# CRES: A Content-based Routing Substrate for Large-Scale Data-Centric Sensor Networks

Cuong Pham Duc A. Tran
Department of Computer Science
University of Massachusetts, Boston
{cpham, duc}@cs.umb.edu

***Abstract*—We propose CRES as a substrate structure to enable content-based routing in large-scale sensor networks without location information. With CRES, traffic is routed based on its content rather than the destination's physical address. CRES is therefore applicable to data-centric sensor networks where sensor data and user queries need to be mapped to their corresponding nodes in a distributed manner such that a query can find its matching data efficiently.**

## I. Introduction

Allowing traffic to be routed based on its content rather than destination address, the content-based networking paradigm is highly suitable for data-centric sensor networks. In such a network, the sensor data is not always forwarded to a central sink for storage; instead, the network nodes serve as a distributed storage, collectively storing the sensor data and waiting to answer user queries. A key question in designing such a sensor network is how to map a data object to the node(s) where it should be stored and also to map a query to the node(s) where it will find the matching data.

Several DHT techniques such as [1], [2] have been proposed to address the above question for Internet-based networks. For sensor networks with known geographic information, one can use the method of GHT [3] – to hash values, $k$, in the data space to geographic locations, $h(k)$. An extension of GHT in combination with landmark-based routing to improve routing efficiency is proposed in [4]. Another technique based on *double ruling* is proposed in [5], whose nice property is the guarantee that the length of the path from the data source node to the query node is within a small factor of their Euclidean geographic distance.

The geographic information, however, is not always available for a sensor network, for which case we propose in this paper a novel DHT technique, called CRES [1]. Compared to existing DHT techniques, e.g. [6], [7] which are designed also for sensor networks without location information, our technique is aimed for large-scale networks. The former have a high cost to maintain the DHT structure under network dynamics. CRES alleviates that problem by building its DHT structure based on a partition of the network into clusters where a failure would have only local impact. Our simulation substantiates several desirable properties of CRES in terms of routing distance, and capability to cope with network failures.

[1]CRES = **C**ontent-based **R**outing Substrat**E** for **S**ensor Networks

## II. CRES: The Proposed Solution

We assume a sensor network with the existence of $m$ "reference" nodes, $\{\Lambda_1, \Lambda_2, ..., \Lambda_m\}$, that are supposed to be stable. Geographic location information is not required for sensor nodes or reference nodes. The purpose of CRES is to (1) organize the network as a partition into $m$ clusters, $\{CL(\Lambda_1), CL(\Lambda_2), ..., CL(\Lambda_m)\}$, each indexed by a reference node; and (2) assign to each node a virtual address (VA), which is a binary identifier determined based on the "position" of the node in its corresponding cluster. A node is uniquely identified by its VA and its cluster membership.

### A. Construction

In CRES, each node $X$ has a *state*, represented by $info(X) \equiv < va(X), cl(X), dist(X) >$, where $va(X)$ is the VA of node $X$, $cl(X)$ the cluster where $X$ resides, and $dist(X) \equiv < dist(X, \Lambda_1), dist(X, \Lambda_2), ..., dist(X, \Lambda_m) >$ the vector of shortest distances from the reference nodes to $X$. If $X$ is a reference node, we always have $va(X) = \varnothing$ and $cl(X) = CL(X)$. The VA information is used to route messages inside a cluster, whereas the distance information is used to route messages from a cluster to another. Except for the reference nodes, all the above information is initially unknown for every node but will eventually be filled as nodes periodically exchange information with their neighbors in a common heartbeat mechanism. Specifically, once a node $X$ has obtained its VA, it periodically sends a heartbeat message advertising $info(X)$ to all its neighbors. Upon receipt of such a message, each neighbor $Y$ updates its state as follows:

1) Update distance information: set $dist(Y, \Lambda_i) \leftarrow \min(dist(Y, \Lambda_i), dist(X, \Lambda_i) + 1)$, for each reference node $\Lambda_i$.
2) If $va(Y)$ has not been set, update VA and cluster information,
   a) Set $cl(Y) \leftarrow cl(X)$ (i.e., $Y$ will be in the same cluster as $X$).
   b) Set $va(Y)$ to a binary identifier according to the *VA prefix rule*: $va(Y)$ is a string (shortest length prefered) of the form $va(X)$ + '$\underbrace{0...0}_{i \geq 0}1$' unused[2]

[2]To avoid VA collisions, $X$ can pre-compute such available strings and include in the advertisement message a string for each of its neighbors.

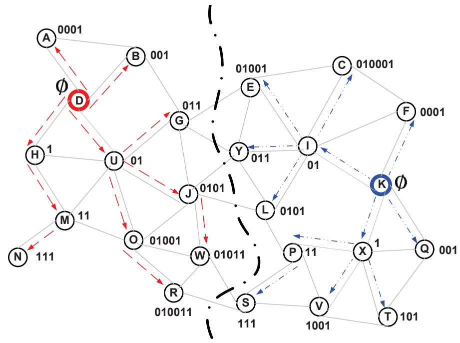



Fig. 1. The CRES substrate with two reference nodes ($D$ and $K$), their corresponding clusters ($CL(D)$ on the left side, $CL(K)$ on the right side, separated by the dotted boundary line). The order of filling the VA information resembles a breadth-first search process, represented by the arrow links, e.g., $J \rightarrow W$ means that $va(W)$ is determined based on $va(J)$.

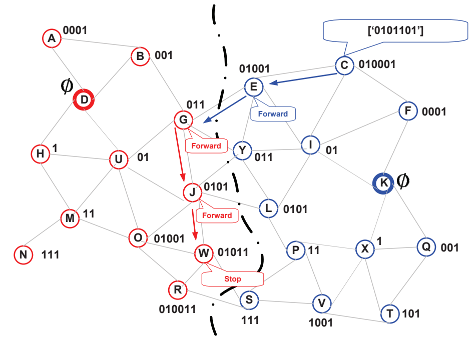



Fig. 2. Routing of key '0101101' from source node $C$: supposing that $h(\text{'0101101'}) = D$, the routing path consists of the shortest-path segment, $C \rightarrow E \rightarrow G$ towards the reference $D$ trying to get to the first node residing in $CL(D)$, and the prefix-routing segment, $G \rightarrow J \rightarrow W$ from $G$ towards $W$ – the designated node of the key '0101101' in $CL(D)$.

by any other neighbor node of $X$. $X$ is called the "predecessor node" of $Y$.

CRES is therefore constructed in a decentralized manner. Each node only needs to communicate with its neighbors, asynchronously and independently. Once a node $Y$ hears from the first neighbor, $X$, with an established VA, $Y$ obtains a VA based on the VA prefix rule and becomes a member of the same cluster that $X$ node belongs. The order of assigning a VA to a node is similar to the order of a breadth-first search process; nodes near the reference nodes will have their VA assigned first before those that are far away. The partition into clusters resembles a Voronoi diagram; nodes are assigned to a cluster represented by a reference node if they are closer to this reference node than they are to any other reference node. An example is illustrated in Figure 1, where the network comprises of 25 nodes $\{A, B, ..., Y\}$, two of which (node $D$ and node $K$) serve as the reference nodes.

The following definition and property of CRES are useful for the next section. Given a node $X$, we define the "key zone" of $X$, denoted by $kzone(X)$, as the set of all the binary strings, of which $va(X)$ is *the longest* prefix. For example, consider Figure 1. The key zone of reference $D$ is $kzone(D) = \{$'0', '00', '000', '0000*'$\}$, and for a normal node $I$, $kzone(I) = \{$'01', '010', '0100', '01000', '010000*'$\}$.

**Property II.1.** *The key zones of all the nodes in any given cluster form a partition of the universe of binary strings; each node's key zone is a "part" of this partition. In other words, for any cluster $\mathbb{C}$, consisting of the nodes $\{X_1, X_2, ..., X_{|\mathbb{C}|}\}$, we have: (1) $kzone(X_p) \neq \varnothing \; \forall p$; (2) $kzone(X_p) \cap kzone(X_q) = \varnothing \; \forall \; p \neq q$; (3) $\sum_j kzone(X_j) = \{0, 1\}^*$.*

### *B. DHT*

We now discuss the mapping between a key value and the node where it will be stored (in the case of a data object) or retrieved (in the case of a user query). We assume a $k$-bit key space. The parameter $k$ should be chosen to be larger than the longest VA length in the network. Property II.1 implies that given any key $\rho$ it must belong to the key zone of one and only node in each cluster; this node is called the "designated node", denoted by $node(\rho : \Lambda_i)$, corresponding to cluster $CL(\Lambda_i)$. For example, consider Figure 1; there are two designated nodes for key '0100010': $node(\text{'0100010'}: D) = U$ in cluster $CL(D)$ and $node(\text{'0100010'}: K) = C$ in cluster $CL(K)$.

Our DHT is simple: *the hash node for key $\rho$ is $node(\rho : h(\rho))$ – the designated node of $\rho$ residing in cluster $CL(h(\rho))$*. Here, $h(.)$ is a pre-determined hash function, which can be any hash function mapping a $k$-bit key to a random reference node in $\{\Lambda_1, \Lambda_2, ..., \Lambda_m\}$. CRES provides a convenient structure for routing to the hash node of a given key $\rho$. If the hash node is in the same cluster with the source node, we can apply *prefix routing*. A node that receives the message always forwards it to a neighbor whose VA shares a longer prefix with $\rho$; if no such neighbor exists and the current node is not a prefix of $\rho$, the message is forwarded to the predecessor node. It is guaranteed that the message will eventually reach the hash node. In the case that the hash node is in a cluster difference from the source node's, a two-phase routing protocol is applied: (1) shortest-path phase: each intermedia node, initially the source node, forwards the message to the neighbor $Y$ with minimum $dist(Y, h(\rho))$ (this information is known from the neighbors' state information). Node $Y$ and the subsequent intermediate nodes that receive the message apply the same greedy strategy. The routing switches to the prefix-routing phase when the message visits a node $Z \in CL(h(\rho))$; (2) prefix-routing phase: starting from node $Z$, the message is routed towards the hash node based on prefix routing as aforementioned. An example is demonstrated in Figure 2.

### *C. Maintenance Mechanisms*

Changes in the network, occurring to a sensor node or a reference node need to be addressed for CRES to maintain its structure. Because CRES is based on a partition of the network into clusters, the failure of a node affects only a subset of nodes in the local cluster. Other DHT-like techniques [6], [7] that have been proposed for sensor networks without

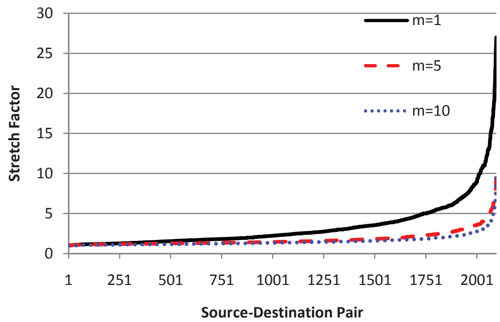



Fig. 3. Stretch factor

location information usually require a high cost to maintain their structure. The maintenance mechanisms are omitted from this paper due to its space limit, but their details can be found in our extended paper [8].

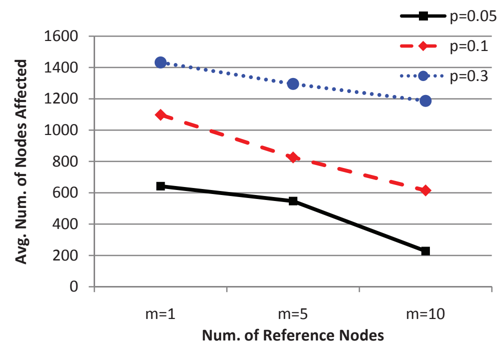
## III. Evaluation Study

We evaluated the proposed technique based on a simulation study. The simulated network consists of 1479 sensor nodes uniformly placed in a 500m $\times$ 200m field, each node having a communication radius of 20m. We assumed a key space of 64 bits. For each key value $\rho$, we define the source node as the node that initiates a data object with key $\rho$ and the destination node as the node that sends a query to get the data object with key $\rho$. To evaluate the efficiency of CRES, we compute the stretch factor, $d_{cres}/d_{optimal}$, where $d_{cres}$ is the total hopcount distance from the source node to the destination node via the hash node, and $d_{optimal}$ is the shortest hopcount distance from the source node to the destination node directly. Figure 3 shows that by using more reference nodes the stretch factor of CRES can be improved; the largest improvement is when $m$ is increased from 1 to 5, while marginal when $m$ is increased from 5 to 10. When $m$ is 5 or 10, the stretch factor for most source-destination paths is about 2 or less.

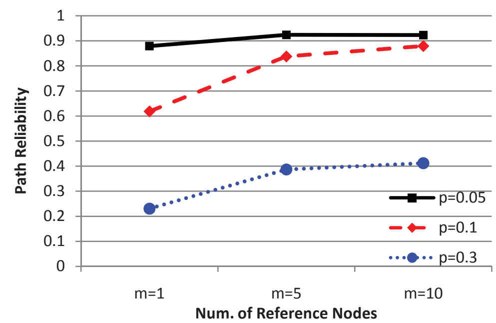
To evaluate the effect of failure, we let a random fraction $p$ of the network down and investigate the impact in terms of: (1) repair cost: the total number of nodes that need to update its state to remain valid members of the CRES substrate; (2) reliability: the success rate of CRES paths; a path is considered "successful" if a key value is routed successfully from the source node to the hash node. Figure 4(a) shows that when $m$ is increased (from 1 to 5 and to 10), the repair cost is significantly reduced. Same kind of improvement regarding reliability is observed in Figure 4(b).

## IV. Conclusions

We have proposed CRES, an efficient DHT substrate for data-centric sensor networks. CRES is based on a partition of the network into clusters, in each cluster nodes are assigned unique virtual addresses in such a way that is convenient for content-based routing. Our preliminary simulation results have demonstrated several promising properties of CRES in terms of routing length and effect of failure. We will investigate the issue of node mobility as part of our future work.




(a) Repair cost




(b) Reliability

Fig. 4. Effect of failure


## Acknowledgment

This work was supported in part by the National Science Foundation under award number CNS-0753066 and by UMass Boston's Proposal Development Grant.


## References


[1] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content addressable network," in *ACM SIGCOMM*, San Diego, CA, August 2001, pp. 161–172.

[2] I. Stoica, R. Morris, D. Karger, M. Kaashock, and H. Balakrishman, "Chord: A scalable peer-to-peer lookup protocol for internet applications," in *ACM SIGCOMM*, San Diego, CA, August 2001, pp. 149–160.

[3] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu, "Data-centric storage in sensornets with ght, a geographic hash table," *Mob. Netw. Appl.*, vol. 8, no. 4, pp. 427–442, 2003.

[4] Q. Fang, J. Gao, and L. J. Guibas, "Landmark-based information storage and retrieval in sensor networks," in *IEEE INFOCOM*, 2006.

[5] R. Sarkar, X. Zhu, and J. Gao, "Double rulings for information brokerage in sensor networks," in *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2006, pp. 286–297.

[6] M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron, "Virtual ring routing: network routing inspired by dhts," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 351–362, 2006.

[7] J. Garcia-Luna-Aceves and D. Sampath, "Scalable integrated routing using prefix labels and distributed hash tables for manets," in *Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference on*, oct. 2009, pp. 188 –198.

[8] D. A. Tran and C. Pham, "An information brokerage framework for data-centric sensor networks," University of Massachusetts, Computer Science, Spring 2010.