

[Research Topic home](#)

Proposal for a Hybrid Implementation of Ad-hoc On-demand Distance Vectoring (AODV)

Stephen Walter
December, 2008

Abstract

Reactive wireless networks are lacking a general protocol that is well suited for all environments. I explain how improvements have been applied to the Ad-hoc On-demand Distance Vectoring protocol, and discuss how these ideas can be combined into a new proposed protocol. This protocol is expected to have improved performance and capacity over the original. I plan to develop a simulation of the protocol to prove or disprove the anticipated outcome.

Background

As more communities demand wireless networks where no wired infrastructures have been established, ad-hoc protocols enter the scene to fulfill the role. Many recently proposed protocols have yet to mature and require modification to reduce their deficiencies while maintaining speed and robustness. I intend on taking one of the commonly known protocols, Ad-hoc On-demand Distance Vector (AODV) and redesigning its methods to address its overhead costs, allow easier integration with existing networks, and increase capacity over the original protocol.

There are two major contenders for ad-hoc routing that use an on-demand approach to save bandwidth on routing overhead: Ad-hoc On-demand Distance Vector and Dynamic Source Routing. Both protocols have been already been studied, simulated, and implemented by several researchers. People modified these protocols in ways such as inserting multiple routing paths to reduce rediscovery requests (Marina and Das, 1), or adding more radio channels to increase capacity (Pirzada, Portmann and Indulska, 2). Many of these changes have resulted in positive performance on tests. I plan on taking the successful concepts and integrating them into a single protocol.

Objective

My research will explore the potential of a hybrid AODV protocol that uses the strengths of the existing AODV, its speed and on-demand nature, while addressing the discovery flooding and the issue of ad-hoc networks requiring enough nodes to operate.

Protocol users need other users in the network for the ad-hoc system to work. I believe this can be accomplished by implementing a protocol that can coexist with the existing common wireless and wired protocols. This will allow a seamless transition into our existing network by letting the computer conduct regular network activities while having the AODV active and ready to use.

To encourage the use of AODV protocol in a wireless environment, it is necessary that AODV does not drain a system's resources or consume too much power. My proposed hybrid system will need to reduce the overhead of the original protocol produced by its route discovery flooding. One idea for this is to treat the network in a tiered tree-like fashion. Short routes will build connections in local networks, however when the IP is further away, I will introduce multi-radio bands to increase capacity. By changing the frequency, long range connections can extend their coverage without interfering with local connections. Each set of failed attempts to locate the destination will change the radio band for a longer range. Additionally, the protocol will attempt to find several unique routes during its request flooding. Since the relative network is already busy searching for the initial connection, searching for several alternative routes preemptively helps compact route requests whenever a connection is broken at a fraction of the cost of individual requests.

To allow the protocol further scalability, I propose to introduce a fixed routing node, which changes the protocol from a true ad-hoc system into a mixed structure. Any company or party that wishes to create access points over a larger distance can set up long range routers that spread around the ad-hoc zone. Each such router will listen to local requests via band frequency and save all IP information for a period of time in a cache, similar to how a proactive network would work. When a long range route request is made, these routers use specialized, directed, long range transmission to other routers in search of their destination, dependant on AODV hop limit. If any of these routers has seen the destination IP in a short range request, it will initiate a short range wireless search at the router using the reactive AODV method. The search is not guaranteed a result; however it should act as a hypothesis and should provide a good starting point.

The advantage of our method is that in high traffic zone where radio space is limit, it is more likely that the routers will see local IPs during their discovery process. Repeat searches will also benefit from cached routing. The downfall is that if connecting nodes are rarely speaking, these routers will not know about

the surrounding nodes, however such cases are unavoidable.

Methods

The proposed wireless protocol must be written in a low-level language that can easily translate across platforms while accessing hardware devices. I plan to use the 'C' language, since it is the most common and well known language and seems appropriate to the task. However, I plan to simulate the protocol instead of using real-live implementation as a far cheaper method of testing the protocol. Several network simulators exist, such as OpNet, QualNet, and NS-2. I plan to use NS-2 as a start, since it is freely available and supports the development of wireless protocols. My proposed protocol might operate beyond the existing simulation boundaries, so in such a situation, the plan is to construct a java based application that either parses the protocol code from 'C', or runs a hardcoded version of the protocol (although more likely the latter).

To evaluate the performance of my protocol, I will track the average speed of route discovery, delay across routes, number of lost connections, number of recovered routes through multipathing, overhead of routing information, process time consumed at each node, and other information over various network configurations. The simulation must consider both brief and long connections, as well as repeated connection requests to the same destination. While this should favor my protocol design, it is very reasonable that a computer would want to make connections to same destination to account for user interaction in situations like accessing websites.

Timeline

The proposed research is expected to span up to a two month period. I will spend the first few days formulating an objective design of my protocol. Using iterative development, I will construct a protocol that steadily adds functionality, which should take about two weeks to include all the elements. At that point, I will be ready to incorporate my protocol into the NS-2 system. This should take a few days of careful interfacing with NS-2. Next, I will conduct simulations and retrieve results to compare my protocol with other protocols. Simulation time is expected to take about a week.

If the NS-2 simulator falls through, the backup java application will require about one month to set up. I will need about four days to compile a list of requirements for the program. I will then design a data storage structure and system for local nodes on the network, as well as mechanics to trace information, which should take another four days. I exist implementations of C parsers online, which should speed up the process of development, however I will need to connect all hardware calls into the simulated network calls, which could take up to a week and a half. Finally, a GUI attachment should consume another four days, with program bug testing for around three days. When all is in working order,

this will leave a week to simulate my protocol.

Qualifications

As a Villanova undergraduate in the Computer Science department, I have ample experience in the languages used to design network protocols. I have studied the C language as early as middle school in the Borland IDE environment, and moved onto objective design in highschool in AP level credit C++ course. During college, I have used C in system programming courses and Assembly for instruction analysis. I understand the pointer and addressing system used in object orientation that is often obscured by the higher tiered languages.

More of my experience stems from the high level languages. I have taken several courses using the Java language, which included data structures for sorting and queuing trees and maps. My courses have covered a variety of languages, such as implementing equivalent code across functional languages of Lisp and Scheme. Outside of schoolwork, I have used C## in small work projects, as well as Microsoft's newer CLI in the .NET framework. I have also experimented with modem commands during work experience for one summer.

I have worked with several developing environments that are suitable for this project. I am most accustomed to the graphical IDE: Netbeans, Visual Studios, and Eclipse. Additionally, I am familiar with text editors: VI, Pico, and EMacs. I can operate all the debuggers in the IDE studios, as well as the GDB debugger in Unix.

Using the information I've collected over the course of my research project on the AODV protocol, I believe I have sufficient knowledge of the protocol to begin modifying its design. I may require support to work out specific tasks such as using the NS-2 simulator and working out hardware calls without having prior experience, but should be manageable.

Conclusion

Our society demands a standardized protocol that is flexible to various environments while improving existing capacity. My proposed protocol is expected to satisfy these requirements as well as to have simple implementation. With my broad experience over the wireless network protocols, I will be able to produce a working protocol within two months.

References

[1] Pirzada, A. Amir, Portmann, M. Portmann and Indulska, Jadwiga, "Evaluation of Multi-Radio Extensions to AODV for Wireless Mesh Networks," *Proceedings of the 4th ACM International Workshop on Mobility Management*

and Wireless Access pp. 45-51, 2006.

[2] Marina, Mahesh K. and Das, Samir R., "Ad hoc on-demand multipath distance vector routing," *Wireless Communications and Mobile Computing*, vol. 6, pp. 969-988, 2006.