

## A Fault Tolerant Scheme for Ad-hoc Networks

Zhenpeng Xu<sup>1, a</sup>, Weini Zeng<sup>1, b</sup> and Weiwei Li<sup>1, c</sup>

<sup>1</sup>Jiangsu Automation Research Institute, Lianyungang, Jiangsu, 222061, China

<sup>a</sup>email: xuzhenpeng@jari.cn, <sup>b</sup>email: zengweini@jari.cn, <sup>c</sup>email: liweiwei@jari.cn

**Keywords:** Distributed Computing; Cluster; Fault Tolerance; Checkpoint; Rollback Recovery

**Abstract.** Different from the traditional distributed computing system, ad-hoc networks have many new features. The transient failure probability of the computing process increases with the increase of system scale in ad-hoc networks. To reduce the loss of computation upon the transient process failure event, a fault tolerant scheme for Ad-hoc networks is proposed in this paper. The proposal can seamlessly cooperate with the cluster-based multi-channel ad-hoc network system. A low overhead of the communication between the cluster head and related ordinary members is required. The recovery process has no domino effect and the failure process can rollback from its latest local consistent fault tolerant information. By the discussion and contrast, the proposed scheme is able to recover quickly upon the transient failure while only a low additional overhead is required.

### Introduction

Due to the features including dynamic mobility, celerity of setting, autonomy, flexibility of topology and equivalence, the mobile computing network has broad application foreground [1]. An ad-hoc network is a collection of wireless mobile nodes forming a temporary network without the aid of any extra static power processing centralized administration acting as the mobile supporting stations [2-3]. This type of network is not supported by a wired infrastructure like a conventional cellular system [4]. Compared to the traditional wired distributed computing system, the topology of such networks is very dynamic because of host mobility, packet loss, interference, and low power. There are several other features of ad-hoc networks different from wired networks, such as low bandwidth, limited storage space and low computing capacity[5].

**Cluster-based multi-channel ad-hoc network.** As shown in Figure 1, the system consists of a set of mobile hosts (MH), which communicate with each other through wireless channels [6-7]. We assume that wireless channels are all FIFO order.

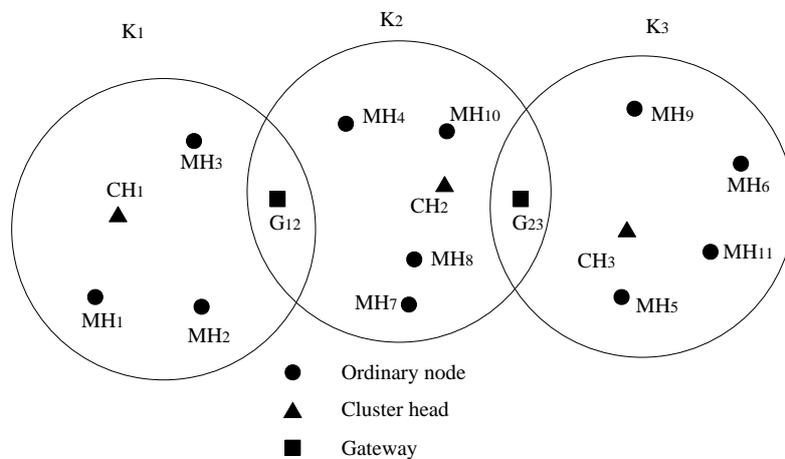


Fig.1. Multi-channel ad-hoc network

In cluster-based network architecture, the network is partitioned into several clusters. The roles of mobile hosts in a cluster can be categorized into cluster head, gateway, and ordinary members. As shown in Figure 1, K<sub>1</sub>, K<sub>2</sub> and K<sub>3</sub> denote three clusters in the wireless network system; CH<sub>1</sub>, CH<sub>2</sub> and CH<sub>3</sub> denote their cluster heads respectively; G<sub>12</sub> and G<sub>23</sub> denote two gateway nodes among three clusters. G<sub>12</sub> is a shared gateway between K<sub>1</sub> and K<sub>2</sub>, and G<sub>23</sub> is a shared gateway between K<sub>2</sub> and K<sub>3</sub>;

$MH_i$  ( $0 < i \leq 11$ ) denote ordinary member nodes; Mobile hosts  $CH_1, G_{12}, MH_1, MH_2$  and  $MH_3$  are in the cluster  $K_1$ ; Mobile hosts  $CH_2, G_{12}, G_{23}, MH_4, MH_7, MH_8$  and  $MH_{10}$  are in the cluster  $K_2$ ;  $CH_3, G_{23}, MH_5, MH_6, MH_9$  and  $MH_{11}$  are in the cluster  $K_3$ . The processes of mobile hosts follow the piecewise deterministic (PWD) model and a reliable communication protocol can be provided[2].

### Related works

The fault tolerance for the mobile computing systems has received tremendous interests in recent years [5]. However, the computing systems are almost modeled with the static base nodes acting as the mobile supporting stations (MSS) with extra processing power and storage capabilities. In this model, a MH's disk storage cannot be considered stable. Park and Kim developed a checkpointing recovery services for a mobile computing system based on the ad-hoc network environment, which tries to reduce disk access frequency for saving recovery information, and also the amount of information saved for recovery [8]. Ono Masakazu and Hiroaki Higaki proposed a novel checkpoint protocol for ad hoc networks. Here, a checkpoint request message is delivered by flooding. State information of a mobile computer is carried by this message and stored into neighbor mobile computers. A candidate of a lost message is detected and stored by intermediate mobile computer on its transmission route [8].

### Fault tolerant scheme for Ad-hoc network

**Checkpoint.** The checkpointing scheme is managed by the local cluster. Cluster head  $CH_i$  must maintain several variables, including a checkpoint index, an ordinary node queue, and a variable storing the number of reply messages. The initial values of the checkpoint index and ordinary node queue are set to zero and null respectively. When the checkpoint period time  $T$  is up, cluster head  $CH_i$  firstly delivers timing parameters of the interval with the checkpoint request to all cluster members through a broadcasted beacon packet in the coming beacon interval  $I_n$ , and the beacon interval  $I_n$  is called the first checkpoint beacon interval.

Upon receipt of the timing parameters of the interval with the checkpoint request, those MHs which intent to data communication, just as normal, then still transmits ATIM packets in ATIM window to inform the receiver keeping awoken and demand the required bandwidth from cluster head. The receiver then replies an ATIM-ACK packet to sender immediately to confirm the communication. Cluster head  $CH_i$  overhears the demanded bandwidth attached in each ATIM packet and records the bandwidth requirement on its internal table. In the channel assignment phase, firstly, cluster head chooses MH pairs with communication admission. Secondly, cluster head assigns MH pairs to suitable channels according to bandwidth requirements and communication targets. Thirdly, cluster head commands idle or improper MHs switching to checkpointing mode rather than power saving mode. All management information is coded into a packet and is broadcasted to the cluster members.

When MHs (including gateway) have received management information packet, the data transmission phase or the checkpointing phase is invoked. Those MHs which are allowed to communicate, switch to the assigned channel and transmit data. Those MHs which are not allowed or needed to communicate will turn to checkpoint mode and take a new checkpoint rather than turn to power saving mode for energy saving. In the next beacon interval  $I_{n+1}$ , which is called second checkpoint beacon interval, those MHs which had taken the required checkpoint in the previous beacon interval  $I_n$ , will attach a checkpoint reply to ATIM packets to receiver (if MH intends to data communication in this interval), or send the reply to  $CH_i$  directly (if MH does not intent to data communication in this interval). Those MHs which had transmitted data in previous beacon interval  $I_n$ , will automatically take the required checkpoint in this beacon interval  $I_{n+1}$ .

**Logging.** In the normal execution phase, each MH (including cluster head and gateways) pessimistically logs the communication (sending and received) messages related to the gateway and output & input commits for independent rollback recovery of a local cluster.

**Rollback recovery.** When a mobile host incurs a failure event, it stops its computational task instantly. If the failure host is not a cluster head, the host sends an ATIM packet with recovery request

to its cluster head in the ATIM phase. When cluster head receives a recovery request from the failure MH, it broadcasts the recovery request to other mobile hosts in the local cluster, including gateways, in the channel assignment phase. Finally, the local cluster head switches to the rollback recovery phase rather than the data transmission phase and rolls back to its recent consistent checkpoint. When a local MH receives a recovery request from its cluster head in the channel assignment phase, it switches to the rollback recovery phase, rolls back to the recent consistent checkpoint, resumes execution and replays logs in order at proper time. The gateway is not required to rollback if no failure event occurs at its processes. That ensures only the local ordinary mobile hosts and cluster head in the cluster impacted by the failure host. To ensure the consistency with the gateway, the repeat messages handing to the gateway generated by the rollback process will be discarded in the recovery phase.

### Results analysis

For simulation, the network topology is a randomly generated, and each cluster includes three channels and 20 or 40 MHs. The supported rate of each cluster is 2.5 Mbps. As shown in the Figure 2, the power consumption of CMMP integrated with the proposal increases slowly. And the power consumption still increases slowly with the decreasing of the checkpoint interval.

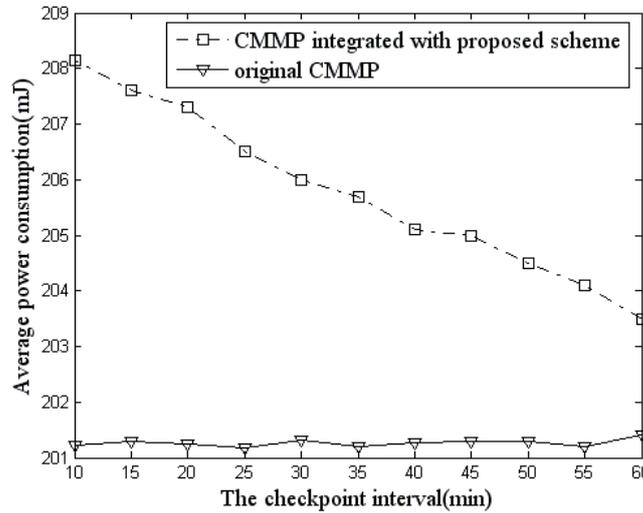


Fig. 2. Average power consumption in failure-free

With a fixed interval  $T=900s$ , the comparison of execution time of benchmark in the cluster-based multi-channel environment is given in Figure 3.

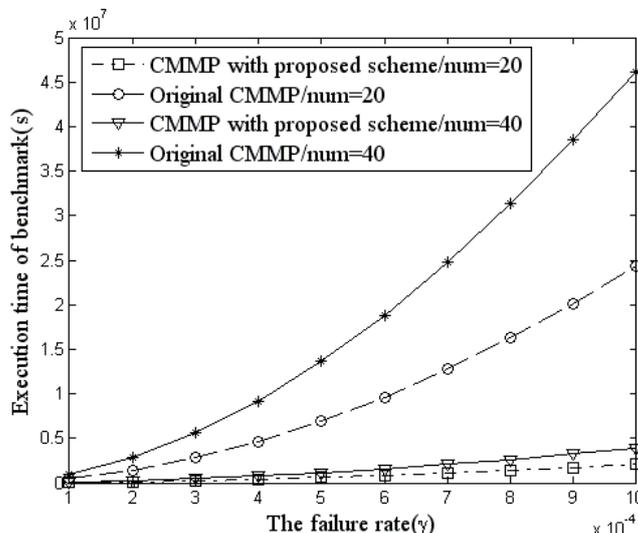


Fig. 3. Execution time with different transient failure rate

As shown in the Figure 3, the execution time of benchmark in CMMP protocol increases significantly with the increasing of the failure rate. The reason of this result is that MH's processes are required to restart from the beginning of the program upon a failure in the original CMMP. However, the execution time of benchmark in CMMP integrated with the proposed fault tolerant scheme increases slowly with the increasing of the failure rate. As a result, the execution time is proportional to the number of local cluster nodes. The reason of this result is that MH's processes are just required to roll back to the latest local consistent checkpoint upon a failure in the alternative CMMP integrated with the proposed fault tolerant scheme. The proposed fault tolerant scheme ensures fewer loss of computing incurred by the failure.

## Conclusions

A checkpointing and rollback recovery scheme for the cluster-based multi-channel ad-hoc network is addressed specifically. The mobile hosts take checkpoints periodically managed by the local cluster head and log the output & input and messages related to gateways. In particular, we are interested in how to tolerate mobile host transient computing failure of ordinary hosts and the crash of the gateway between two neighbor clusters. This fault tolerant scheme can achieve the local consistent checkpoints in two beacon intervals, and invoke the rollback recovery procedure in only one beacon interval. The fault tolerant scheme can seamlessly cooperate with ad-hoc networks as only low additional power consumption is required.

## Acknowledgements

This work was supported by Natural Science Foundation of Jiangsu (Project No. BK2012237).

## References

- [1] G. Li, H. Wang. A novel min-process checkpointing scheme for mobile computing systems[J]. *Journal of Systems Architecture: the EUROMICRO Journal*, 2005 51(1) 45-61.
- [2] G.H. Cao, M. Singhal. Checkpointing with mutable checkpoints[J]. *Theoretical Computer Science*, 2003 290 1127-1148.
- [3] E.N. Elnozahy, L. Alvisi, Y.M. Wang and D.B. Johnson. A Survey of Rollback-Recovery Protocols in Message-Passing Systems[J]. *ACM Computing Surveys*, 2002 34(3) 375-408.
- [4] T. Park, N. Woo, H. Y. Yeom. An Efficient Optimistic Message Logging Scheme for Recoverable Mobile Computing Systems[J]. *IEEE Transactions on Mobile Computing*, 2002 1(4) 265-277.
- [5] I.R. Chen, B. Gu, S. E. George, and S. Cheng. On failure recoverability of client-server applications in mobile wireless environments[J]. *IEEE Transactions on Reliability*, 2005 54(1) 115-122.
- [6] Lili. Z., S. Boon-Hee . An Integrated Cluster-Based Multi-Channel MAC Protocol for Mobile Ad Hoc Networks[J]. *IEEE Transactions on Wireless Communications*, 2007 6(11) 3964-3974.
- [7] Gwo-Jong Yu, Chih-Yung Chang, An efficient cluster-based multi-channel management protocol for wireless Ad Hoc networks[J]. *Computer Communications*, 2007 30(8) 1742-1753.
- [8] T. PARK, K. KIM. Fault-Tolerance for the Mobile Ad-Hoc Environment[J]. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2008 E91-A (1) 413-416.