Asynchronous Neighbor Discovery: Finding Needles of Connectivity in Haystacks of Time

Prabal Dutta, David Culler, and Scott Shenker
{prabal,culler,shenker}@cs.berkeley.edu
Computer Science Division
University of California, Berkeley
Berkeley, California 94720

1 Introduction

We present Disco, an asynchronous neighbor discovery and rendezvous protocol that allows two or more nodes operating their radios at low duty cycles (e.g. 1%) to discover and communicate with each other during opportunistic encounters and without any prior synchronization information. The key challenge is to operate the radio at a low duty cycle but still ensure that discovery is fast, reliable, and predictable over a range of operating conditions. Disco nodes pick a pair of prime numbers such that the sum of their reciprocals is equal to the desired radio duty cycle. Each node increments a local counter with a globally-fixed period. If a node's local counter value is divisible by either of its primes, then the node turns on its radio for one period. This protocol ensures that two nodes will have some overlapping radio on-time within a bounded number of periods, even if nodes independently set their own duty cycle.

2 The Discovery Algorithm

The idea behind the discovery algorithm is simple. Two nodes, i and j, pick two numbers, m_i and m_j, such that m_i and m_j are relatively prime and $1/m_i$ and $1/m_j$ are approximately equal to i and j's desired duty cycles, respectively. Time is divided into fixed-width *reference periods* and consecutive periods are labeled with consecutive integers. Nodes i and j start counting the passage of these periods at times a_i and a_j, with their respective counters, c_i and c_j, initialized to zero, and with i and j counts synchronized to the reference period (we will relax this last assumption in later sections). If $c_i|m_i$ (c_i is divisible by m_i), then i turns on its radio for one period and beacons. Similarly, if $c_j|m_j$, then j turns on its radio for one period and beacons. When both i and j turn on their radios during the same period, they can exchange beacons and discover each other.

Let us consider a concrete example. Let node i select $m_i = 3$ (so i's duty cycle is 33%), start counting at reference period $x = 2$ (so that $a_i = 2$), with counter values c_i. Similarly, let node j select $m_j = 5$ (so j's duty cycle is 20%), start counting at reference period $x = 1$ (so that $a_j = 1$), with counter values c_j. Table 1 illustrates these timelines and counter values.

Table 1: Example discovery timeline.

x	0	1	2	3	4	5	6	7	8	9	10	11
c_i	-	-	*0*	1	2	*3*	4	5	*6*	7	8	*9*
c_j	-	*0*	1	2	3	4	*5*	6	7	8	9	*10*

Italicized values of c_i indicate $c_i|m_i$ and italicized values of c_j indicate $c_j|m_j$. Columns where both c_i and c_j are italicized indicate values of x where both i and j have their radios turned on and can communication with each other. In this example, when $x = 11$, we see that both i and j are turned on, and are able to discover each other.

It is easy to show that there is exactly one such overlapping period every $m = m_i m_j$ periods. Letting x represent the reference period number, we have

$$\begin{aligned} c_i &= x - a_i \\ c_j &= x - a_j \end{aligned}$$

Our goal is to find an x such that $c_i|m_i$ and $c_j|m_j$. We can express this as a pair of simultaneous congruences

$$\begin{aligned} x &\equiv a_i \pmod{m_i} \\ x &\equiv a_j \pmod{m_j} \end{aligned}$$

Such a set of congruences are known to have a common solution by the Chinese remainder theorem. This theorem states that if x_0 is one such solution, then an integer x satisfies the congruences if and only if x is of the form $x = x_0 + km$ for some integer k. One x_0 is

$$x_0 = a_i b_i m_j + a_j b_j m_i$$

978-0-7695-3157-1/08 $25.00 © 2008 IEEE
DOI 10.1109/IPSN.2008.60

where the solution is unique $\pmod{m}$ for $m = m_i m_j$, and where b_i and b_j must satisfy the following congruences

$$\begin{aligned} b_i m_j &\equiv 1 \pmod{m_i} \\ b_j m_i &\equiv 1 \pmod{m_j} \end{aligned}$$

We can express our earlier example as the following simultaneous congruences

$$\begin{aligned} x &\equiv 2 \pmod{3} \\ x &\equiv 1 \pmod{5} \end{aligned}$$

and see that when $x = 11$, both congruences are solved

$$\begin{aligned} &(2-11)|3 \\ &(1-11)|5 \end{aligned}$$

An analytic solution requires finding b_i and b_j

$$\begin{aligned} 5b_i &\equiv 1 \pmod{3} \\ 3b_j &\equiv 1 \pmod{5} \end{aligned}$$

We see that values of $b_i = 2$ and $b_j = 2$ satisfy these congruences and hence one solution x_0 is

$$\begin{aligned} x_0 &= a_i b_i m_j + a_j b_j m_i \\ x_0 &= 2 \cdot 2 \cdot 5 + 1 \cdot 2 \cdot 3 \\ x_0 &= 26 \end{aligned}$$

Since all solutions are unique $\pmod{15}$, we have

$$x_0 = 26 \pmod{15} = 11$$

which agrees with our earlier solution from Table 1 and gives $x = 11 + 15k$, for all $k \in \mathbb{Z}^+$.

The preceding analysis sidesteps a number of practical considerations. Since, for example, the Chinese remainder theorem requires the moduli m_i and m_j be relatively prime to guarantee a solution to the simultaneous congruences, these values cannot be independently chosen by the nodes, which is limiting. We also required that nodes be able to express their desired duty cycle as the reciprocal of a positive integer (e.g. $1, 1/2, 1/3, \ldots, 1/k$, where $k \in \mathbb{Z}^+$), which is restrictive. We assumed that nodes i and j synchronized their counting with the reference phase, which aids analysis but is unlikely to hold in practice. The preceding analysis also fails to explore the effect of clock drift and counter overflow on discovery, ignores radio startup time and energy overhead, and assumes that communications jitter is negligible. Our work relaxes these assumptions but space constraints do not permit inclusion in this paper.

Figure 1 shows radio power state (green line), radio transmissions (blue line), and the system current (in mA/mV) during a radio "on slot" with a T_{period} of 25 ms. Note that 5 ms of inactivity occurs prior to a packet transmission.

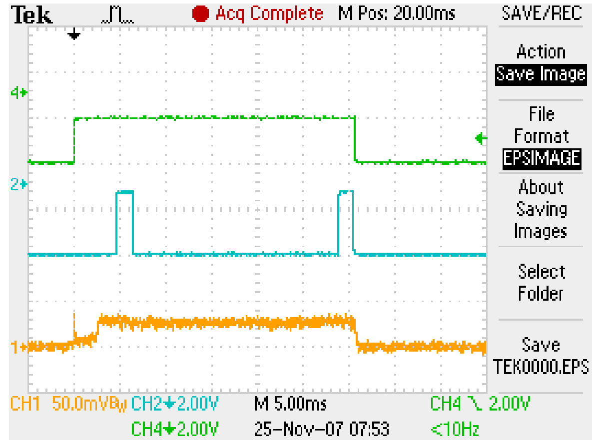

Figure 1: The timing and current draw details of a slot. $T_{count} = 25$ ms.

Figure 2 shows the discovery rate for two different T_{count} values (10 ms and 25 ms) using prime pair values of 100 and 101. The expected discovery time is 10100 slots, and with our slot durations, the worst-case discovery latency is 101 seconds.

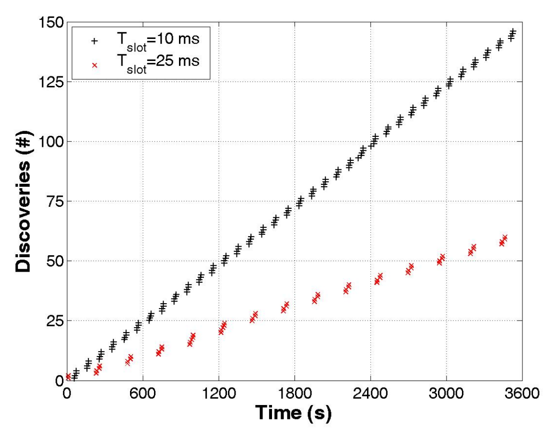

Figure 2: The discovery rate for two different T_{count} values.

3 Demonstration

We will demonstrate several nodes running the Disco protocol, asynchronously discovering each other, and sharing neighbor table information with new nodes to speed up discovery as more nodes are added to the the neighborhood.