Summary of Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks

Juho Törmä

28th February 2005

Abstract

Span is an algorithm for multi-hop ad hoc wireless networks that allows most of the nodes in the network to enter into a power saving sleep mode without significantly diminishing the capacity or connectivity of the network. Span achieves this by maintaining a routing backbone of a connected dominating set of the graph using only local decisions in each node. This paper summarizes Span as it is presented in [1] and provides some critique for it in section 7.

1 Introduction

Minimizing energy consumption is an important challenge in mobile networking. Since the radio of a network interface in a mobile device uses almost as much energy when idly listening as it does when transmitting or receiving, the simplest solution to conserve energy is to turn the radio off. This requires cooperation between power saving and routing protocols to enable transmission of packets through the network even when most of the nodes are sleeping. Span attempts to provide this coordination.

A good power saving coordination technique for wireless ad hoc networks ought to have the following characteristics:

- It should allow as many nodes as possible to turn their radio receivers off most of the time.
- It should forward packets between any source and destination with minimally more delay than if all nodes were awake.
- It should be distributed, requiring each node to make only local decisions.
- It should provide about as much total capacity as the original network to avoid increasing congestion.

- It should work with any link layer that provides for sleeping and periodic polling.
- It should inter-operate correctly with whatever routing system the ad hoc network uses.

Span fulfills the above requirements. Each node makes periodic, local decisions on whether to sleep or stay awake as a *coordinator* and participate in the forwarding backbone topology. To preserve capacity, a node volunteers to be a coordinator if it discovers that two of its neighbors cannot communicate with each other directly or through one or two existing coordinators. Each nodes delays announcing its willingness to become a coordinator by a random time that takes into account both the remaining battery energy and the number of pairs of neighbors it can connect together. This combination ensures, with a high probability, a capacity-preserving connected backbone at any point in time, where nodes tend to consume energy at about the same rate.

Since Span does all this using only local information, it scales very well with the number of nodes.

2 Related work

Span uses a connected dominating set of the ad hoc network as a backbone to route packets. A connected dominating set S of a graph G is a connected subgraph of G such that every vertex u in G is either in S or adjacent to some v in S. The paper lists three other works that discuss using or approximating connected dominating sets in ad hoc wireless networks.

The paper also lists several other works that discuss power-saving schemes that allow some of the nodes in a network to turn their radios off for some of the time. All of them are presented as lacking in one aspect or another when compared to Span.

A third category of related works listed in the paper is works that present power-saving by varying the transmission powers of the nodes. It is suggested that this could potentially be combined with Span.

3 Span design

Span is designed to achieve the following four goals:

- Ensure that there is at least one coordinator within radio range of every node.
- Rotate coordinators so that all nodes have the responsibility of providing global connectivity roughly equally.
- Attempt to minimize the number of coordinators without significant loss of capacity or an increase in latency.
- Require only local decisions from the nodes in the ad hoc network.

Span works by periodically broadcasting a HELLO message that contains the node's status, its current coordinators, and its current neighbors. From these messages, each node can construct a list of its neighbors and coordinators and each neighbors coordinators.

Using this locally available data, each non-coordinator node can calculate whether it should become a coordinator using the following criteria:

> **Coordinator eligibility rule.** A non-coordinator node should become a coordinator if it discovers, that two of its neighbors cannot reach each other either directly or via one or two coordinators.

To avoid several nodes becoming coordinators simultaneously and redundantly, a node delays announcing itself as coordinator by a random amount of time. Equation 1 was chosen for the delay, where N_i is the number of neighbors of node i, C_i is the number of additional pairs of nodes that would be connected if i became a coordinator, E_r and E_m are the amounts of remaining and maximum energies of the node, T is the round-trip delay of a packet and R is picked uniformly at random from the interval $(0, 1]$.

$$\text{delay} = \left(\left(1 - \frac{E_r}{E_m}\right) + \left(1 - \frac{C_i}{\binom{N_i}{2}}\right) + R\right) * N_i * T \tag{1}$$

Only if the above criteria holds even after this delay, does a node broadcast a HELLO message indicating that it is now a coordinator. The random back-off delay used depends on the number of additional pairs of neighbors a node would connect as a coordinator and the amount of energy the node has left.

The same criteria holds inversely and can be used by a redundant coordinator to withdraw. Coordinators should also withdraw after some period of time to rotate the burden of being a coordinator fairly. A coordinator withdraws by marking itself as a tentative coordinator. A tentative coordinator acts as a normal coordinator in forwarding packets, but it is not considered a coordinator by the coordinator announcement algorithm when determining the need for additional coordinators.

4 Simulator implementation

For performance evaluation purposes Span was implemented using the *ns-2* network simulator environment.

In the simulation Span was implemented on top of the 802.11 MAC layer and the actual routing was performed by a geographic forwarding algorithm. Geographic forwarding was selected for simplicity and it was implemented using the GOD module of ns to get the location of nodes. Other, more practical, routing protocols could be used as well.

Certain modifications were made to the power saving mode of the underlying 802.11 MAC layer to better accommodate the specific properties of Span. These include more efficient transmission of packets between coordinators, since they are always on there is no need for power saving techniques between them, and more restrictions for communication to and from non-coordinators to allow them to shut down as often as possible.

The energy model for the simulation was based on actual measurements of the transmission, receiving, idle and sleeping modes of a network interface card. These measurements confirmed the initial assumption behind the development of Span that there is a significant difference between the power consumptions of idle and sleeping modes. The idle consumption was over six times that of the sleeping mode.

5 Performance evaluation

The above simulator implementation of Span was ran on several static and mobile network topologies to determine its effectiveness. Span was compared against both unmodified 802.11 MAC in power saving mode (802.11.PSM) and unmodified 802.11 MAC not in power saving mode (802.11).

The capacity preservation property of Span was confirmed. Span performed better than 802.11 PSM and only slightly worse than 802.11 where all nodes are on all the time. Similar result were noted in latencies where Span performed significantly better than 802.11 PSM.

Simulations showed that mobility does not significantly affect routing with Span coordinators. In fact, the packet loss rate of Span was lower than both variants on unmodified 802.11.

Simulations also confirmed that Span elects more coordinators than a theoretical ideal number but also that it divides the responsibility of being a coordinator equally between nodes, as designed.

The most interesting results came from energy consumption and node lifetime where Span significantly outperformed both variants of unmodified 802.11. These results also showed that 802.11 PSM performed only slightly better than 802.11 without power saving.

6 Conclusion

Span is a distributed coordination technique for multi-hop ad hoc wireless networks that reduces energy consumption without significantly diminishing the capacity or connectivity of the network. Simulation results show that for many practical situations the system lifetime with Span is more than a factor of two better than without Span.

Future research should be concentrated on finding a more robust and efficient power saving MAC layer. This would achieve even greater energy savings, since in the current implementation of Span the large number of broadcast messages becomes expensive when the node density increases.

7 Critique

The paper proposes some modifications to the 802.11 power saving mode and all simulations of Span were run using these modifications. It would have been nice to see some measurements regarding the effects of these modifications. For example in the form of comparisons between Span with and without them or comparisons between 802.11 PSM with and without them.

Probably a common practical problem for a completely decentralized algorithm, such as Span, is the possibility of "parasitic nodes" that do not volunteer as coordinators. This is probably an issue that goes beyond the scope of the paper, but it did come into mind while reading it.

References

[1] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, *SPAN: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks.* Wireless Networks 8 (2002), 481-494.