ORIENTAL JOURNAL OF
COMPUTER SCIENCE & TECHNOLOGY
An International Open Free Access, Peer Reviewed Research Journal
Published By: **Oriental Scientific Publishing Co., India.**
www.computerscijournal.org

ISSN: 0974-6471
April 2016,
Vol. 9, No. (1):
Pgs. 17-23

Constructing Well-Organized Wireless Sensor Networks with Low-level Identification

M.RAJESH [1] and J.M GNANASEKAR [2]

[1]Research Scholar, Department of Computer Science & Engineering,
St.Peter's University Chennai, India.
[2]Professor & Head Department of Computer Science & Engineering,
Karpaga Vinayaga College of Engineering & Technology, India.
*Corresponding author E-mail: rajesmano@gmail.com

http://dx.doi.org/10.13005/ojcst/901.04

(Received: March 11, 2016; Accepted: April 20, 2016)

ABSTRACT

In most distributed systems, naming of nodes for low-level communication leverages topological location (such as node addresses) and is independent of any application. In this paper, we investigate an emerging class of distributed systems where low-level communication does not rely on network topological location. Rather, low-level communication is based on attributes that are external to the network topology and relevant to the application. When combined with dense deployment of nodes, this kind of named data enables *in*-network processing for data aggregation, collaborative signal processing, and similar problems. These approaches are essential for emerging applications such as sensor networks where resources such as bandwidth and energy are limited. This paper is the first description of the software architecture that supports named data and in-network processing in an operational, multi-application sensor-network. We show that approaches such as in-network aggregation and nested queries can significantly affect network traffic. In one experiment aggregation reduces traffic by up to 42% and nested queries reduce loss rates by 30%. Although aggregation has been previously studied in simulation, this paper demonstrates nested queries as another form of in-network processing, and it presents the first evaluation of these approaches over an operational testbed.

Keywords: Topological; network processing; Multi-application.

INTRODUCTION

In most distributed systems, naming of nodes for low-level communication leverages topological location (such as node addresses) and is independent of any application. Typically, higher-level, location-independent naming and communication is built upon these low-level communication primitives using one or more levels of (possibly distributed) binding services that map higher-level names to topological names and sometimes consider application-specific requirements. An example of this is the Internet where IP addresses provide the low-level names suitable for routing. IP addresses are assigned topologically: the addresses for nodes that are topologically proximate are usually drawn from the same address prefix[18]. (By topology, we mean logical connectivity as distinct from physical geography.) This topological assignment is essential for scaling the routing system and was carried forward into IPv6[30].

DNS provides a text-based hierarchical node naming system[26] that is implemented using IP. Above this system, the web and search engines provide a document and object naming system, and content distribution networks add geographic or application-level constraints. As an alternative, systems such as Jini[35] and INS[1] layer different approaches for resource discovery above IP for networks of devices. In this paper, we investigate an emerging class of distributed systems where low-level communication does not rely on network topological location. Rather, low-level communication is based on names that are external to the network topology and relevant to the application; names can be based on capabilities such as sensor types or geographic location. Such an approach to naming allows two kinds of efficiencies. First, it eliminates the overhead of communication required for resolving name bindings. Second, because data is now self-identifying, it enables activation of application-specific processing inside the network, allowing data reduction near where data is generated. These two benefits do not apply to the Internet as a whole, where, by comparison, bandwidth is plentiful, delay is low, and throughput (router processing capability) is the primary constraint. Technology trends suggest, however, that these conditions are reversed in wireless sensor networks. Sensor networks are predicated on the assumption that it will be feasible to have small form factor devices containing significant memory resources, processing capabilities, and low-power wireless communication, in addition to several on-board sensors. In sensor networks processing time per bit communicated is plentiful (CPUs are fast and bandwidths low), but bandwidth is dear. For example, in one scenario Pottie and Kaiser observe that 3000 instructions could be executed for the same energy cost of sending a bit 100m by radio[29]. This environment encourages the use of computation to reduce communication.

In that context, fewer levels of naming indirection and the use of in-network, application-specific message processing (as opposed to opaque packet forwarding) are essential to the designof sensor networks. Our thesis, then, is that the resource constraints of wireless sensor networks can be better met by an attribute-based naming system with an external frame of reference than by traditional approaches. There have been many attribute-based naming schemes, but most build over an underlying topological naming scheme such as IP[28, 10, 6, 38, 4, 27, 1, 20, 22]. Multiple layers of naming may not be a bottleneck with a few or even tens of nodes, but the overhead becomes unreasonable with hundreds or thousands of nodes that vary in availability (due to movement and failures). However, constrained, application-specific domains such as sensor networks can profit by eliminating multiple layers and naming and routing data directly in application-level terms. Efficient attribute naming is based on external frames of reference such as pre-defined attributes and geography. Pre-defined sensor types reduce the levels of run-time binding and geographic-aided routing reduces resource consumption.

In addition to attribute-based naming, application-specific, in network processing is essential in resource-constrained sensor networks. As suggested by the above trade-off between computation and communication, application-specific caching, aggregation, and collaborative signal processing should occur as close as possible to where the data is collected. Such processing depends on attribute-identified data to trigger application-specific filters, pre-defined attributes and data types to allow pre-deployment of these filters, and hop-by-hop processing of the data. This kind of processing is similar to Active Networks[34], but differs by operating in the constrained, bandwidth-poor environment of sensor networks where an integrated, application-specific solution is appropriate.

As an illustration of attribute-based naming and in-network processing in a sensor network, consider a wireless monitoring system with a mixture of light or motion sensors (constantly vigilant at low-power), and a few higher-power and higher-bandwidth sensors such as microphones or cameras. To conserve energy and bandwidth the audio sensors would be off (or not recording) at most times, except when triggered by less expensive light sensors. Instead this computation can be distributed throughout the network. Queries (user requests) are labeled with sensor type (audio or light) known to the system at design time. Queries diffuse through the network to be handled by nodes with matching sensors in the relevant geographic region. The application will hear from whatever relevant sensors respond. Moreover, the decision of one sensor triggering another can be moved into the network to be handled directly between the light and audio sensors. The alternative Internet-based architecture would have a central directory of active sensors and a central application that interrogates this database, monitors specific sensors, and then triggers others. Our goal is to eliminate the communication costs of maintaining this central information to provide more robust and long-lived networks in spite of changing communications, moving nodes, and limited battery power. (We explore exactly how these approaches work in Section 5 and quantify potential savings in Section 6.) In this paper, we demonstrate that there exists a simple architecture that uses topology-independent naming for low-level communications to achieve flexible, yet highly energy efficient application designs.

The key contributions of this work are therefore:

- Identifying the building-blocks of this architecture, specificallyan attribute-based naming scheme with flexible matching rules grounded in a shared framework of attributes (such as sensor types and geography).
- Showing how this approach to naming enables application specific, in-network processing such as localized data aggregation, and to quantify these benefits in a running system.

In previous work[23], we have discussed the low-level communication primitives that constitute directed diffusion. This work focused on understanding the design space of the network protocols underlying directed diffusion. It also evaluated their performance through simulation, finding that scalability is good as numbers of nodes and traffic increases. However, this work did not develop the software architecture necessary for realizing attributes and in-network processing in an operational system (for example, it employed a simplified attribute scheme and hard-coded aggregation methods). In addition, simulations necessitate approximating environmental effects such as radio propagation, and many parameters of those simulations were not set to match the sensor networking hardware that is only now becoming available. By contrast, this paper evaluates the design questions concerning naming and in-network processing encountered in deploying a sensor network, and it presents the first experimental results of data diffusion in a testbed (reflecting the details of an implementation such as non-idealized radios, propagation, MAC protocols, etc.).

Table 1: Simulation results for tcp and udp flows

Traffic type	UDP (node 1 to 9)	6 TCP flows
Aggregate throughput (Kbps)	212	205
Average end-to-end delay (s)	0.0687	0.627
RTT(s)	0.132	1.120
Dropped packets / s	0	1.276

Average end-to-end delay of the two UDP flows

Table 2 : Congestion header

r_p (sender’s permit arriving rate)
ci (sender’s control interval)
fb (feedback)

Numerous early systems have developed attribute-based naming systems, for general use[28, 10, 6], as an approach to software design[9, 4, 27, 17, 25] and for sensor networks[1, 22]. Our work is unique in that it replaces rather than augments the underlying networking routing layers, and that it provides matching rules that allow efficient implementation and yet are expressive enough to cover a wide range of applications, and provides in-network processing.

Medium contention and its impact

Two UDP flows have the same sending rate. We gradually increase the sending rate until there are some packet losses for collisions in the 300 seconds simulation. The simulation results are summarized in Table I where the performances are also included when there are six TCP flows for comparison.

Thus the optimum aggregate sending rate is 212 Kbps H" 25 pkt/s given that there is no packet loss due to collisions. And RTTo = 0.132s and Keachtcp = 0.55pkt/RTT when N = 6. The more TCP flows there are, the smaller Keachtcp. In summary, to provide high throughput, short delay and stable performance with few packet collisions, window-based congestion control is not appropriate for WANET. And we will design an efficient rate-based congestion control algorithm in the following sections.

Hybrid rate-based congestion control

Our proposed hybrid rate-based end-to-end control protocol (HRBCC) is aimed to work around the optimal point in the sense that the channel capacity is fully utilized while no congestion is caused. As mentioned earlier, in mobile ad hoc networks, the optimal window size for TCP is very small, typically less than 5 in packets and even less than 1. Hence any change in congestion window may result in large throughput oscillation in each RTT, failing to stabilize the throughput. For this reason, RBCC, unlike TCP, employs rate based control, in order to stabilize the throughput of each flow while making full use of the available bandwidth.

Protocol Overview

HRBCC controls the sending rate of each flow by the explicit feedback carried in the ACKs. And each HRBCC packet carries a congestion header as shown in Table II, which is used to communicate a flow's state to intermediate nodes and the feedback from the intermediate nodes further to the destination. The field rp is the sender's current permit arriving rate, and the field ci is the sender's currently used control interval. They are filled in by the sender and never modified in transit. The last field, fb, is initiated by the sender and all the intermediate nodes along the path may modify it to directly control the packet sending rate of the sources.

The HRBCC sender maintains an estimate of the round trip time rtt and accordingly calculates the control interval ci. It also adjusts the permit arriving rate rp according to the explicit feedback in the ACKs. Each time the sender transmits a packet, it attaches a congestion header to the packet with the latest rp and ci. All the nodes along the flow's path, including the HRBCC sender and receiver, keep monitoring the channel busyness ratio rb, and calculate the feedback accordingly.

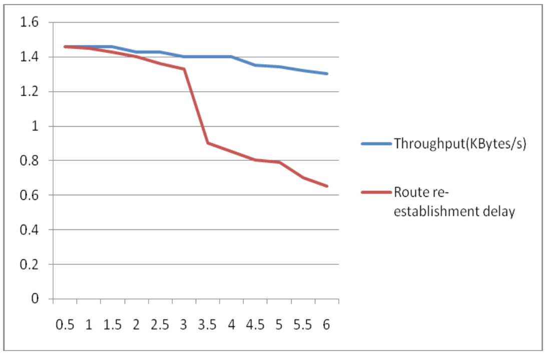

Fig.1: Throughput and retransmissions

Then, according to the rules specified in 5.3, the node will decide whether and how to modify the fb field in the congestion header. The more congested node later in the path can overwrite the fb field in the congestion header. Ultimately, the packet will contain the feedback from the bottleneck node along the path. When the feedback reaches the receiver, it is returned to the sender in an ACK packet, and the sender updates its permit arriving rate rp accordingly. The updated arriving rate rp is then used by the sender to control the permit arriving rate at the leaky bucket.

A HRBCC receiver is similar to a TCP receiver except that when acknowledging a packet, it copies the congestion header from the data packet to its ACK.

Hybrid feedback based approach

From the preceding discussion, it is clear that treating route failure as congestion and invoking congestion control is not advisable as congestion control and route failure are disparate phenomena which have to be handled independently and separately. Therefore, we propose a scheme by which the source is informed of the route failure so that it does not unnecessarily invoke congestion control, and it refrains from sending any further packets until the route is restored. Feedback based schemes for TCP have already been proposed in the form of explicit congestion notification (ECN)[8] for fixed networks and EBSN[3] in cellular networks. ECN is used in order to hasten the process of congestion detection while in our case we use feedback to explicitly notify the source of route failure. EBSN is used in cellular networks to handle transient wireless errors using the reliability of the fixed portion of the network. As we do not have a reliable backbone in case of ad-hoc networks, neither of these methods is directly applicable in our case. Therefore, we propose a feedback based scheme for handling route failures in ad-hoc networks termed as HRBCC which is described below:

Consider, for simplicity, a single bulk data transfer session, where a source MH is sending packets to a destination MH. As soon as the network layer at an intermediate MH (hence forth referred to as the failure point or FP) detects the disruption of a route due to the mobility of the next MH along that route, it explicitly sends a Route Failure Notification (RFN) packet to the source and records this event. Each intermediate node that receives the RFN packet invalidates that particular route and prevents incoming packets intended for that destination from passing through that route. If the intermediate node knows of an alternate route to the destination, this alternate route can now be used to support further communication and the RFN is discarded. Otherwise, the intermediate node simply propagates the RFN towards the source.

We have conducted simulation experiments based on HRBCC and have compared it with basic TCP. Our results, which are described in the next section, suggest that HRBCC results in significant improvements in throughput as well as good put.

Simulated performance

Simulation Parameters: For our experiments, we used a fixed packet size of 200 bytes and data rate of 12.8 Kbps (on the wireless channels), which are reasonable values for wireless networks. The number of hops between the source and destination is set to 10 and the corresponding window size is 4 Kbytes (20 packets). We have assumed that the network in our simulation does not suffer from congestion. Consequently, any packet loss is attributed to route failure only. The input parameters to the simulation are the failure rate and route re-establishment delay each run of the simulation is for a period of 100 seconds and is repeated 10 times. The averages of the ten values are reported.

The simulation is event-driven and proceeds as follows: Transmission of a packet leads to a SEND event at the source which in turn triggers an ACK event and a TIMEOUT event. The timestamp of the ACK event is calculated as follows: tACK = tSEND + delay based on number of hops, data rate and packet size + random variance The timestamp of the TIMEOUT event is based on TCP's timeout estimation mechanism. Depending on the conditions as described below, one of the above events (ACK/TIMEOUT) will be valid and the other event is then discarded. In order to simulate the reconfiguration of the network due to mobility, route failure events are periodically generated. The route is reestablished after a delay corresponding to the

RRD. Based on the current time instant, location and duration of the failure and the timestamp of a SEND event, we can determine whether a packet reached the destination successfully and if an ACK successfully reached the source. Once we have determined which packets are lost," their ACK events are then deleted from the event queue.

Over this model, we implemented basic TCP and the proposed HRBCC. We then ran the simulation for both cases using different values of failure interval and route reestablishment delay (RRD). It was ensured that both TCP and HRBCC were simulated under the same conditions.

CONCLUSIONS

This paper has described an approach to distributed systems built around attribute-named data and in-network processing. By using attributes with external meaning (such as sensor type and geographic location) at the lowest levels of communication, this approach avoids multiple levels of name binding common to other approaches. Attribute-named data in turn enables in-network processing with filters, supporting data aggregation, nested queries and similar techniques that are critical to reduce network traffic and conserve energy. We evaluated the effectiveness of these techniques by quantifying the benefits of in-network processing for data aggregation and nested queries. In one experiment we found that aggregation reduces traffic by up to 42% and nested queries reduces loss rates by 15–30%. Although aggregation has previously been studied in simulation, these experiments are the first evaluation of these techniques in an operational testbed. These approaches are important in the emerging domain of wireless sensor networks where network and power resource constraints are fundamental.

REFERENCES

1. Zhiqiang Shi, Ionescu, D., Dongli Zhang, "A Token Based Method for Congestion and Packet Loss Control " Latin America Transactions, IEEE (Revista IEEE America Latina) (Volume:11 , Issue: 2) March 2013.
2. D. Miorandi, E. Altman, and G. Alfano, "The impact of channel randomnesson coverage and connectivity of ad hoc and sensor networks, *IEEE Trans. Wireless Commun.*, **7**(3): 1062–1072 (2008).
3. M. Haenggi, "A geometric interpretation of fading in wireless networks: theory and applications," *IEEE Trans. Inf. Theory*, **54**(12): 5500–5510 (2008).
4. F. Baccelli and B. Blaszczyszyn, "Stochastic geometry and wireless networks volume 2: applications," *Found. Trends Netw.*, **4**(1-2): pp. 1–312 (2009).
5. V. Chandrasekhar and J. G. Andrews, "Uplink capacity and interference avoidance for two-tier femtocell networks," *IEEE Trans. Wireless Commun*, **8**(7): 3498–3509 (2009).
6. K. Huang, V. K. N. Lau, and Y. Chen, "Spectrum sharing between cellular and mobile ad hoc networks: transmission-capacity trade-off," *IEEE J. Sel. Areas Commun.*, **27**(7): 1256–1267 (2009).
7. P. C. Pinto, A. Giorgetti, M. Z. Win, and M. Chiani, "A stochastic geometry approach to coexistence in heterogeneous wireless networks," *IEEE J. Sel. Areas Commun.*, **27**(7); 1268–1282 (2009).
8. A. Rabbachin, T. Q. S. Quek, H. Shin, and M. Z. Win, "Cognitive network interference," *IEEE J. Sel. Areas Commun.*, **29**(2): 480–493 (2011).
9. M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, andM. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE J. Sel. Areas Commun.*, **27**(7), pp. 1029–1046 (2009).
10. S. Weber, J. G. Andrews, and N. Jindal, "An overview of the transmission capacity of wireless networks," *IEEE Trans. Commun.*, **56**(12): 3593–3604 (2010).
11. M. Haenggi and R. K. Ganti, "Interference in large wireless networks,"*Found. Trends*

Netw., **3**: 127–248 (2009).

12. I. Stanojev, O. Simeone, Y. Bar-Ness, and D. Kim, "On the energy efficiency of hybrid-ARQ protocols in fading channels," in *Proc. 2007* IEEE ICC.
13. J. Kim, H. Jin, D. K. Sung, and R. Schober, "Optimization of wireless multicast systems employing hybrid-ARQ with Chase combining," *IEEETrans. Veh. Technol.*, **59**(7); 3342–3355 (2010).
14. F. Wang, A. Ghosh, C. Sankaran, P. Fleming, F. Hsieh, and S. Benes, "Mobile WiMAX Systems: Performance and Evolution," IEEE Comm. Magazine, **46**(10): 41-49 (2008).
15. S. Shin and H. Schulzrinne, "Measurement and Analysis of the VoIP Capacity in IEEE 802.11 WLAN," IEEE Trans. *Mobile Computing,* **8**(9): 1265-1279 (2009).
16. C. So-In, R. Jain, and A.-K. Tamimi, "Scheduling in IEEE 802.16e Mobile WiMAX Networks: Key Issues and a Survey," *IEEE J. Selected Areas in Comm.,* **27**(2): 156-171 (2009).
17. M. Andrews and L. Zhang, "Satisfying Arbitrary Delay Requirements in Multihop Networks," Proc. IEEE 27th Conf. Computer Comm., 116-120 (2008).
18. S. Ganguly, V. Navda, K. Kim, A. Kashyap, D. Niculescu, R. Izmailov, S. Hong, and S. Das, "Performance Optimizations for Deploying VoIP Services in Mesh Networks," *IEEE J. Selected Areas in Comm.,* **24**(11): 2147-2158 (2006).
19. Q. Xue and A. Ganz, "Ad Hoc QoS On-Demand Routing (AQOR) in Mobile Ad Hoc Networks," *J. Parallel and Distributed Computing,* **63**(2): 154-165 (2003).
20. Y. Hwang and P. Varshney, "An Adaptive QoS Routing Protocol with Dispersity for Ad-Hoc Networks," Proc. 36th Ann. Hawaii Int'l Conf. *System Sciences;* 302a (2003).
21. H. Badis and K.A. Agha, "QOLSR, QoS Routing for Ad Hoc Wireless Networks Using OLSR," *European Trans. Telecomm.,* **16**(5): 427-442 (2005).
22. S. Shah and K. Nahrstedt, "Predictive Location-Based QoS Routing in Mobile Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm.,* **2**: 1022-1027 (2002).