World Applied Sciences Journal 29 (Computer Sciences, Engineering and Its Applications): 228-233, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.29.csea.2272

Energy Efficient Reprogramming for Mobile Sensor Network

R. Udayakumar, V. Khanaa and K.P. Thooyamani

School of Computing Science, Bharath University - 73, India

Abstract: In this paper, ReMo, an energy efficient, multihop reprogramming protocol for mobile sensor networks was proposed. Without making any assumptions on the location of nodes, ReMo uses the LQI and RSSI measurements of received packets to estimate link qualities and relative distances with neighbors in order to select the best node for code exchange. Contrary to previous protocols, ReMo downloads pages regardless of their order and facilitating a fast transfer of the code. Our simulation results show significant improvement in reprogramming time and number of message transmissions over other existing protocols under different settings of network mobility.

Key words: Code Dissemination · Network Reprogramming · Mobile Sensor Network

INTRODUCTION

Several sensor networks applications [1], however, require nodes to be mobile. In this paper, ReMo, a reprogramming protocol specifically designed for mobile sensor networks was proposed. Without assuming nodes know their respective locations, ReMo tries to infer relative distance with neighboring nodes and the link qualities with them from parameters measured from received packets.

We address two main issues for code exchange between neighbors [1, 2]. First, nodes try to ascertain which neighbor has the best link quality as well as a higher chance of staying within communication range for a longer duration. Second, a node also tries to ascertain which neighbor has the highest potential for providing pages for download [3]. The primary local goal of ReMo running at a node is to optimally choose a neighbor based on the above two requirements while simultaneously trying to minimize its energy usage by intelligently optimizing the number of control messages transmitted in a neighborhood. Since each node takes local decisions based only on neighborhood information, ReMo can scale to large network sizes and also efficient in terms of speed of propagation [4].

Link Quality and Relative Distance Estimate: Since the mobile nodes are not assumed to know their own location, it is important to determine not only the relative distance between two neighbors but also the link quality between them in terms of the packet reception rate. These two parameters would help a node to select the best neighbor for code downloads [5].

The ReMo Protocol: In ReMo, nodes are not required to be aware of their locations. It also does not make any assumptions on network connectivity being maintained at all times [6, 7].

n this section, we first introduce the design directions of ReMo before describing the working principle of the protocol.

Data Representation: Similar to the data representation of Deluge [2], ReMo divides the code image into fixed sized packets of size Spkt. The protocol uses a basic unit of transfer called a page which is of size N:Spkt where N is a fixed number of packets.

Breaking the code image into pages enables pipelining the transfer of the file over multiple hops across the network. A version number is used to distinguish between different code updates and must be monotonically increasing to maintain an order for all updates. A node needs to compare version numbers to decide on whether it requires an update [8].

Page Download Potential (PDP): The page download potential ω_j^i of node i with respect to node j as a measure of the potential that j has in providing pages to i. Let the code image C_{img} consist of *k* pages, such that.

Corresponding Author: R. Udayakumar, School of Computing Science, Bharath University - 73, India.

$C_{img} = \{P_1; P_2; \ldots; P_k\}$,

where P_l is the lth page. Let S_i and S_j denote the set of pages of node I and node j, respectively. Then the page download potential ω_j^i of i with respect to j is denoted by $\omega_j^i = |S_j - S_i| / C_{img}$ where $S_j - S_i$ is the set difference between the page sets of node j with node i. Thus, for each node i with σ neighbors, the page download potential vector Ω is denoted by:

$$\Omega = \{\omega_1^{i,} \ \omega_2^i \ \ \omega_3^i, \ldots\ldots \ \omega_\sigma^{\ i} \ \}$$

Neighbor Link Profile (NLP): Each node i snoops in its neighborhood for packets transmitted by other nodes in order to build a Neighbor Link Profile vector $\Phi_{i.}$ φ_j^i which is an element of Φ_i, represents an entry of the NLP of node *i* with respect to node *j*. It is a 2-tuple $<lq_j^i; dm_j^i>$, where lq_j^i is a normalized representation □[0,1] of the link quality with node *j* and dm_j^i is a direction of motion indicator. lq_j^i is calculated based on the average of the LQI values of packets received from node *j*.

On the other hand, dm_j^i is calculated based on RSSI values of multiple packets from the same node. If no packet is heard from an existing neighbor in the NLP in the last time slot, it is marked as *stale*. Moreover, a *stale* neighbor is again made fresh when a packet is received from it [9].

Probability of Metadata Broadcast: One of the central features on which ReMo is based is an adaptive metadata advertisement scheme. In this section, we formulate β_t, which is the probability of metadata broadcast [3] by a node at each time slot t. Each node dynamically updates β_t at each time slot based on gathered observation at the previous time slot. Not only is the computation of this transmission probability based on the presence of new code information in the neighborhood, but also on the relative density of the current neighborhood. Thus, each node tries to proactively ascertain if there is any neighbor with new code and also control the level of gossip in order to avoid collisions and packet loss [10].

Let N_t^d and N_t^s, respectively, denote the number of *different* and *similar* metadata advertisements heard by a node during slot *t*. Moreover, let A_t denote the sum of all advertisement messages heard by the node in slot t and is denoted by:

$$A_t = N_t^d + N_t^s$$

Thus A_t, which is proportional to the number of nodes in the neighborhood, is representative of the average node density. More importantly, nodes keep track of the number of neighbors that are advertising the same metadata, N_t^s. They, accordingly, decrease their transmission probability in order to minimize redundantly broadcast metadata in the neighborhood. Thus, for each slot *t*, nodes gather these information about their neighbors and update their advertisement probability for the next slot $t + 1$.

Protocol Description: In ReMo, the goal of a node is to periodically keep its neighbors updated on the version of its code and its location information.

Broadly, each node lies in either of three major states, viz., Advertise(ADV), Receive(RX) and Transmit(TX). Fig. 1 shows the detailed state transition diagram of the protocol.

In the ADV state, a node performs important functions like periodic advertisement of code metadata, neighborhood assessment and optimal decision making for different actions like choosing an appropriate neighbor to download code from or modifying its advertisement transmission rate based on dynamic information about its current neighbors. In this state, a node broadcasts an advertisement message Mdata containing some meta information. Mdata is primarily comprised of two components, viz.: 1) Version number and 2) Downloadable data information.

The Request messages are of two types: 1) A Half Request (HReq) message and 2) A Full Request (FReq) message.. When a node is sure that the destination neighbor is close enough for a reliable page download, it sends a FReq message, whereas it sends a HReq message when it is unsure of the reliability. In the latter case, it is upto the neighbor to respond with the requested page or ignore the HReq message. The choice of a neighbor to transmit a Request message to, is based on which neighbor has a high page download potential as well as a sufficiently good link quality.

If a node in the ADV state finds that it has a nonempty PDP and NLP list and it needs code to download, it would transit to the RX state at the next timer expiry and send out an appropriate Request message targeted at the chosen neighbor and wait for the requested data to be downloaded [11, 12].

When a node receives a FReq message in the ADV state, it transits to the TX state and transmits all the packets of the first page of the requested page vector. The recipient node sends a DataACK message for the page sent. On receiving this acknowledgment, the sender transmits the data packets of the next page in the requested vector provided the link quality of the

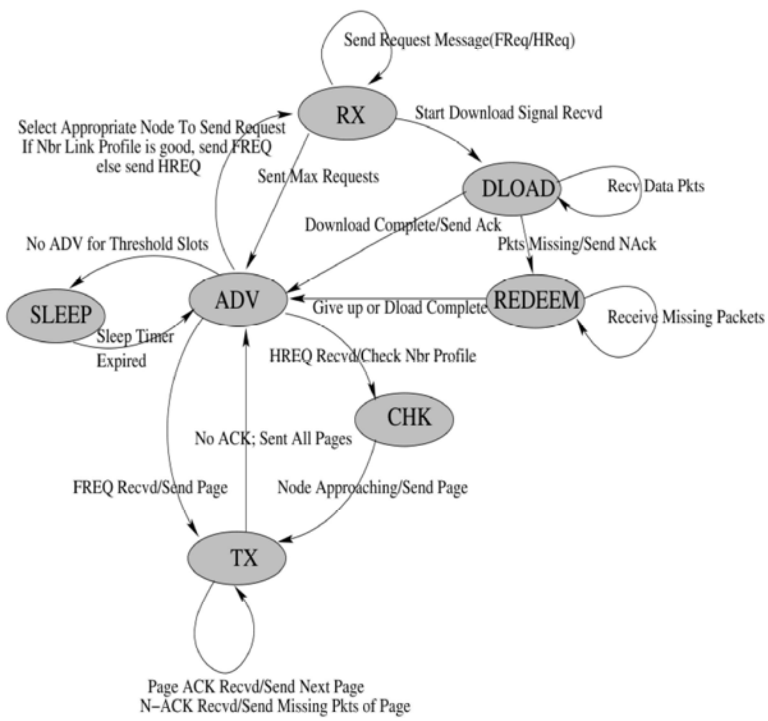

Fig. 1: ReMo state transition diagram

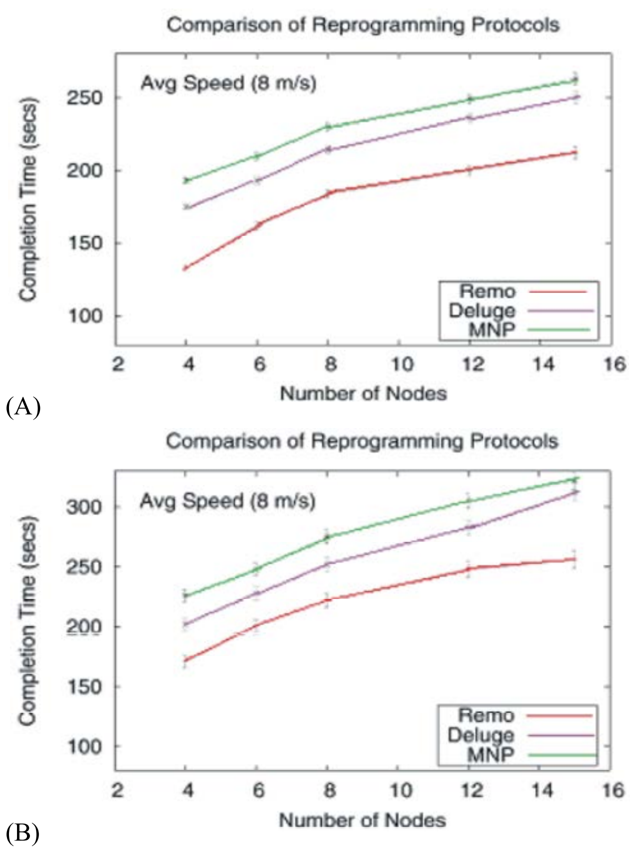

Fig. 2: Completion Time Of Deluge, MNP And Remo In A Mobile Sensor Network., (A) 30 KB and (B) 40 KB

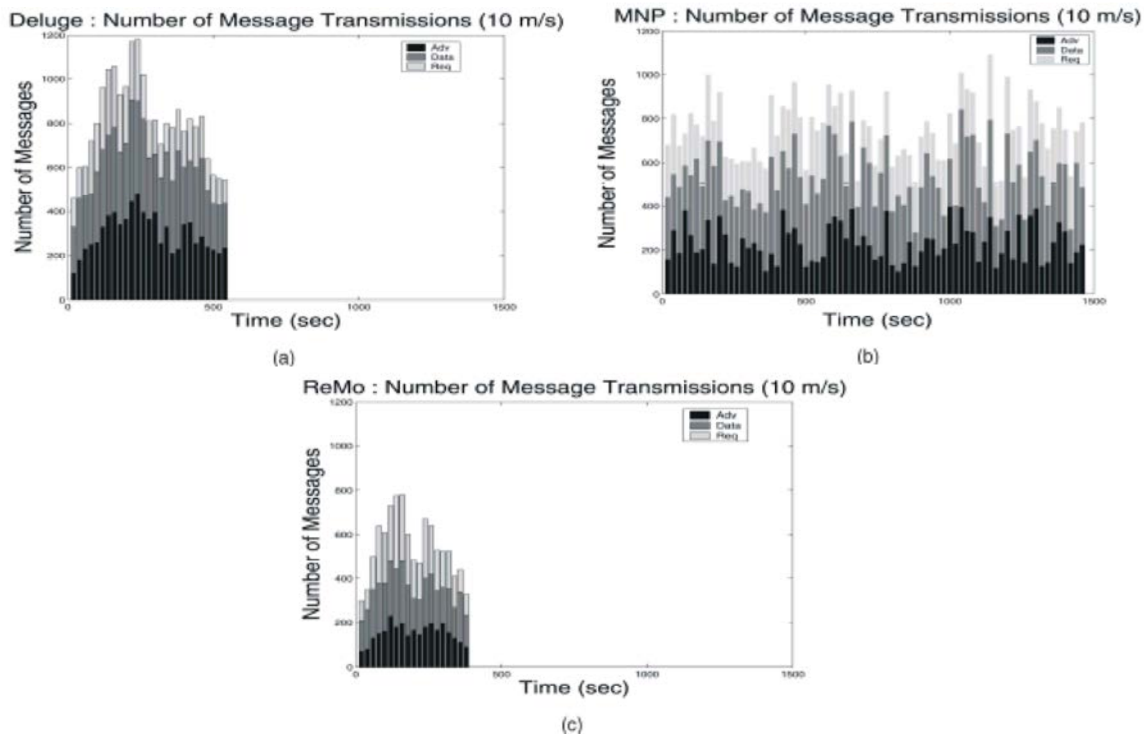

Fig. 3: Number of transmitted messages (120 nodes). (a) Deluge, (b) MNP and (c) ReMo

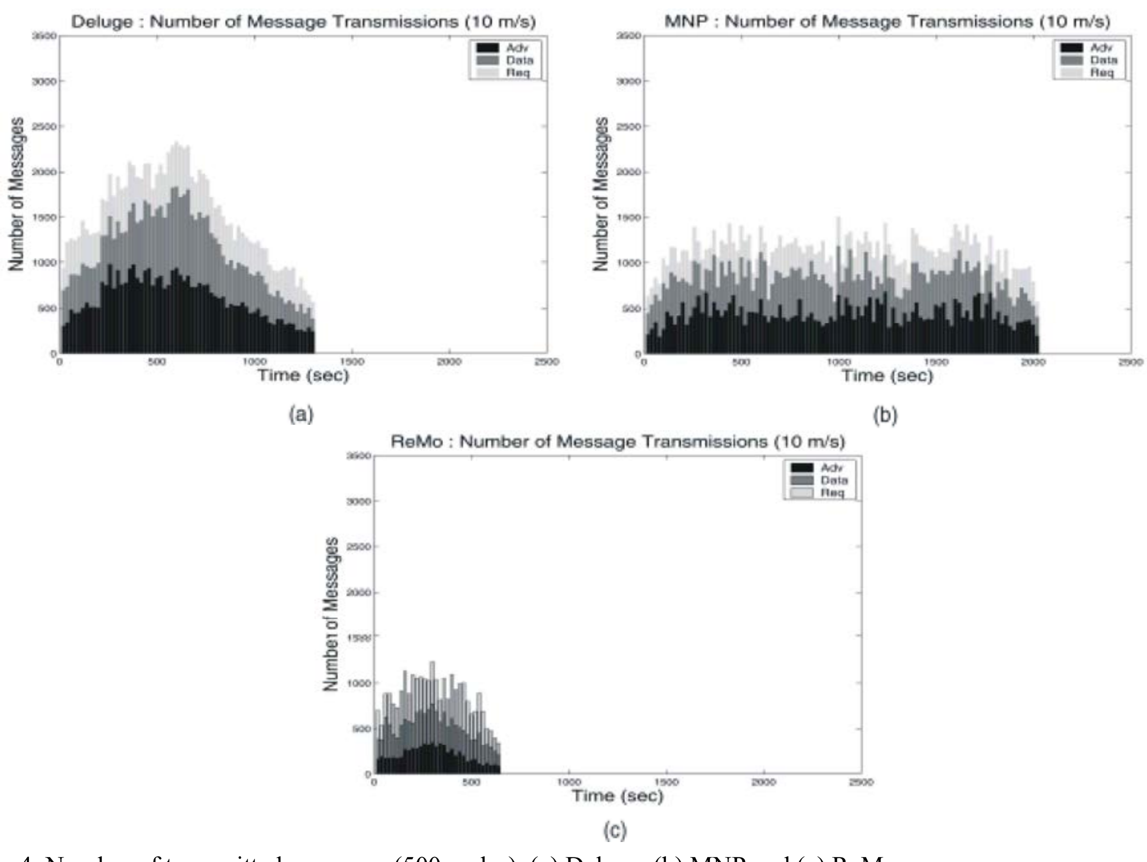

Fig. 4: Number of transmitted messages (500 nodes). (a) Deluge, (b) MNP and (c) ReMo

destination node (as per the last measured estimates) and the RSSI of the received DataACK packet is above the required threshold. However, if no acknowledgment arrives, or the sender transits to the ADV state and starts broadcasting its metadata with probability β_0. A recipient node transits to the REDEEM state to download missing packets of a page. Upon receiving a HReq message, a sender node migrates to an intermediate CHECK state to decide whether the recipient node is suitable for a page transfer by evaluating the link quality and the RSSI of the received HReq packet. If the link quality is satisfactory and the node is approaching, then the page is transmitted. The protocol messages with their essential fields are defined as follows:

- Advertise Message: (a) ImageName, (b) Version Number, (c) Image Size and (d) Image Page Bit vector.
- HReq Message: (a) Requested Version, (b) ImageName, (c) Sequence Number, (d) Destination Address, (e) RSSI of Recvd Adv and (f) Requested Vector of Pages.
- FReq Message: (a) Requested Version, (b) Image-Name, (c) Sequence Number, (d) Destination Address and (e) Requested Vector of Pages.
- Data Message: (a) ImageName, (b) Version Number, (c) Page Number, (d) Packet Index, (e) Data Size and (f) Data.

Performance Evaluation: The performance of ReMo is evaluated in comparison with Deluge [2] and MNP [4] using the packet level network simulator GloMoSim [5]. All the three protocols have been implemented at the application layer of GloMoSim. Network reprogramming time and the total number of packets transmitted to achieve that were the primary metrics of measurement for evaluating the protocols.

Reprogramming Completion Time: In this section, we focus on the completion time for transferring images of different sizes from a source node into the entire network. Each page is of size 1 KB comprising of 16 packets of size 64 bytes each. We vary different network parameters and study their effects on the transfer time of Remo against existing standard code update protocols.

Number of Message Transmissions: We look into the number of messages transmitted by each protocol in fixed sized time windows of 20 seconds for the entire duration of the code update. Fig. 3 depicts the message transmission distribution for each protocol.

In Fig. 4, we look at the number of message transmissions in a denser network of 500 nodes moving at the same average speed of 10 m/s _ 10 percent.

MNP has more latency for the entire completion of code transfer but transmits less number of messages than Deluge in each 20 second window. ReMo completes the transfer in lesser time and also transmits almost half the total number of messages as Deluge does in each time window.

Thus, the lower average message transmission rate also indicates that ReMo is more energy efficient than Deluge or MNP in a mobile environment.

CONCLUSIONS

In this paper, we presented ReMo, an energy-efficient reprogramming protocol specifically suited for mobile sensor networks. We showed how the existing reprogramming paradigm for static networks fails to adapt to a mobile scenario. The protocol takes advantage of the mobility of the nodes by having them download pages out of order, thus expediting the download process. It also smoothly adapts its periodic metadata advertisements to cope with the constantly varying node density of the mobile environment and suppress redundant transmissions as much as possible, thereby saving valuable energy. Moreover, it constantly monitors its neighborhood and maintains the link quality statistics of its neighbors so that it can choose the best current neighbor for code exchange, thus, further improving on the performance and energy savings. Our comparative results indicate significant improvements in completion time of reprogramming the whole network and the number of messages transmitted over existing reprogramming protocols like Deluge and MNP.

REFERENCES

1. Chong, C. and S.P. Kumar, 2003. “Sensor Networks: Evolution opportunities and Challenges,” Proc. IEEE, 8: 1247- 1256.
2. Hui, J.W. and D.E. Culler, 2004. “The Dynamic Behavior of A Data Dissemination Protocol for Network Programming at Scale,” Proc. Int’l Conf. Embedded Networked Sensor Systems, pp: 81-94, http://dblp.uni-trier.de/db/conf/sensys/ sensys2004. html.
3. Kyasanur, P., R.R. Choudhury and I. Gupta, 2006. “Smart Gossip: An Adaptive Gossip-Based Broadcasting Service for Sensor Networks,” Proc. Conf. Mobile Adhoc and Sensor Systems, pp: 91-100.

4. Wang, L., 2004. “MNP: Multihop Network Reprogramming Service for Sensor Networks,” Proc. Int’l Conf. Embedded Networked Sensor Systems, pp: 285-286.
5. Zeng, X., R. Bagrodia and M. Gerla, 1998. “Glomosim: A Library for Parallel Simulation of Large-Scale Wireless Networks,” Proc. Workshop Parallel and Distributed Simulation, pp: 154-161.
6. De, P., Y. Liu and S.K. Das, 2008. “ReMo: An Energy Efficient Reprogramming Protocol for Mobile Sensor Networks,” Proc. IEEE Int’l Conf. Pervasive Computing and Comm., pp: 60-69.
7. Saravanan, T. and R. Udayakumar, 2013. Comparision of Different Digital Image watemarking techniques, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1684-1690.
8. Saravanan, T. and R. Udayakumar, 2013. Optimization of Machining Hybrid Metal matrix Composites using desirability analysis, Middle-East Journal of Scientific Re., ISSN:1990-9233, 15(12): 1691-1697.
9. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Improving Web Information gathering for personalised ontology in user profiles, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1675-1679.
10. Thooyamani, K.P., V. Khanaa and R. Udayakumar, 2013. Detection of Material hardness using tactile sensor, Middle-East Journal of Scientific Research, ISSN:1990-9233, 15(12): 1713-1718.
11. Udayakumar, R., A. Kumaravel, Rangarajan, 2013. Introducing an Efficient Programming Paradigm for Object-oriented Distributed Systems, Indian Journal of Science and Technology, ISSN: 0974-6846, 6(5S): 4596-4603.
12. Udayakumar, R., V. Khanaa and K.P. Kaliyamurthie, 2013. Performance Analysis of Resilient FTTH Architecture with Protection Mechanism, Indian Journal of Science and Technology, ISSN: 0974-6846, 6(6): 4737-4741.