Chapter 63
Unreliable Failure Detectors for Mobile Ad–Hoc Networks

Luciana Arantes
University Paris 6, France

Fabíola Greve
Federal University of Bahia, Brazil

Pierre Sens
University Paris 6, France

ABSTRACT

Failure detection is an important abstraction for the development of fault-tolerant middleware, such as group communication toolkits, replication and transaction services. Unreliable failure detector (namely, FD) can be seen as an oracle which provides information about process failures. The dynamics and self-organization of Mobile Ad-hoc Networks (MANETs) introduce new restrictions and challenges for the implementation of FDs with which static traditional networks do not have to cope. It is worth mentioning that in some way, fault tolerance is more critical for MANETs than for the latter, since wireless network can present high error rates and mobile nodes are more prone to failures, physical damages or transient disconnections. The aim of this chapter is thus to discuss the impact of all these characteristics, intrinsic to MANET, in the implementation of FDs. It presents a survey of the few works about FD implementations for wireless networks, including the different possible assumptions to overcome the dynamics and lack of both global view and synchrony of MANETs.

INTRODUCTION

The distributed computing scenario is rapidly evolving for integrating unstructured, self-organizing and dynamic systems, such as peer-to-peer, wireless sensor and mobile ad-hoc networks. Nonetheless, the issue of designing reliable services which can support the high dynamics of these systems is a challenge. Current large-scale distributed applications are usually built on top of failure-prone asynchronous distributed systems, i.e., systems where there are no bounds for communication delays, neither for process speeds and nodes can crash. The design of fault-tolerant applications on top of such systems is a very difficult task (or even impossible) due to the

difficulty of correctly distinguish a crashed process from a process which is slow or with which communication is slow. There are even some fundamental problems, such as the **consensus**, which is impossible to be solved deterministically in a pure asynchronous distributed system where nodes can crash (Fischer, M. J., Lynch, M. J., & Paterson, M. S., 1985). Roughly, a consensus is an agreement problem which allows a set of processes to agree on a common proposal output value. To circumvent such difficulties and impossibilities, Chandra and Toueg (Chandra, T. D., & Toueg, S., 1996) have introduced the concept of unreliable failure detectors.

Unreliable failure detectors (namely, FDs) can be seen as oracles which provide information about process failures. Each process has access to a local failure detector module which, when queried, returns a list of processes that it currently suspects of having crashed. A local failure detector is unreliable in the sense that it may not suspect a crashed process or suspect a correct one. In other words, it may erroneously add to its list a process which is actually correct. However, if the detector later believes that suspecting this process is a mistake, it then removes the process from its list. FDs are abstractly characterized by two properties: *completeness* and *accuracy*. Two kinds of completeness and four kinds of accuracy are defined in (Chandra, T. D., & Toueg, S., 1996), which once combined yield eight classes of failure detectors. All of them can be used to solve the above mentioned consensus and many other agreement problems in a crash-prone asynchronous distributed system.

Many papers (Chandra, T. D., & Toueg, S., 1996), (Bertier, M., Marin, O., & Sens P., 2002), (Larrea, M., Arévalo, S., & Fernandez, A., 1999), (Gupta, I., Chandra, T. D., & Goldszmidt, G. S., 2001) in the literature have proposed to implement some or all of the classes of FDs. Nevertheless, the majority of them consider fully connected static networks with reliable links where nodes fail by crashing. The number of crashes is usually

bounded. Both the initial number of processes in the system and the identity of processes are known by all processes. In addition, some synchrony, such as eventually timely links or relative difference in link latencies, is usually added to the system. However, all these assumptions are not actually suitable for mobile ad hoc networks (MANETs) which are characterized as extremely dynamic systems where connections between nodes frequently change due to different reasons such as arbitrary failures, lack of node energy, limited bandwidth, disconnections, node arrivals and departures, mobility of nodes, etc. Furthermore, in MANETs, nodes do not have a global knowledge of the system and the number of participant nodes is unknown. The network is not fully connected and a node can only send messages to nodes that are within its transmission range. Hence, it may happen that a message sent by a node should be routed through a set of intermediate nodes until reaching the destination node. Therefore, the dynamics and self-organization of MANETs introduce new restrictions and challenges for the implementation of FDs to which static traditional networks do not face.

Considering the importance of unreliable failure detectors to solve some fundamental problems and the above dynamics of MANETs, some authors have recently proposed to implement FDs on top of MANET (Friedman, R., & Tcharny, G., 2005), (Tai, A., Tso, K.S., & Sanders, H., 2004), (Sens, P., Arantes, L., Bouillaguet, M., Simon, V., & Greve, F., 2008), (Cao, J., Raynal, M., Travers, C., & Wu, W., 2007), and (Sridhar, N., 2006). An advantage of providing a FD for these networks is that existing applications or algorithms that already run on top of static distributed systems using FDs could be easily ported to MANET. It is worth also pointing out that in some way, fault tolerance is more critical for MANETs than for the latter, since wireless network can present high error rates and mobile nodes are more prone to failures, physical damages or transient disconnections.

Related Content

3D Mesh Model Coding

Zhen Li (2011). *Streaming Media Architectures, Techniques, and Applications: Recent Advances (pp. 95-123).*

www.irma-international.org/chapter/mesh-model-coding/47516

Vehicular Delay Tolerant Networks

Daniel Câmara, Nikolaos Frangiadakis, Fethi Filaliand Christian Bonnet (2011). *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts (pp. 356-367).*

www.irma-international.org/chapter/vehicular-delay-tolerant-networks/50598

Multimedia Security Through 1D Chaotic Systems: Review and Analysis

Achraf Daouiand Ahmed A. Abd El-Latif (2023). *Recent Advancements in Multimedia Data Processing and Security: Issues, Challenges, and Techniques (pp. 1-31).*

www.irma-international.org/chapter/multimedia-security-through-1d-chaotic-systems/331432

Web Healthcare Applications in Poland: Trends, Standards, Barriers and Possibilities of Implementation and Usage of E-Health Systems

Anna Sotysik-Piorunkiewicz, Magorzata Furmankiewiczand Piotr Ziuziaski (2016). *Experimental Multimedia Systems for Interactivity and Strategic Innovation (pp. 258-283).*

www.irma-international.org/chapter/web-healthcare-applications-in-poland/135133

Infrastructures for Development of Context-Aware Mobile Applications

Hugo Feitosa de Figueirêdo, Tiago Eduardo da Silva, Anselmo Cardoso de Paiva, José Eustáquio Rangel de Queirozand Cláudio De Souza Baptista (2011). *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts (pp. 1104-1118).*

www.irma-international.org/chapter/infrastructures-development-context-aware-mobile/50642