Journal of King Saud University – Computer and Information Sciences (2015) 27, 402–415

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Rollback recovery with low overhead for fault tolerance in mobile ad hoc networks

Parmeet Kaur Jaggi [a,*], **Awadhesh Kumar Singh** [b]

[a] *Department of Computer Science, Jaypee Institute of Information Technology, Noida, India*
[b] *Department of Computer Engineering, National Institute of Technology, Kurukshetra, India*

Received 14 November 2013; revised 4 February 2014; accepted 13 March 2014
Available online 18 June 2015

KEYWORDS

Ad hoc network;
Mobile backbone;
Rollback recovery;
Checkpointing;
Message logging;
Routing protocols

Abstract Mobile ad hoc networks (MANETs) have significantly enhanced the wireless networks by eliminating the need for any fixed infrastructure. Hence, these are increasingly being used for expanding the computing capacity of existing networks or for implementation of autonomous mobile computing Grids. However, the fragile nature of MANETs makes the constituent nodes susceptible to failures and the computing potential of these networks can be utilized only if they are fault tolerant. The technique of checkpointing based rollback recovery has been used effectively for fault tolerance in static and cellular mobile systems; yet, the implementation of existing protocols for MANETs is not straightforward. The paper presents a novel rollback recovery protocol for handling the failures of mobile nodes in a MANET using checkpointing and sender based message logging. The proposed protocol utilizes the routing protocol existing in the network for implementing a low overhead recovery mechanism. The presented recovery procedure at a node is completely domino-free and asynchronous. The protocol is resilient to the dynamic characteristics of the MANET; allowing a distributed application to be executed independently without access to any wired Grid or cellular network access points. We also present an algorithm to record a consistent global snapshot of the MANET.

1. Introduction

Mobile ad hoc networks (MANETs) have extensively enhanced the wireless networks as they eliminate the need for any fixed infrastructure, in the form of base stations, routers etc. These networks are formed by nodes that communicate over wireless links without the control of any central or fixed administration. Each node performs the dual roles of a node as well as a router. As MANETs are self organizing and rapidly deployable, these have been frequently used for communication in places where it is either expensive or difficult to install network infrastructure, such as in battlefields, search-and-rescue or space exploration. In addition, the computational power of mobile computing platforms of the present day exceeds that of the workstations from a few years ago. The explosive and continuing growth of wireless devices and networks along with their widespread availability provides

* Corresponding author.

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2014.03.022
1319-1578

a thrust for understanding and utilizing the computing potential of mobile ad hoc networks. These networks are increasingly being used in collaboration with LAN/WAN scenarios, for parallel processing systems, as a means of expanding the computing capacity of existing networks such as cellular mobile systems and even for implementing mobile Grid computing systems (Darby, 2010; Wang et al., 2006; Rao et al., 2006; Jipping, 2001).

A variety of lightweight, distributed applications can be executed successfully on mobile ad hoc platforms without the support of fixed infrastructure. These applications include mobile agents providing location-aware services; local and collaborative processing of sensor data collected from a number of MHs, update of maps in real-time battle scenarios etc. Other applications include collaborative mobile gaming, context-aware applications for internetworked vehicles, bio-informatics and other scientific applications; especially in remote areas where access to the wired network is infeasible (Darby and Tzeng, 2010). Smart phones having high computational capabilities along with laptops and Personal Digital Assistants (PDAs) may be used for creating computing clouds in trains, colleges etc. Such clouds could be used to calculate weather forecasts for passengers at their destination using environmental data from public sensing systems, cracking of encryption codes, development of mobile health care and education applications besides participating in scientific projects (Buschin et al., 2012). Some mobile Grid projects, such as the Akogrimo (2010), have explored the use and practical applications of mobile Grid concepts, so that idle resources from a great number of mobile devices could be used for the development of a mobile Grid computing platform.

Due to the vast number of feasible practical applications, the current mobile computing platforms are increasingly being utilized as viable compute resources. However, the nodes in such systems vary greatly in their capabilities such as computation power and battery power and may be susceptible to different types of transient and permanent failures. Therefore, the applications designed to execute on these systems should be fault tolerant so that they can complete successfully without access to any wired Grid or cellular network access points. Checkpointing and rollback recovery have been used widely and effectively to provide fault tolerance for distributed systems in static as well as dynamic environments (Elnozahi et al., 2002). Checkpointing results in a significant performance enhancement as it allows a failed node to resume execution from its latest saved error-free state at the time of recovery and thus avoids the need to restart job execution from the very beginning. In contrast, in the absence of execution checkpointing the failure at one node may cause some other nodes to suspend execution as well, if they are waiting for intermediate results from the failed node. Thus, process failures can lead to severe performance degradation or even total job abortion in the absence of checkpointing.

Though a number of checkpointing and rollback recovery protocols exist for static distributed systems or cellular mobile computing systems, these are not trivially applicable to MANETs as they pose some categorically different set of challenges. Ad hoc wireless networks are characterized by limitation of resources as wireless bandwidth, stable storage, battery power etc. Moreover, the absence of fixed infrastructure generates new problems for ad hoc networks, such as self-routing and a highly unpredictable and dynamic topology. The traditional systems rely on stable storage available at nodes or Base Transceiver Stations, for saving recovery related information (Prakash and Singhal, 1996; Li and Shu, 2005; Tantikul and Manivannan, 2005). On the other hand, the ad hoc environment lacks such capable stations and large data carrying reliable links. The mobile ad hoc networks also have an intrinsic scalability limitation. As the size of the network increases, the performance of the ad hoc network rapidly degrades because a large network with flat structure results in long hop paths which are susceptible to link breaks.

The paper presents a checkpointing and rollback recovery protocol to provide fault tolerance in MANETs. We consider a backbone based mobile ad hoc network which is a type of hierarchical network used for scalability and implementation of efficient protocols (Rubin et al., 2004). Such a network comprises of some particular backbone capable nodes (BCNs) which have powerful radios and are functionally more capable than other ordinary nodes. A virtual backbone is formed by dynamically electing some BCNs to act as backbone nodes (BNs) and forming links between interconnecting neighboring BNs. Each of the other BCNs and ordinary nodes affiliate with one BN such that clusters of nodes are formed with the BN acting as the cluster-head. The communication between the nodes uses the backbone and thus, avoids long hop paths and improves the network performance. Nodes communicate with each other through the BNs to which they are affiliated. If the communicating nodes are affiliated to the same BN, routing is straightforward. However, if they are at remote locations, the routing protocol existing in the network is used for routing through the backbone network. A location based routing protocol, GOAFR+ (Kuhn et al., 2008), has been assumed for the current work. It employs a combination of greedy and face routing to reach destinations using their geographic information. However, our recovery protocol is independent of and can be integrated with any routing protocol for MANETs.

The presented recovery protocol has been designed to handle the specific challenges posed by the dynamic topology and resource constraints of a MANET. The protocol does not add a high overhead to the normal process execution as it takes advantage of the routing protocol already existing in the network. The proposed scheme is an application of cross-layer optimization where the routing protocol existing in the network has been utilized for implementing a message efficient checkpoint and recovery mechanism. The use of the backbone clustered structure provides for added scalability of the protocol. The contributions of the paper can be summarized as follows: (1) The paper presents a checkpointing and rollback recovery protocol which does not assume access to any fixed host or wired network and is therefore appropriate for MANETs. (2) The checkpointing process does not require control messages as the control information required by the protocol is piggybacked on the application messages. (3) The recovery procedure may involve a few control messages; imposing only a low overhead on the network. (4) Rapid and efficient recovery of a mobile node is possible despite the dynamic topology of the network. Even if a mobile node recovers at a location different from the location of its crash, its checkpoint and related information can be located in the network without much delay using the network backbone.