{2N}$, and this is, on the average, the maximum number of hops that separates source from destination.

[1] Diffusion schemes are not purely flooding schemes, since Diffusion remembers paths to published source/sink.

**Full Knowledge** A routing scheme that stores next hop routing information for all nodes in the network has a huge memory requirement. In fact, each node needs to store information about $N-1$ other nodes in the network. Considering that each routing entry requires $E$ bytes, the total memory requirement in the network is $E\,N(N-1)$. However, because of the complete knowledge, the number of transmissions triggered and the number of transmissions needed to send the packet are equal. These are equal to the estimated maximum and minimum number of hops in the flooding case.

**Cluster Flooding** In both *Flooding* and *Full Knowledge* schemes destination sensors are sure to be reached. In "Tree" or "Mesh" schemes below, however, packets reaching the intended leaf cluster(s) still need to reach the sensors. Assuming the intended destination address "resolves" into one leaf cluster, to flood that cluster the number of additional transmissions is equal to $\rho\,(L/2^{(l_h-1)})^2 = N/2^{(2l_h-2)}$ is needed. This term appears in all "NumTx" entries in Table 2.

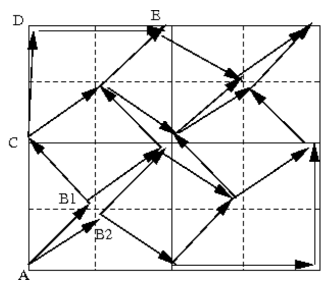


Figure 2: Propagation path for *Mesh* mode when resolving unknown destination address

**Mesh** We study the performance of a routing scheme in a mesh like topology at only one attribute hierarchy level (say $l_h$). In a mesh like routing scheme, we assume each cluster leader tracks only its (at most) four neighbor clusters, resulting in a memory requirement of $E\,4\,4^{(l_h-1)}$. Also all sensors track their cluster leader ($E\,N$ of memory), and sensors that lie at the attribute border will track the two clusters for which it is the border. At level $l_h$, the total length of the border is $2(2^{(l_h-1)}-1)L$, which, when multiplied by $\sqrt{N}/L$ and summed with the other terms, results in the memory requirement equation seen in Table 2.

We assume that when a packet with an unknown destination is received it will be transmitted to the neighbor clusters other than the ones from which the packet arrived. Thus if a packet is sent from the lower left cluster leader, with a destination that is unknown to the cluster leader, but whose final sink is in the top right cluster, then the packet will be propagated across all attribute regions. The total length traversed as the packet is distributed in the network is longer if the cluster leaders are located close to opposite corners across the diagonal, in the zigzag pattern shown in Fig. 2(b). In this figure we show the traversal taken when there are three nodes in the line C-DAG. The cluster leader of the lower left attribute region ($A$) sends the packet to its immediate neighbor cluster leaders ($B1$ and $B2$). As these are located close to the corner across the diagonal the length traversed is $2L\sqrt{2}/2^{(l_h-1)}$. To increase the length traversed, as the packet gets closer to the top left and bottom right corners, we assume the cluster leaders are located at the corners of their respective attribute regions. In this way we force comparison of the worst case in a mesh approach with the worst case of the tree based scheme analyzed previously. Notice that essentially the packet traverse the diagonals of squares with side length $2jL/2^{(i-1)}, j \in \{1,2,3,...,2^{(i-1)}\}$ in a regular fashion, discounting the borders and the top left and bottom right corners. The total length traversed, and the corresponding expected number of transmissions (both maximum and minimum) are given by the corresponding expressions in Table 3.

When the transmission radius $R \gg L/2^{(i-1)}$, then it takes at least one transmission to cross one attribute region, and assuming each attribute region will transmit to two of its immediate neighbors (with top and right border attribute regions transmitting only once), the total number of transmissions will be $2(2^{(l_h-1)}-1)+2(2^{(l_h-1)}-1)^2 = 2^{l_h}(2^{(l_h-1)}-1)$, as seen in the table.

The shortest path that separates the source from the destination must traverse $2(2^{(l_h-1)}-1)+1$ attribute regions (the $+1$ is because the source attribute region also must be traversed). However, if the packet goes through only the diagonals, only $2(2^{(l_h-1)}-1)$ diagonals need be crossed. One of the attribute region leaders will receive the packet from the left and can immediately forward to the upper region, without needing to traverse itself. Thus the worst case scenario is actually when the source is at the top left corner while the destination is at the bottom right corner (or vice-versa). In this case there are additional four traversals across the border of the attribute region $(4(L/2^{(l_h-1)}))$ and two less diagonal traversals. This explains the second term in the "NumHopMax" and the second term in the first argument to the $\max$ function in "NumHopMin." When we are considering the minimum number of hops, this must be lower bounded by the number of attribute regions that need be crossed ($2(2^{(l_h-1)}-1)$), since in principle the cluster leader only tracks the four adjacent clusters. We show some plots of the equations of Table 2 in Fig. 3.

We can see from Fig. 3(a) and Fig. 3(b) that the expected number of transmissions to resolve an unknown address in the worst case is higher for the Mesh traversal mode than for the Tree cases. In fact, when cluster leaders track full cluster information, the performance dramatically improves. This is because the root node need not propagate the packet with unknown address down to all of its children clusters. We can see that the high number of levels in the attribute hierarchy contributes to the inefficiency of the process (Fig. 3(b) and 3(e)). With the increase in the number of hierarchies, the packet with unknown destination address need essentially be distributed to the whole network in the Mesh and Tree (with one level information) schemes at increasing levels of granularity (i.e., covering more of the network), contributing to their performance degradation.

A high number of levels will involve transmission costs to cross adjacent clusters in the Mesh case and costs to resolve all the way to the leaf cluster in the Tree (one level info) case. These costs surpass those of the mere flooding schemes and should be avoided. The cost for resolving an unknown address in the Tree (full cluster info) case remains constant. However, the memory requirements are high (Figs. 3(c) and 3(f)).

When we consider the number of hops metric, we find that Mesh schemes are able to find shorter paths between source and destination. The only drawback is that Mesh schemes currently only cross spatially adjacent attribute regions. Thus when the number of levels in the hierarchy increases, there is a corresponding increase in the hop distance (Figs. 3(h) and 3(i)).

From the graphs in Fig. 3 we can see that if the network is composed of heterogeneous nodes, in which some nodes have higher capacity, then a Tree (full cluster info) scheme will be the most economical in transmission costs related to address resolution issues. Sensor networks that have a high inquiry arrival, especially from a large user base, will benefit from the increased savings in Tree based address resolution schemes, while applications that require fast response can invoke Mesh traversal mode for their data packets.

# 5 Conclusion

In this paper we presented examples of different applications being tasked to the same sensor network simultaneously. Due to the different objectives of the applications, their underlying data communication and dissemination patterns favor different routing schemes. We envision that sensor networks will be widespread in the future. and to tap into the full potential of such sensorsphere, the underlying routing infrastructure must support dynamic routing scheme selection. With this feature, tasked applications can request routing support from a scheme whose packet forwarding rules match their data communication requirements, thus maximizing their performance.

In order to enable dynamic routing scheme selection, we propose using sets of routing rules that forward data in pre-defined ways as the elements to be selected at runtime. We assume that the underlying sensor network has been clustered according to a hierarchy of attributes, and that containment and adjacency relationships between the clusters (or the attributes) are clearly defined. We present in this paper two routing rules set for applications deployed in a sensor network with the above mentioned logical structure. One rules set implements Tree traversal mode while the other Mesh traversal mode. We show analytical performance results of the two traversal modes and show that Mesh traversal mode favors applications that need fast response, while Tree traversal mode has less transmission cost when resolving a previously unknown destination address.

# References


[1] S. Calomme and G. Leduc. Performance Study of an Overlay Approach to Active Routing in Ad Hoc Networks. In *Proc. 3rd Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2004)*, Bodrum, Turkey, Jun 2004.

[2] A. Carzaniga, M. J. Rutherford, and A. L. Wolf. A Routing Scheme for Content-Based Networking. In *Proc. IEEE INFOCOM'04*, Hong Kong, China, 2004.

[3] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proc. 5th ACM MobiCom Conference*, Seattle, WA, August 1999.

[4] D. Ganesan, D. Estrin, and J. Heidemann. DIMENSIONS: Why do we need a new Data Handling architecture for Sensor Networks? In *Proc. 1st Workshop on Hot Topics In Networks (HotNets-I)*, Princeton, NJ, October 2002.

[5] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker. DIFS: A Distributed Index for Features in Sensor Networks. In *Proc. 1st IEEE Intl. Workshop on Sensor Network Protocols and Applications (SNPA'03)*, 2003.

[6] J. Hill and D. Culler. MICA: A Wireless Platform For Deeply Embedded Networks. *IEEE Micro*, 22(6):12–24, Nov/Dec 2002.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proc. International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, August 2000.

[8] B. Karp and H. T. Kung. Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of the 6th ACM MobiCom Conference*, pages 243–254, Boston, MA, August 2000.

[9] W. Ke, S. A. Ayyash, P. Basu, and T. D. C. Little. Attribute-Based Clustering for Information Dissemination in Wireless Sensor Networks. In *Proc. 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON'05)*, Sant Clara, CA, USA, September 2005.

[10] W. Ke and T. D. C. Little. Dynamic Route Selection for Wireless Sensor Networks. Technical report, Boston University, 2006. MCL Technical Report 05-02-2006 `http://hulk.bu.edu/pubs/papers/2006/TR-05-02-2006.pdf`.

[11] X. Li, Y. J. Kim, R. Govindan, and W. Hong. Multi-dimensional Range Queries in Sensor Networks. In *Proc. 1st ACM Intl. Conference on Embedded Networked Sensor Systems (Sensys'03)*, Los Angeles, CA, USA, November 2003.

[12] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. The Design of an Acquisitional Query Processor for Sensor Networks. In *Proc. ACM SIGMOD*, San Diego, CA, June. 2003.

[13] C. Prehofer and Q. Wei. Active Networks for 4G Mobile Communication: Motivation, Architecture, and Application Scenarios. In *Proc. IFIP-TC6 International Working Conference (IWAN'02)*, London, UK, 2002.

[14] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A Geographic Hash Table for Data-Centric Storage. In *Proc. 1st ACM Intl. Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, September 2002.

[15] C. Tschudin, H. Gulbrandsen, and H. Lundgren. Active routing for ad-hoc networks. *IEEE Communications Magazine, Special issue on Active and Programmable Networks*, April 2000.

[16] Y. Yu, R. Govindan, and D. Estrin. Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks. Technical report, UCLA Computer Science Department Technical Report UCLA/CSD-TR-01-0023, May 2001.

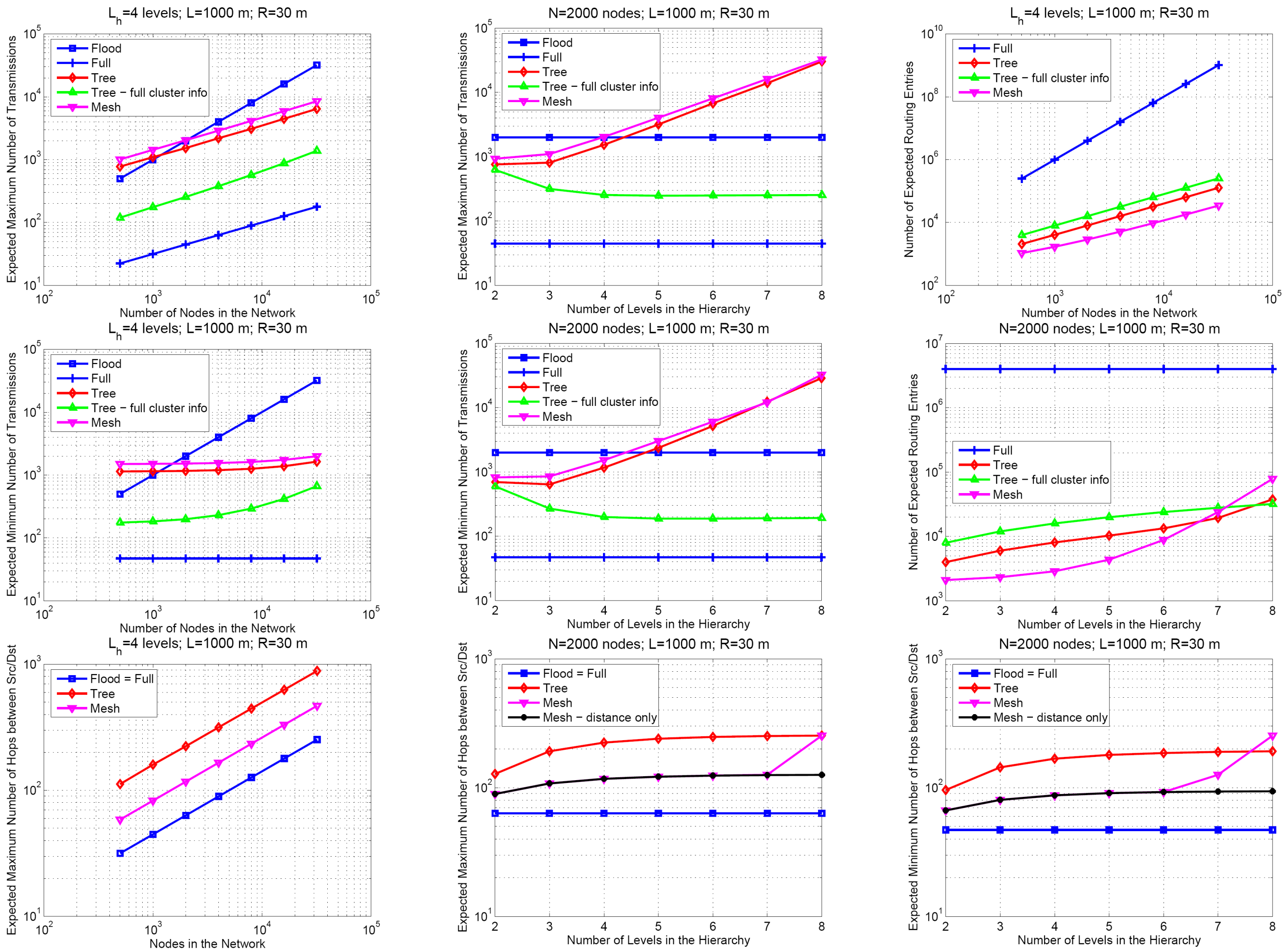


Figure 3: (a) *Top,Left:* Graph of expected maximum number of transmissions ($NumTxMax$) triggered by a packet with unknown destination (b) *Top,Center:* Graph of how $NumTxMax$ varies according to the number of levels in the hierarchy (c) *Top,Right:* Graph of Memory requirements with increasing number of nodes in the network (d) *Center,Left:* Graph of expected minimum number of transmissions ($NumTxMin$) triggered by a packet with unknown destination (e) *Center,Center:* Graph of how $NumTxMin$ varies according to the number of levels in the hierarchy (f) *Center,Right:* Graph of how memory requirements varies according to the number of levels in the hierarchy (g) *Bottom,Left:* Graph of expected maximum number of hops ($NumHopMax$) that separates source from destination as the number of nodes in the network increases (h) *Bottom,Center:* Graph of how $NumHopMax$ varies according to the number levels in the hierarchy (i) *Bottom,Right:* Graph of expected minimum number of hops ($NumHopMin$) that separates source from destination as the number of levels in the hierarchy changes