Spray and Focus Protocol

Apurv Verma & Ashish Prasad

Introduction

- Based on the following research paper:

  Thrasyvoulos Spyropoulos, Student Member, IEEE, Konstantinos Psounis, Member, IEEE, and Cauligi S. Raghavendra, Fellow, IEEE,

- *"Efficient Routing in Intermittently Connected Mobile Networks - Multiple Copy Case",* 2006.
- *"Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility",* 2006.

Problem Setup

- Our problem set up consists of a number of nodes moving inside a bounded area according to a stochastic mobility model. Additionally, we assume that the network is disconnected at most times, and that transmissions are faster than node movement (i.e., it takes less time to transmit a message using the wireless medium—ignoring queuing delay—than to move it physically for the same distance using node mobility).

Need for a new Protocol?

- Intermittently connected mobile networks are wireless networks, where a complete path from the source to the destination does not exist most of the time.
- Conventional routing schemes fail, because they try to establish complete end-to-end paths, before any data is sent.
- To deal with such networks researchers have suggested to use flooding based routing schemes. However, they waste a lot of energy and suffer from severe contention, which degrades their performance seriously.
- With this in mind, we introduce a new family of routing schemes that "spray" a few message copies into the network, and then route each copy independently towards the destination.

Components of "Sr" Protocol

- Spray routing in intermittently connected mobile networks (ICMNs), consists of two phases:
- *Spray phase*: The source of a message initially starts with L copies; any node A that has n > 1 message copies (source or relay), and encounters another node B (with no copies), hands over to B _n/2_ and keeps _n/2_ for itself; when it is left with only one copy, it switches to direct transmission.
- *Focus phase*: Let $U_x(Y)$ denote the utility of node X for destination Y; a node A, carrying a copy for destination D, forwards its copy to a new node B it encounters, if and only if, U_B(D) > U_A(D) + Uth, where (utility threshold) where U_{th} is a parameter of the algorithm. The implementation of such a utility function has been realized by maintaining weighted time between encounters which is explained later in the slides.

Spray Phase : Implementation

- Data Structure:
  - All the nodes maintain a bufferList.
  - (‘source_addr_’ ,‘source_seq_num_’) serve as an identifier for a particular buffer record.

- Each buffer record contains
  - all the packets received by it in the pktList ; and
  - a vnList consisting of nsaddr_t values to which the node has already delivered message copy(s).

- Whenever a data packet is generated, the source node caches in its bufferList, the ‘SR_L’ copies of packets in a new packetList and initializes the vnList corresponding to a new (‘source_addr_’ ,‘source_seq_num_’).

- Receive_data
  - Whenever a packet is being received, it is appended to the packetList of the particular type in the bufferList.

Implementing forward_data_spray

- Each node sends data packets at regular intervals of times denoted by the macro IDATA which is defined in 'sr_constants.h' file.
  - The node scans through its bufferList and schedules data packets waiting to be sent.
  - If the node finds that it has more than one data copy of a particular ('source_addr_' ,'source_seq_num_'), then it sends the packets in spray fashion [send_data_spray(Packet* p)].
  - Othewise, if it has just one copy of a packet type, it sends the packet via focus routing.

- The node first looks at its 'rtable_' and determines a neighbor which is not in the 'vnList' of the particular packet ('source_addr_' ,'source_seq_num_') and then sends half of the packets in the packetList to that neighbor.

Focus Phase

- Each node broadcast control packets at regular intervals of times denoted by the macro ICTRL which is defined in ‘sr_constants.h’ file.
- The broadcasted control packets are received by all the neighbors of the node.
- Suppose X broadcasted ctrl packets to its neighbor Y. Y on receiving the packet finds the current time and updates its ‘time_of_last_encounter’ with X to the CURRENT_TIME and also maintains a value called ‘time_elapsed’ which denotes the avg time b/w successive encounters of X and Y. This is maintained using an exponential weighted avg formula which gives more preference to the recent encounters than to the older ones.

Focus Phase cont.

- deltaT = CURRENT_TIME - time_of_encounter;
- t_elapsed = BETA*(deltaT)+(1-BETA)*time_elapsed;

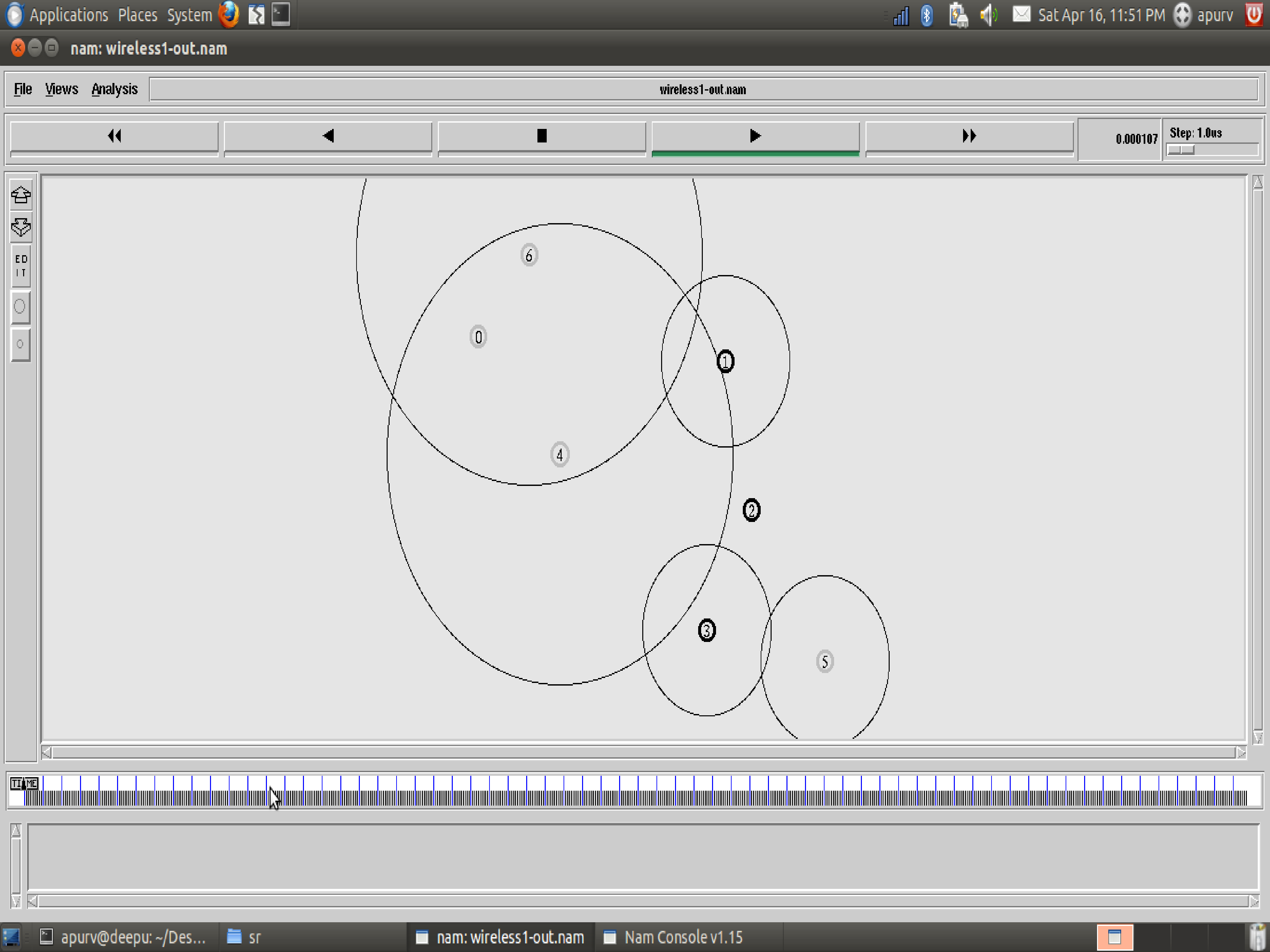
- This t_elapsed is a direct indicator of distance between nodes. If two nodes have small t_elapsed then it means that are very near to each other and vice versa.

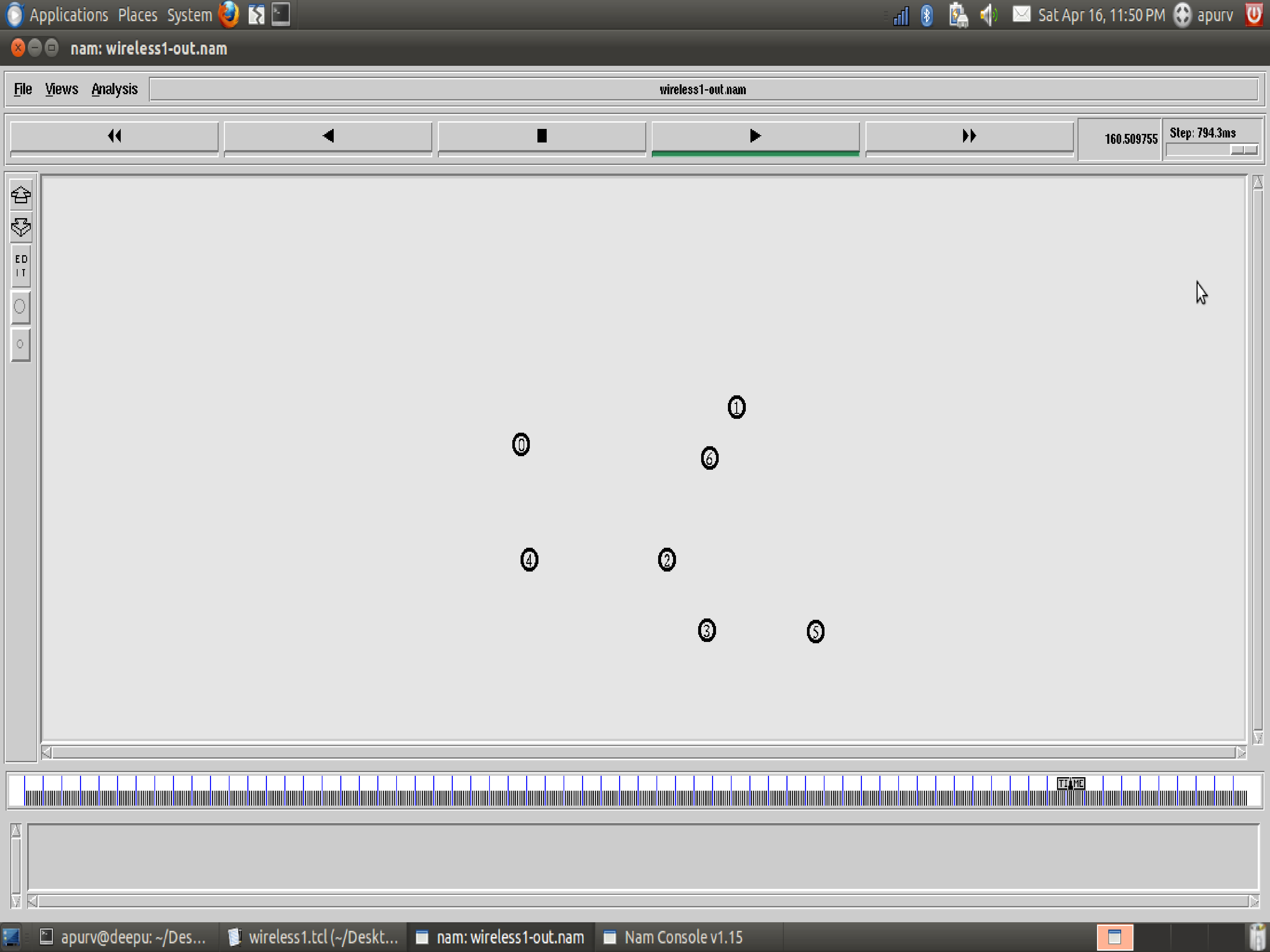
- But this value can be maintained only for nodes that are immediate neighbors. So how do we maintain t_elapsed for nodes which are not immediate neighbors of each other?
- To overcome this issue we assume that t_elapsed follows rules of transitivity,

Focus Phase cont.

✓ **CRUX:**

1. Consider 3 nodes A, B,D.
2. Let, t(X,Y) denotes time elapsed between successive encounters of nodes X and Y.
3. If t(A,D) > t(A,B)+ t(B,D) + constant, then A should route via B to D instead of going directly because B is more proximal to D than A.


Applications Places System
Sat Apr 16, 11:51 PM
apurv
nam: wireless1-out.nam
File
Views
Analysis
wireless1-out.nam
0.000107
Step: 1.0us
EDIT
6
0
1
4
2
3
5
TIME
apurv@deepu: ~/Des...
sr
nam: wireless1-out.nam
Nam Console v1.15


Applications Places System
Sat Apr 16, 11:50 PM apurv
nam: wireless1-out.nam
File Views Analysis
wireless1-out.nam
160.509755
Step: 794.3ms
ED
IT
1
0
6
4
2
3
5
TIME
apurv@deepu: ~/Des...
wireless1.tcl (~/Deskt...
nam: wireless1-out.nam
Nam Console v1.15

```
P 105.000000 _6_ Buffer Table
P   source_addr source_seq_num num_of_copies
P   5       37      1
P   5       35      4
P   5       34      4
P   5       33      4
P   5       32      4
P   5       30      4
P   5       25      5
P 105.000000 _6_ Routing Table
P   dest    next    elapsed      encounter   expire
P   0   0   10.258964   100.359144  130.359144
P   1   0   31.092175   -5.000000   130.359144
P   2   0   41.301889   -5.000000   130.359144
P   3   4   20.197219   -5.000000   130.116834
P   4   4   9.902733    100.116834  130.116834
P   5   4   30.492115   -5.000000   130.116834
```

THANK YOU