Experimental Performance Evaluation of AODV Implementations

Koojana Kuladinithi, Asanga Udugama, Nikolaus A. Fikouras, Carmelita Görg
ComNets, Universität Bremen, Otto-Hahn-Allee NW1, 28359 Bremen, Germany
{koo|adu|niko|cg}@comnets.uni-bremen.de

ABSTRACT

The Ad hoc On-Demand Distance Vector (AODV) routing protocol is designed for use in mobile ad hoc networks. As of this study, there exist several implementations of the AODV protocol for a range of operating systems (e.g. Unix/Linux, Windows) and for different devices (e.g., notebook computers, PDAs) each developed using different techniques and programming languages. This study looks at two important areas of AODV protocol handlers: firstly the implementation aspects of two implementations compliant to RFC 3561, namely of JAdhoc an inter-platform AODV protocol handler in Java and of AODV-UU a Linux protocol handler written in C, both working in user space and kernel space. Secondly, an experimental comparison of these two implementations with respect to their effect on transport layer protocol (UDP and TCP) performance.

AODV is a widely researched protocol among the research community. Most of the research effort has focused on simulations aimed at determining the performance of AODV also in comparison to the performance of other ad hoc routing protocols. However, it has been largely unknown how implementations perform against each other and how different design decisions affect the performance of transport protocols.

The investigated scenario involves a stationary AODV test-bed that consists of 6 nodes connected using WLAN 802.11b. This scenario resembles the conditions encountered in conference rooms, class rooms, hotel rooms or home networks. The performance of UDP was evaluated with respect to parameters such as packet delivery fraction, offered load, out of order sequence packets and jitter. The performance of TCP was evaluated with respect to throughput together with TCP parameters of Retransmission Time Out (RTO) and congestion window size (cwnd). The Iperf traffic generator software was utilized for generation of UDP and TCP flows. Web100 was used to measure a selected set of TCP parameters. Each measurement was taken while changing the number of hops from two to six.

Our observations: Both implementations utilized different mechanisms to manage the behavior as described in the RFC 3561. These decisions did not make not much of a difference on how they operated except for the way packet buffering was done, which resulted in initial delays and unordered packets in the Java implementation, being a user space program. In general, performance degrades with an increasing number of nodes. More than UDP, TCP was effected as TCP's transmission control mechanism has to be adjusted with the initial delays of AODV route establishment.

It was also found that change of AODV parameters (e.g. Hello Loss & minimal lifetime) would affect the throughput, especially in TCP communication. Reconfiguring AODV parameters (depends on the network topology) can improve the transport layer performance. Current research issues include performing similar tests in a mobile environment, considering the movements and also determining the effect of value ranges of AODV parameters.

Key words: implementation issues, AODV testbed, interoperability, TCP/UDP performance